



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ ΔΙΑΤΑΞΕΩΝ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΑΠΟΦΑΣΕΩΝ

**Προσαρμογή Επιχειρησιακών
Διαδικασιών Οδηγούμενη από Γεγονότα
(Event-driven Business Process Adaptation)**

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

Ιωάννης Γ. Πατινωτάκης

Αθήνα, Ιούλιος 2015



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ ΔΙΑΤΑΞΕΩΝ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ

ΑΠΟΦΑΣΕΩΝ

Προσαρμογή Επιχειρησιακών Διαδικασιών Οδηγούμενη από Γεγονότα (Event-driven Business Process Adaptation)

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

Ιωάννης Γ. Πατινωτάκης

Τριμελής Συμβουλευτική Επιτροπή: *Γρηγόριος Μέντζας, Καθηγητής Ε.Μ.Π. (επιβλέπων)*
Βασίλειος Ασημακόπουλος, Καθηγητής Ε.Μ.Π.
Δημήτριος Ασκούνης, Καθηγητής Ε.Μ.Π.

Επταμελής Εξεταστική Επιτροπή:

Γρηγόριος Μέντζας
Καθηγητής Ε.Μ.Π.

Βασίλειος Ασημακόπουλος
Καθηγητής Ε.Μ.Π.

Ιωάννης Ψαρράς
Καθηγητής Ε.Μ.Π.

Δημήτριος Ασκούνης
Καθηγητής Ε.Μ.Π.

Χάρης Δούκας
Επ. Καθηγητής Ε.Μ.Π.

Δημήτριος Αποστόλου
Επ. Καθηγητής
Πανεπιστημίου Πειραιώς

Κωνσταντίνος Κούτσικος
Επ. Καθηγητής Πανεπιστημίου
Αιγαίου

Αθήνα, Ιούλιος 2015

.....

Ιωάννης Γ. Πατινωτάκης

Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © **Ιωάννης Γ. Πατινωτάκης, 2015**

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

*Αφιερώνεται στους γονείς μου
Γεώργιο και Δήμητρα*

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΠΕΡΙΛΗΨΗ	9
ABSTRACT	11
ΕΥΧΑΡΙΣΤΙΕΣ	13
1 ΕΙΣΑΓΩΓΗ	15
1.1 ΓΕΝΙΚΟ ΠΛΑΙΣΙΟ.....	16
1.2 ΚΙΝΗΤΡΟ	18
1.3 ΣΧΕΤΙΚΗ ΕΡΕΥΝΑ	19
1.4 ΕΥΡΟΣ ΤΗΣ ΔΙΑΤΡΙΒΗΣ ΚΑΙ ΕΡΕΥΝΗΤΙΚΟΣ ΣΤΟΧΟΣ	22
1.5 ΠΡΟΤΕΙΝΟΜΕΝΗ ΠΡΟΣΕΓΓΙΣΗ.....	23
1.6 ΚΑΙΝΟΤΟΜΙΕΣ – ΣΥΝΕΙΣΦΟΡΑ.....	24
1.7 ΔΟΜΗ ΔΙΑΤΡΙΒΗΣ	26
1.8 ΣΧΕΣΗ ΜΕ ΤΙΣ ΔΗΜΟΣΙΕΥΣΕΙΣ.....	26
1.9 ΣΧΕΣΗ ΜΕ ΕΡΕΥΝΗΤΙΚΑ ΈΡΓΑ	27
2 ΕΠΙΓΝΩΣΗ ΚΑΤΑΣΤΑΣΗΣ ΟΔΗΓΟΥΜΕΝΗ ΑΠΟ ΓΕΓΟΝΟΤΑ	29
2.1 ΠΛΑΙΣΙΟ ΤΗΣ ΒΙΒΛΙΟΓΡΑΦΙΚΗΣ ΕΡΕΥΝΑΣ	29
2.2 ΕΠΕΞΕΡΓΑΣΙΑ ΣΥΝΘΕΤΩΝ ΓΕΓΟΝΟΤΩΝ	30
2.3 ΠΛΑΙΣΙΟ ΚΑΙ ΕΠΙΓΝΩΣΗ ΠΛΑΙΣΙΟΥ	32
2.3.1 <i>Επίγνωση Πλαισίου (Context Awareness)</i>	33
2.4 ΚΑΤΑΣΤΑΣΗ ΚΑΙ ΕΠΙΓΝΩΣΗ ΚΑΤΑΣΤΑΣΗΣ.....	34
2.4.1 <i>Διαχείριση Καταστάσεων (Situation Management)</i>	35
2.4.2 <i>Επίγνωση Κατάστασης (Situation Awareness)</i>	36
2.4.3 <i>Εκτίμηση Καταστάσεων (Situation Assessment)</i>	38
2.5 ΚΑΙΝΟΤΟΜΕΣ ΠΡΟΣΕΓΓΙΣΕΙΣ ΓΙΑ ΟΔΗΓΟΥΜΕΝΗ ΑΠΟ ΓΕΓΟΝΟΤΑ ΕΠΙΓΝΩΣΗ ΚΑΤΑΣΤΑΣΕΩΝ.....	40
3 ΠΡΟΣΑΡΜΟΓΗ ΕΠΙΧΕΙΡΗΣΙΑΚΩΝ ΔΙΑΔΙΚΑΣΙΩΝ	43
3.1 ΕΥΕΛΙΞΙΑ ΣΥΣΤΗΜΑΤΩΝ ΚΑΙ ΔΙΑΔΙΚΑΣΙΩΝ (FLEXIBILITY)	43
3.2 ΚΑΤΗΓΟΡΙΟΠΟΙΗΣΗ ΠΡΟΣΑΡΜΟΓΩΝ.....	45
3.3 ΕΝΑΥΣΜΑΤΑ ΠΡΟΣΑΡΜΟΓΩΝ ΣΕ ΕΦΑΡΜΟΓΕΣ ΥΠΗΡΕΣΙΩΝ	47
3.3.1 <i>Παρακολούθηση Εφαρμογών για την Εκκίνηση Προσαρμογών</i>	47
3.3.2 <i>Δοκιμές Παρεμβολής και Δοκιμές Χρόνου Εκτέλεσης για την Εκκίνηση Προσαρμογών</i> ..	50
3.3.3 <i>Προσαρμογή οδηγούμενη από το Πλαίσιο</i>	52
3.4 ΠΡΟΣΑΡΜΟΓΗ ΔΙΑΔΙΚΑΣΙΩΝ ΚΑΙ ΕΦΑΡΜΟΓΩΝ	53
3.5 ΑΝΑΖΗΤΗΣΗ ΥΠΗΡΕΣΙΩΝ, ΚΑΝΟΝΕΣ ΠΡΟΣΑΡΜΟΓΗΣ ΚΑΙ ΠΡΟΣΑΡΜΟΓΕΙΣ.....	55
3.6 ΠΡΟΤΑΣΕΙΣ ΠΡΟΣΑΡΜΟΓΩΝ ΓΙΑ ΔΙΑΔΙΚΑΣΙΕΣ ΚΑΙ ΕΦΑΡΜΟΓΕΣ	58
3.7 ΠΡΟΣΑΡΜΟΓΗ ΔΙΑΔΙΚΑΣΙΩΝ ΟΔΗΓΟΥΜΕΝΗ ΑΠΟ ΓΕΓΟΝΟΤΑ & ΕΠΙΓΝΩΣΗ ΠΛΑΙΣΙΟΥ	60
4 ΠΡΟΤΑΣΗ ΤΗΣ ΔΙΑΤΡΙΒΗΣ	63
4.1 ΔΙΑΜΟΡΦΩΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ	63
4.1.1 <i>Η δική μας οπτική για το πρόβλημα</i>	65
4.2 ΕΡΕΥΝΗΤΙΚΑ ΕΡΩΤΗΜΑΤΑ	68
4.2.1 <i>Κύκλος Ζωής Προσαρμογών (Adaptation Life-Cycle)</i>	68

4.2.2	Ανίχνευση της ανάγκης προσαρμογής με χρήση γεγονότων.....	70
4.2.3	Προσαρμογή διαδικασιών οδηγούμενη από την επίγνωση κατάστασης	72
4.2.4	Συμβάσεις και Επιλογές της διατριβής.....	73
4.3	ΠΡΟΤΑΣΗ ΤΗΣ ΔΙΑΤΡΙΒΗΣ	75
4.3.1	Περίγραμμα της προτεινόμενης λύσης	75
4.3.2	Λήψη και αξιοποίηση γεγονότων	75
4.3.3	Ανίχνευση αναγκών προσαρμογής.....	77
4.3.4	Επιλογή και Πρόταση προσαρμογών.....	80
4.3.5	Εφαρμογή προσαρμογών	83
4.3.6	Υλοποίηση της προτεινόμενης προσέγγισης	86
4.3.7	Αντιστοίχιση του πλαισίου εργασίας με τα ερευνητικά ερωτήματα.....	86
4.3.8	Σύγκριση με τη βιβλιογραφία.....	88
5	ΔΙΚΤΥΑ ΚΑΤΑΣΤΑΣΗΣ – ΕΝΕΡΓΕΙΑΣ	93
5.1	ΣΧΕΤΙΚΕΣ ΠΡΟΣΕΓΓΙΣΕΙΣ ΣΤΗ ΒΙΒΛΙΟΓΡΑΦΙΑ.....	93
5.1.1	Ιεραρχικά Δίκτυα Εργασιών.....	94
5.1.2	Δυναμικός Σχεδιασμός	95
5.1.3	Δένδρα Συμπεριφορών στην Τεχνολογία Συμπεριφορών.....	95
5.1.4	Δένδρα Συμπεριφορών στην Τεχνητή Νοημοσύνη παιχνιδιών Η/Υ.....	97
5.2	ΠΑΡΟΥΣΙΑΣΗ ΤΩΝ ΔΙΚΤΥΩΝ ΚΑΤΑΣΤΑΣΗΣ – ΕΝΕΡΓΕΙΑΣ.....	99
5.2.1	Εισαγωγή – Σκοπός	99
5.2.2	Ορισμός των Δικτύων Κατάστασης – Ενέργειας.....	100
5.2.3	Τρόποι αναπαράστασης	104
5.2.4	Αλληλεπίδραση με το περιβάλλον	105
5.3	ΛΕΙΤΟΥΡΓΙΑ ΚΑΙ ΕΚΤΕΛΕΣΗ ΤΩΝ SANS	106
5.3.1	Οργάνωση και αποθήκευση των SANS.....	106
5.3.2	Εκτέλεση των SANS	107
5.3.3	Πλαίσιο εκτέλεσης.....	108
5.3.4	Αλγόριθμος διάσχισης.....	109
5.4	ΔΟΜΙΚΑ ΣΤΟΙΧΕΙΑ ΤΩΝ SANS	112
5.5	ΣΥΓΚΡΙΣΗ ΜΕ ΤΙΣ ΣΧΕΤΙΚΕΣ ΠΡΟΣΕΓΓΙΣΕΙΣ ΤΗΣ ΒΙΒΛΙΟΓΡΑΦΙΑΣ	120
5.5.1	Σύγκριση με κανόνες ECA	120
5.5.2	Σύγκριση με τη γλώσσα BPMN 2.0	124
5.5.3	Σύγκριση με Ιεραρχικά Δίκτυα Εργασιών και Αλγορίθμους Σχεδιασμού	126
5.5.4	Σύγκριση με Δένδρα Συμπεριφορών στην Τεχνητή Νοημοσύνη παιχνιδιών Η/Υ	128
5.5.5	Ικανοποίηση των ερευνητικών ερωτημάτων.....	130
6	ΥΛΟΠΟΙΗΣΗ ΤΩΝ ΔΙΚΤΥΩΝ ΚΑΤΑΣΤΑΣΗΣ – ΕΝΕΡΓΕΙΑΣ	133
6.1	ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΟΥ ΜΗΧΑΝΙΣΜΟΥ ΕΚΤΕΛΕΣΗΣ ΤΩΝ SANS (SAN ENGINE)	135
6.2	ΥΛΟΠΟΙΗΣΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ ΕΠΙΓΝΩΣΗΣ ΚΑΤΑΣΤΑΣΗΣ (SAS)	139
6.3	ΕΠΕΞΕΡΓΑΣΙΑ ΤΩΝ SANS (SAN EDITOR)	140
6.4	ΕΝΑΛΛΑΚΤΙΚΕΣ ΧΡΗΣΕΙΣ ΤΩΝ SANS.....	144
7	ΑΞΙΟΛΟΓΗΣΗ ΤΩΝ ΔΙΚΤΥΩΝ ΚΑΤΑΣΤΑΣΗΣ – ΕΝΕΡΓΕΙΑΣ	147
7.1	ΣΕΝΑΡΙΟ ΧΡΗΣΗΣ ΜΕ SANS	147
7.1.1	Σενάριο του παραδείγματος.....	151
7.1.2	Εκτέλεση του σεναρίου	155

7.2	ΑΠΟΔΟΤΙΚΟΤΗΤΑ ΧΡΗΣΗΣ ΤΩΝ ΓΕΓΟΝΟΤΩΝ	156
7.3	ΕΠΙΔΟΣΗ ΚΑΙ ΚΛΙΜΑΚΩΣΗ	160
7.4	ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΚΑΙ ΜΕΙΟΝΕΚΤΗΜΑΤΑ	162
8	ΘΕΜΑΤΟΣΤΡΕΦΗΣ ΠΡΟΣΑΡΜΟΓΗ – Η ΜΕΘΟΔΟΣ FLEXiBPMN2.0.....	165
8.1	ΕΙΣΑΓΩΓΗ.....	166
8.2	ΣΧΕΤΙΚΕΣ ΠΡΟΣΕΓΓΙΣΕΙΣ ΣΤΗ ΒΙΒΛΙΟΓΡΑΦΙΑ.....	166
8.2.1	<i>Βασικές έννοιες του Θεματοστρεφούς Προγραμματισμού.....</i>	<i>167</i>
8.2.2	<i>ΑΟ4BPEL</i>	<i>168</i>
8.2.3	<i>BPEL'η'Aspects.....</i>	<i>171</i>
8.2.4	<i>CEVICHE</i>	<i>172</i>
8.2.5	<i>Λοιπές εργασίες.....</i>	<i>173</i>
8.2.6	<i>Περιορισμοί.....</i>	<i>176</i>
8.3	ΠΡΟΣΕΓΓΙΣΗ ΤΗΣ ΛΥΣΗΣ	177
8.3.1	<i>Η μέθοδος FlexiBPMN2.0</i>	<i>177</i>
8.3.2	<i>Βασικές έννοιες της γλώσσας BPMN 2.0.....</i>	<i>179</i>
8.3.3	<i>Βασικά σημεία της FlexiBPMN2.0.....</i>	<i>179</i>
8.3.4	<i>Καινοτομίες της FlexiBPMN2.0</i>	<i>181</i>
8.3.5	<i>Είδη προσαρμογών.....</i>	<i>181</i>
8.3.6	<i>Τρόποι εφαρμογής των προσαρμογών.....</i>	<i>183</i>
8.4	Η ΠΛΑΤΦΟΡΜΑ ΠΡΟΤΑΣΕΩΣ ΠΡΟΣΑΡΜΟΓΩΝ (ARP)	183
8.4.1	<i>Χαρακτηριστικά</i>	<i>184</i>
8.4.2	<i>Αρχιτεκτονική της πλατφόρμας.....</i>	<i>185</i>
8.4.3	<i>Υλοποίηση του APES.....</i>	<i>187</i>
8.5	ΠΑΡΑΔΕΙΓΜΑ ΧΡΗΣΗΣ ΤΗΣ ΠΛΑΤΦΟΡΜΑΣ ARP.....	188
8.5.1	<i>Σενάριο του παραδείγματος.....</i>	<i>188</i>
8.5.2	<i>Παραδείγματα προσαρμογών</i>	<i>191</i>
8.6	ΣΥΓΚΡΙΣΗ ΜΕ ΣΥΣΤΗΜΑΤΑ ΠΡΟΤΑΣΕΩΝ (RECOMMENDER SYSTEMS).....	201
9	ΔΥΝΑΜΙΚΑ ΕΞΕΛΙΣΣΟΜΕΝΑ SANS.....	203
9.1	ΕΙΣΑΓΩΓΗ.....	204
9.2	ΜΕΘΟΔΟΙ ΚΑΙ ΕΡΓΑΛΕΙΑ ΓΙΑ ΛΗΨΗ ΑΠΟΦΑΣΕΩΝ ΜΕ ΠΟΛΛΑΠΛΑ ΚΡΙΤΗΡΙΑ.....	204
9.2.1	<i>Μέθοδοι της Θεωρίας Χρησιμότητας με Πολλαπλά Κριτήρια</i>	<i>205</i>
9.2.2	<i>Διαδικασία Ιεραρχικής Ανάλυσης και Διαδικασία Δικτυακής Ανάλυσης.....</i>	<i>205</i>
9.2.3	<i>Ασαφή Σύνολα και Ασαφής Πολυκριτηριακή Λήψη Αποφάσεων</i>	<i>206</i>
9.2.4	<i>Γλωσσική Διατεταγμένη Σταθμισμένη Άθροιση.....</i>	<i>206</i>
9.2.5	<i>Μέθοδοι Σχέσεων Υπεροχής</i>	<i>207</i>
9.2.6	<i>Προγραμματισμός Προτιμήσεων και Μέθοδοι Ανάλυσης Προτιμήσεων</i>	<i>208</i>
9.2.7	<i>Βελτιστοποίηση Πολλαπλών Στόχων.....</i>	<i>209</i>
9.3	ΠΡΟΣΕΓΓΙΣΗ ΕΠΕΚΤΑΣΗΣ ΤΩΝ SANS.....	209
9.3.1	<i>Δεξαμενές Ενεργειών (Action Pools)</i>	<i>210</i>
9.3.2	<i>Μετα-δεδομένα των SANS και Δεξαμενές Ενεργειών</i>	<i>211</i>
9.3.3	<i>Μέθοδοι Αναζήτησης και Επιλογής (Search and Selection Methods)</i>	<i>212</i>
9.3.4	<i>Πολιτική Επίλυσης (Resolution Policy).....</i>	<i>212</i>
9.3.5	<i>Πλεονεκτήματα της προτεινόμενης επέκτασης.....</i>	<i>213</i>
9.4	ΜΕΘΟΔΟΙ ΑΝΑΖΗΤΗΣΗΣ & ΕΠΙΛΟΓΗΣ ΚΑΙ ΠΟΛΙΤΙΚΗΣ ΕΠΙΛΥΣΗΣ.....	213
9.4.1	<i>Αναζήτηση και επιλογή με τη μέθοδο SMART.....</i>	<i>214</i>

9.4.2	Αναζήτηση και επιλογή με τη μέθοδο LOWA	214
9.4.3	Πολιτικές επίλυσης που υλοποιήθηκαν	216
9.5	ΥΛΟΠΟΙΗΣΗ – ΕΠΕΚΤΑΣΗ ΤΗΣ ΠΛΑΤΦΟΡΜΑΣ ΤΩΝ SANS.....	216
9.5.1	Παραμετροποίηση της αναζήτησης και επιλογής του μηχανισμού των SANS	216
9.6	ΕΠΕΞΕΡΓΑΣΙΑ SANS ΜΕ ΚΟΜΒΟΥΣ ΑΦΗΡΗΜΕΝΩΝ ΕΝΕΡΓΕΙΩΝ.....	220
10	ΑΞΙΟΛΟΓΗΣΗ ΤΩΝ ΔΥΝΑΜΙΚΑ ΕΞΕΛΙΣΣΟΜΕΝΩΝ SANS	225
10.1	ΠΑΡΑΔΕΙΓΜΑ ΧΡΗΣΗΣ ΚΟΜΒΟΥ ΑΦΗΡΗΜΕΝΩΝ ΕΝΕΡΓΕΙΩΝ ΓΙΑ ΠΡΟΤΑΣΕΙΣ ΠΡΟΣΑΡΜΟΓΩΝ	225
10.1.1	Επεξεργασία του κόμβου αφηρημένης ενέργειας του παραδείγματος	230
10.1.2	Εκτέλεση του παραδείγματος για την προσαρμογή “Aspect 4”	231
10.2	ΑΞΙΟΛΟΓΗΣΗ ΤΩΝ ΚΟΜΒΩΝ ΑΦΗΡΗΜΕΝΩΝ ΕΝΕΡΓΕΙΩΝ.....	234
10.2.1	Βελτίωση εκφραστικότητας με χρήση κόμβων αφηρημένων ενεργειών	234
10.2.2	Πειραματική αξιολόγηση	238
11	ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΠΕΡΑΙΤΕΡΩ ΈΡΕΥΝΑ.....	253
11.1	ΣΥΜΠΕΡΑΣΜΑΤΑ.....	253
11.2	ΠΕΡΙΟΡΙΣΜΟΙ ΚΑΙ ΠΙΘΑΝΕΣ ΕΠΕΚΤΑΣΕΙΣ.....	255
11.3	ΠΕΡΑΙΤΕΡΩ ΕΡΕΥΝΑ	256
11.4	ΕΠΙΛΟΓΟΣ.....	258
	ΔΗΜΟΣΙΕΥΣΕΙΣ ΚΑΙ ΑΝΑΚΟΙΝΩΣΕΙΣ.....	259
	ΜΕΤΑΦΡΑΣΗ ΌΡΩΝ	263
	ΒΙΒΛΙΟΓΡΑΦΙΑ – ΑΝΑΦΟΡΕΣ – ΠΑΡΑΠΟΜΠΕΣ	265
	ΠΑΡΑΡΤΗΜΑΤΑ	287
	ΠΑΡΑΡΤΗΜΑ Ι – ΠΑΡΑΔΕΙΓΜΑΤΑ ΚΟΜΒΩΝ SAN ΣΕ RDF/N3.....	289
	ΠΑΡΑΡΤΗΜΑ ΙΙ – ΠΑΡΑΔΕΙΓΜΑ ΚΟΜΒΟΥ ΑΦΗΡΗΜΕΝΗΣ ΕΝΕΡΓΕΙΑΣ & ΔΕΞΑΜΕΝΗΣ ΕΝΕΡΓΕΙΩΝ .	291
	ΠΑΡΑΡΤΗΜΑ ΙΙΙ – ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΔΙΕΠΑΦΗ (API) ΤΟΥ SAN ENGINE	301

ΠΙΝΑΚΑΣ ΣΧΗΜΑΤΩΝ

ΣΧΗΜΑ 1-1. ΠΡΟΒΛΕΨΗ ΑΝΑΠΤΥΞΗΣ ΤΗΣ ΑΓΟΡΑΣ ΒΡΜ ΠΑΓΚΟΣΜΙΩΣ (FORRESTER, 2013)	17
ΣΧΗΜΑ 2-1. ΠΛΑΙΣΙΟ ΤΗΣ ΒΙΒΛΙΟΓΡΑΦΙΚΗΣ ΕΡΕΥΝΑΣ	30
ΣΧΗΜΑ 2-2. Ο ΚΥΚΛΟΣ ΖΩΗΣ ΤΗΣ ΔΙΑΧΕΙΡΙΣΗΣ ΚΑΤΑΣΤΑΣΗΣ (JAKOBSON ET AL., 2007)	35
ΣΧΗΜΑ 4-1. ΠΑΡΑΓΟΝΤΕΣ ΠΟΥ ΕΠΗΡΕΑΖΟΥΝ ΤΙΣ ΕΠΙΧΕΙΡΗΣΙΑΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ	65
ΣΧΗΜΑ 4-2. ΣΧΕΣΕΙΣ ΤΩΝ ΠΑΡΑΓΟΝΤΩΝ ΠΟΥ ΕΠΗΡΕΑΖΟΥΝ ΤΙΣ ΕΠΙΧΕΙΡΗΣΙΑΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ	65
ΣΧΗΜΑ 4-3. ΚΥΚΛΟΣ ΖΩΗΣ ΤΩΝ ΠΡΟΣΑΡΜΟΓΩΝ	69
ΣΧΗΜΑ 4-4. ΚΥΚΛΟΣ ΖΩΗΣ ΤΩΝ ΠΡΟΣΑΡΜΟΣΤΙΚΩΝ ΕΦΑΡΜΟΓΩΝ (BUCCHIARONE ET AL., 2010)	70
ΣΧΗΜΑ 4-5. ΚΥΚΛΟΣ ΖΩΗΣ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΙΧΕΙΡΗΣΙΑΚΩΝ ΔΙΑΔΙΚΑΣΙΩΝ (TER HOFSTEDÉ ET AL., 2010)	70
ΣΧΗΜΑ 4-6. ΚΥΚΛΟΣ ΖΩΗΣ ΤΩΝ ΠΡΟΣΑΡΜΟΓΩΝ ΣΤΟ ΠΛΑΙΣΙΟ ΤΗΣ ΔΙΑΤΡΙΒΗΣ	74
ΣΧΗΜΑ 4-7. ΦΑΣΕΙΣ ΤΗΣ ΠΡΟΤΕΙΝΟΜΕΝΗΣ ΠΡΟΣΕΓΓΙΣΗΣ	75
ΣΧΗΜΑ 4-8. ΠΑΡΑΔΕΙΓΜΑ ΑΠΟΤΥΠΩΣΗΣ ΤΟΥ ΤΡΟΠΟΥ ΑΝΤΙΛΗΨΗΣ ΤΗΣ ΑΡΓΟΠΟΡΙΑΣ ΜΙΑΣ ΔΙΑΔΙΚΑΣΙΑΣ	79
ΣΧΗΜΑ 4-9. ΣΥΣΧΕΤΙΣΗ ΦΑΣΕΩΝ, ΜΕΘΟΔΩΝ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΤΟΥ ΠΡΟΤΕΙΝΟΜΕΝΟΥ ΠΛΑΙΣΙΟΥ ΕΡΓΑΣΙΑΣ	85
ΣΧΗΜΑ 4-10. ΛΟΓΙΚΗ ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΗΣ ΥΛΟΠΟΙΗΣΗΣ ΤΟΥ ΠΛΑΙΣΙΟΥ ΕΡΓΑΣΙΑΣ	86
ΣΧΗΜΑ 5-1. ΣΥΣΧΕΤΙΣΗ ΣΤΟΧΩΝ-ΣΥΜΠΕΡΙΦΟΡΩΝ ΚΑΙ ΚΑΤΑΣΤΑΣΕΩΝ	100
ΣΧΗΜΑ 5-2. ΒΑΣΙΚΗ ΔΟΜΗ ΕΝΟΣ SAN	103
ΣΧΗΜΑ 5-3. SAN ΔΥΟ ΕΠΙΠΕΔΩΝ ΜΕ ΑΝΑΛΥΣΗ ΣΕ ΥΠΟΣΤΟΧΟΥΣ	103
ΣΧΗΜΑ 5-4. ΙΕΡΑΡΧΙΑ ΚΑΤΑΣΤΑΣΕΩΝ ΤΟΥ SAN ΣΤΟ ΣΧΗΜΑ 5-3	104
ΣΧΗΜΑ 5-5. ΣΧΕΔΙΟ ΕΚΤΕΛΕΣΗΣ ΤΟΥ SAN ΣΤΟ ΣΧΗΜΑ 5-3, ΩΣ ΡΟΗ ΕΡΓΑΣΙΩΝ	104
ΣΧΗΜΑ 5-6. ΠΛΗΡΗΣ ΚΑΙ ΑΠΛΟΥΣΤΕΥΜΕΝΗ ΑΝΑΠΑΡΑΣΤΑΣΗ ΕΝΟΣ SAN	105
ΣΧΗΜΑ 5-7. ΚΑΝΟΝΙΚΗ ΚΑΙ BAIL OUT ΔΙΑΣΧΙΣΗ ΕΝΟΣ SAN	110
ΣΧΗΜΑ 5-8. ΠΑΡΑΔΕΙΓΜΑ SAN ΜΕ ΣΤΟΧΟ, ΥΠΟΣΤΟΧΟ ΚΑΙ DECORATOR	112
ΣΧΗΜΑ 6-1. Η ΛΟΓΙΚΗ ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΟΥ SAS	133
ΣΧΗΜΑ 6-2. ΛΟΓΙΚΗ ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΟΥ ΜΗΧΑΝΙΣΜΟΥ ΕΚΤΕΛΕΣΗΣ ΤΩΝ SANS	136
ΣΧΗΜΑ 7-1. ΙΕΡΑΡΧΙΑ ΚΑΤΑΣΤΑΣΕΩΝ ΤΟΥ SAN ΣΤΗΝ ΕΙΚΟΝΑ 7-7	154
ΣΧΗΜΑ 8-1. ΜΟΝΤΕΛΟ ΚΑΙ ΣΤΙΓΜΙΟΤΥΠΑ ΔΙΑΔΙΚΑΣΙΩΝ	179
ΣΧΗΜΑ 8-2. ΛΟΓΙΚΗ ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΗΣ ΠΛΑΤΦΟΡΜΑΣ ΠΡΟΤΑΣΕΩΝ ΠΡΟΣΑΡΜΟΓΩΝ	186
ΣΧΗΜΑ 8-3. ΤΟ ΜΟΝΤΕΛΟ ΤΗΣ ΔΙΑΔΙΚΑΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΚΡΙΣΕΩΝ ΣΕ ΒΡΜΝ 2.0	190
ΣΧΗΜΑ 8-4. ΠΡΟΣΑΡΜΟΓΕΣ “ΠΡΙΝ”, “ΜΕΤΑ” ΚΑΙ “ΠΕΡΙΕ” ΓΙΑ ΤΗ ΔΙΑΔΙΚΑΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΚΡΙΣΕΩΝ	192
ΣΧΗΜΑ 8-5. SAN ΓΙΑ ΤΗ ΔΗΜΙΟΥΡΓΙΑ ΠΡΟΤΑΣΕΩΝ ΠΡΟΣΑΡΜΟΓΩΝ ΓΙΑ ΤΗ ΔΙΑΔΙΚΑΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΚΡΙΣΕΩΝ	193
ΣΧΗΜΑ 8-6. ΔΙΑΓΡΑΜΜΑ ΑΛΛΗΛΕΠΙΔΡΑΣΗΣ ΚΑΤΑ ΤΗΝ ΕΦΑΡΜΟΓΗ ΤΗΣ ΠΡΟΣΑΡΜΟΓΗΣ ASPECT 2	194
ΣΧΗΜΑ 9-1. SAN ΜΕ ΚΟΜΒΟ ΑΦΗΡΗΜΕΝΗΣ ΕΝΕΡΓΕΙΑΣ ΚΑΙ ΟΙ ΙΔΙΟΤΗΤΕΣ ΑΥΤΟΥ	220
ΣΧΗΜΑ 9-2. SAN ΜΕ ΚΟΜΒΟ ΑΦΗΡΗΜΕΝΗΣ ΕΝΕΡΓΕΙΑΣ	221
ΣΧΗΜΑ 9-3. ΡΥΘΜΙΣΕΙΣ ΚΟΜΒΟΥ ΑΦΗΡΗΜΕΝΗΣ ΕΝΕΡΓΕΙΑΣ	221
ΣΧΗΜΑ 9-4. ΕΙΣΑΓΩΓΗ ΚΡΙΤΗΡΙΩΝ ΣΕ ΚΟΜΒΟ ΑΦΗΡΗΜΕΝΗΣ ΕΝΕΡΓΕΙΑΣ	222
ΣΧΗΜΑ 9-5. ΚΑΝΟΝΙΚΟΠΟΙΗΣΗ ΤΩΝ ΒΑΡΩΝ ΤΩΝ ΚΡΙΤΗΡΙΩΝ	222
ΣΧΗΜΑ 9-6. ΑΝΤΙΣΤΟΙΧΙΣΗ ΤΩΝ ΤΙΜΩΝ ΑΡΙΘΜΗΤΙΚΟΥ ΚΡΙΤΗΡΙΟΥ ΣΕ ΓΛΩΣΣΙΚΟΥΣ ΟΡΟΥΣ (LOWA)	223
ΣΧΗΜΑ 9-7. ΟΡΙΣΜΟΣ ΜΕΤΑ-ΔΕΔΟΜΕΝΩΝ ΣΕ ΔΕΞΑΜΕΝΗΣ ΕΝΕΡΓΕΙΩΝ	223
ΣΧΗΜΑ 9-8. ΟΡΙΣΜΟΣ ΤΩΝ ΤΙΜΩΝ ΤΩΝ ΚΡΙΤΗΡΙΩΝ	224
ΣΧΗΜΑ 10-1. ΔΙΑΔΙΚΑΣΙΑ ΧΕΙΡΙΣΜΟΥ ΚΡΙΣΕΩΝ	226
ΣΧΗΜΑ 10-2. SAN ΓΙΑ ΠΡΟΤΑΣΕΙΣ ΠΡΟΣΑΡΜΟΓΩΝ ΣΤΗ ΔΙΑΔΙΚΑΣΙΑ ΧΕΙΡΙΣΜΟΥ ΚΡΙΣΕΩΝ (ΜΕ ΧΡΗΣΗ ΔΕΞΑΜΕΝΗΣ ΕΝΕΡΓΕΙΩΝ)	227
ΣΧΗΜΑ 10-3. ΠΡΟΣΑΡΜΟΓΕΣ ΤΟΥ ΠΑΡΑΔΕΙΓΜΑΤΟΣ ΕΦΑΡΜΟΣΜΕΝΕΣ ΣΤΗ ΔΙΑΔΙΚΑΣΙΑ ΧΕΙΡΙΣΜΟΥ ΚΡΙΣΕΩΝ	230
ΣΧΗΜΑ 10-4. ΕΠΕΞΕΡΓΑΣΙΑ ΠΑΡΑΜΕΤΡΩΝ ΤΗΣ ΜΕΘΟΔΟΥ LOWA	231

ΣΧΗΜΑ 10-5. ΓΕΓΟΝΟΣ ΠΡΟΤΑΣΗΣ ΕΦΑΡΜΟΓΗΣ ΤΗΣ ΠΡΟΣΑΡΜΟΓΗΣ “ASPECT 2”	231
ΣΧΗΜΑ 10-6. ΓΕΓΟΝΟΣ ΠΡΟΤΑΣΗΣ ΕΦΑΡΜΟΓΗΣ ΤΗΣ ΠΡΟΣΑΡΜΟΓΗΣ “ASPECT 4”	231
ΣΧΗΜΑ 10-7. ΜΗΝΥΜΑΤΑ ΤΟΥ ΜΗΧΑΝΙΣΜΟΥ ΤΩΝ SANS ΚΑΘΩΣ ΑΠΟΣΤΕΛΛΕΙ ΤΗΝ ΠΡΟΤΑΣΗ ΕΦΑΡΜΟΓΗΣ ΤΗΣ ΠΡΟΣΑΡΜΟΓΗΣ “ASPECT 4”	232
ΣΧΗΜΑ 10-8. Η ΕΡΓΑΣΙΑ “STUDY ADVICE” (ΠΡΙΝ ΤΗΝ ΕΦΑΡΜΟΓΗ ΤΗΣ ΠΡΟΣΑΡΜΟΓΗΣ)	232
ΣΧΗΜΑ 10-9. ΜΗΝΥΜΑ ΤΗΣ ΔΙΑΔΙΚΑΣΙΑΣ ΤΟΥ IRSN – ΑΙΤΗΣΗ ΑΝΑΦΟΡΑΣ ΕΝΕΡΓΕΙΩΝ (ACTIVITY REPORT)	233
ΣΧΗΜΑ 10-10. ΛΗΨΗ ΑΝΑΦΟΡΑΣ ΕΝΕΡΓΕΙΩΝ ΚΑΙ ΕΚΚΙΝΗΣΗ ΤΗΣ ΕΡΓΑΣΙΑΣ “STUDY REPORT”	233
ΣΧΗΜΑ 10-11. Η ΕΡΓΑΣΙΑ ΤΗΣ ΚΑΝΟΝΙΚΗΣ ΡΟΗΣ, ΜΕΤΑ ΤΗΝ ΟΛΟΚΛΗΡΩΣΗ ΤΗΣ ΠΡΟΣΑΡΜΟΓΗΣ	233
ΣΧΗΜΑ 10-12. SAN ΜΕ ΔΕΞΑΜΕΝΗ ΕΝΕΡΓΕΙΩΝ	234
ΣΧΗΜΑ 10-13. SAN ΜΕ ΚΟΜΒΟ ΚΑΤ’ ΕΠΙΛΟΓΗ ΕΚΤΕΛΕΣΗΣ (SELECTOR)	235
ΣΧΗΜΑ 10-14. SAN ΜΕ ΚΟΜΒΟ ΣΕΙΡΙΑΚΗΣ ΕΚΤΕΛΕΣΗΣ (SEQUENCE)	235
ΣΧΗΜΑ 10-15. SAN ΜΕ ΚΟΜΒΟ ΠΑΡΑΛΛΗΛΗΣ ΕΚΤΕΛΕΣΗΣ ΤΥΠΟΥ ANY	236
ΣΧΗΜΑ 10-16. SAN ΜΕ ΚΟΜΒΟ ΠΑΡΑΛΛΗΛΗΣ ΕΚΤΕΛΕΣΗΣ ΤΥΠΟΥ ALL	236
ΣΧΗΜΑ 10-17. SAN ΜΕ ΚΟΜΒΟ ΚΑΤ’ ΕΠΙΛΟΓΗ ΕΚΤΕΛΕΣΗΣ ΜΕ ΧΡΗΣΗ ΥΠΟΣΤΟΧΩΝ ΓΙΑ ΕΠΙΛΟΓΗ	237
ΣΧΗΜΑ 10-18. ΤΟ SAN ΤΟΥ ΠΕΙΡΑΜΑΤΟΣ ΑΝΑΦΟΡΑΣ (BASELINE)	240
ΣΧΗΜΑ 10-19. SAN ΜΕ ΔΕΞΑΜΕΝΗ ΕΝΕΡΓΕΙΩΝ ΜΕ 10 ΕΠΙΛΟΓΕΣ	240
ΣΧΗΜΑ 10-20. SAN ΜΕ ΚΟΜΒΟ ΚΑΤ’ ΕΠΙΛΟΓΗ ΕΚΤΕΛΕΣΗΣ ΜΕ 10 ΕΠΙΛΟΓΕΣ	241
ΣΧΗΜΑ 10-21. ΜΕΣΟΣ ΧΡΟΝΟΣ ΕΠΕΞΕΡΓΑΣΙΑΣ ΓΕΓΟΝΟΤΩΝ ΑΝΑ ΡΥΘΜΟ ΑΠΟΣΤΟΛΗΣ	243
ΣΧΗΜΑ 10-22. ΜΕΣΟΣ ΧΡΟΝΟΣ ΕΠΕΞΕΡΓΑΣΙΑΣ ΓΕΓΟΝΟΤΩΝ ΑΝΑ ΠΕΙΡΑΜΑ	245
ΣΧΗΜΑ 10-23. ΜΕΣΟΣ ΧΡΟΝΟΣ ΕΠΕΞΕΡΓΑΣΙΑΣ ΓΕΓΟΝΟΤΩΝ ΑΝΑ ΑΡΙΘΜΟ ΚΡΙΤΗΡΙΩΝ	246
ΣΧΗΜΑ 10-24. ΜΕΣΟΣ ΧΡΟΝΟΣ ΕΠΕΞΕΡΓΑΣΙΑΣ ΓΕΓΟΝΟΤΩΝ ΑΝΑ ΡΥΘΜΟ ΑΠΟΣΤΟΛΗΣ	247
ΣΧΗΜΑ 10-25. ΜΕΓΙΣΤΟ ΠΛΗΘΟΣ ΕΝΕΡΓΩΝ ΣΤΙΓΜΙΟΤΥΠΩΝ SANS/ΝΗΜΑΤΩΝ ΑΝΑ ΡΥΘΜΟ ΑΠΟΣΤΟΛΗΣ	247
ΣΧΗΜΑ 10-26. ΜΕΓΙΣΤΟ ΠΛΗΘΟΣ ΕΝΕΡΓΩΝ ΣΤΙΓΜΙΟΤΥΠΩΝ SANS/ΝΗΜΑΤΩΝ ΑΝΑ ΠΕΙΡΑΜΑ	248
ΣΧΗΜΑ 10-27. ΜΕΓΙΣΤΟ ΠΛΗΘΟΣ ΕΝΕΡΓΩΝ ΣΤΙΓΜΙΟΤΥΠΩΝ SANS/ΝΗΜΑΤΩΝ ΑΝΑ ΡΥΘΜΟ ΑΠΟΣΤΟΛΗΣ ΚΑΙ ΑΝΑ ΑΡΙΘΜΟ ΚΡΙΤΗΡΙΩΝ	249
ΣΧΗΜΑ 10-28. ΜΕΓΙΣΤΟ ΠΛΗΘΟΣ ΕΝΕΡΓΩΝ ΣΤΙΓΜΙΟΤΥΠΩΝ SANS/ΝΗΜΑΤΩΝ ΑΝΑ ΡΥΘΜΟ ΑΠΟΣΤΟΛΗΣ ΚΑΙ ΑΝΑ ΑΡΙΘΜΟ ΚΡΙΤΗΡΙΩΝ	249

ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ

ΕΙΚΟΝΑ 5-1. ΠΑΡΑΔΕΙΓΜΑ SAN ΚΑΙ Ο ΑΝΤΙΣΤΟΙΧΟΣ ΚΑΝΟΝΑΣ ECA ΓΙΑ ΤΟ DROOLS ECA ENGINE	122
ΕΙΚΟΝΑ 6-1. ΣΤΙΓΜΙΟΤΥΠΟ ΑΠΟ ΤΗ ΛΕΙΤΟΥΡΓΙΑ ΤΟΥ SAN ENGINE – CLI	140
ΕΙΚΟΝΑ 6-2. ΠΑΡΑΔΕΙΓΜΑΤΑ ΤΩΝ ΔΥΟ ΟΠΤΙΚΟΠΟΙΗΣΕΩΝ SANS ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΕΙ Ο SAN EDITOR	141
ΕΙΚΟΝΑ 6-3. Ο ΕΠΕΞΕΡΓΑΣΤΗΣ SANS (SAN EDITOR)	142
ΕΙΚΟΝΑ 6-4. ΤΑ ΕΡΓΑΛΕΙΑ ΕΠΕΞΕΡΓΑΣΙΑΣ ΤΩΝ SANS	142
ΕΙΚΟΝΑ 6-5. ΤΑ ΠΑΡΑΘΥΡΑ ΕΠΙΛΟΓΗΣ (ΑΡΙΣΤΕΡΑ) ΚΑΙ ΕΠΕΞΕΡΓΑΣΙΑΣ ΤΩΝ SANS (ΔΕΞΙΑ)	142
ΕΙΚΟΝΑ 6-6. ΕΝΑΛΛΑΚΤΙΚΗ ΑΠΕΙΚΟΝΙΣΗ ΤΟΥ ΕΠΙΛΕΓΜΕΝΟΥ SAN	143
ΕΙΚΟΝΑ 6-7. ΣΤΙΓΜΙΟΤΥΠΑ ΤΟΥ ΠΑΡΑΘΥΡΟΥ ΕΠΕΞΕΡΓΑΣΙΑΣ ΙΔΙΟΤΗΤΩΝ	143
ΕΙΚΟΝΑ 6-8. ΠΑΡΑΘΥΡΟ ΕΠΕΞΕΡΓΑΣΙΑΣ ΤΟΥ CONTEXTUALIZATION ΤΟΥ ΕΠΙΛΕΓΜΕΝΟΥ ΚΟΜΒΟΥ-ΚΑΤΑΣΤΑΣΗΣ	144
ΕΙΚΟΝΑ 6-9. ΠΑΡΑΔΕΙΓΜΑ CERAT ΣΕ EP-SPARQL ΚΑΙ ΜΕ ΣΗΜΕΙΟΓΡΑΦΙΑ SANS	145
ΕΙΚΟΝΑ 6-10. ΠΑΡΑΔΕΙΓΜΑ ΔΙΑΔΙΚΑΣΙΑΣ ΣΕ BPMN ΚΑΙ ΜΕ ΣΗΜΕΙΟΓΡΑΦΙΑ SANS	145
ΕΙΚΟΝΑ 7-1. ΟΘΟΝΗ ΤΟΥ AISHUB.NET	148
ΕΙΚΟΝΑ 7-2. ΤΡΕΧΟΥΣΕΣ ΠΛΗΡΟΦΟΡΙΕΣ ΕΝΟΣ ΠΛΟΙΟΥ	149
ΕΙΚΟΝΑ 7-3. ΚΥΚΛΟΦΟΡΙΑ ΣΕ ΠΕΡΙΟΧΗ ΤΗΣ ΣΙΝΙΚΗΣ ΘΑΛΑΣΣΑΣ	149
ΕΙΚΟΝΑ 7-4. ΛΕΠΤΟΜΕΡΕΙΣ ΠΛΗΡΟΦΟΡΙΕΣ ΠΛΟΙΟΥ ΜΕ ΠΡΟΗΓΟΥΜΕΝΟΥΣ ΠΡΟΟΡΙΣΜΟΥΣ	150
ΕΙΚΟΝΑ 7-5. ΠΑΡΑΠΛΕΟΝΤΑ ΠΛΟΙΑ	150

ΕΙΚΟΝΑ 7-6. ΤΟ SAN ΓΙΑ ΤΗΝ ΑΣΦΑΛΗ ΠΛΕΥΣΗ ΜΕ ΜΕΓΑΛΕΣ ΤΑΧΥΤΗΤΕΣ (1/2)	151
ΕΙΚΟΝΑ 7-7. ΤΟ SAN ΓΙΑ ΤΗΝ ΑΣΦΑΛΗ ΠΛΕΥΣΗ ΜΕ ΜΕΓΑΛΕΣ ΤΑΧΥΤΗΤΕΣ (2/2)	152
ΕΙΚΟΝΑ 7-8. ΕΙΔΟΠΟΙΗΣΗ ΓΙΑ ΠΛΕΥΣΗ ΣΕ ΘΥΕΛΛΩΔΕΙΣ ΣΥΝΟΗΚΕΣ.....	155
ΕΙΚΟΝΑ 7-9. ΓΕΓΟΝΟΣ ΣΤΗ ΡΟΗ ΓΕΓΟΝΟΤΩΝ «ΕΙΔΟΠΟΙΗΣΕΙΣ ΛΙΜΕΝΙΚΗΣ ΑΡΧΗΣ»	156
ΕΙΚΟΝΑ 7-10. ΑΡΙΘΜΟΣ SANs ΣΕ ΣΧΕΣΗ ΜΕ ΤΟ ΣΥΝΟΛΙΚΟ ΑΡΙΘΜΟ ΓΕΓΟΝΟΤΩΝ	157
ΕΙΚΟΝΑ 7-11. ΣΥΓΚΡΙΣΗ ΧΡΗΣΙΜΟΠΟΙΗΣΗΣ ΓΕΓΟΝΟΤΩΝ	158
ΕΙΚΟΝΑ 7-12. ΧΡΗΣΙΜΟΠΟΙΗΣΗ ΓΕΓΟΝΟΤΩΝ.....	159
ΕΙΚΟΝΑ 7-13. ΕΓΓΡΑΦΕΣ, ΔΙΑΓΡΑΦΕΣ ΚΑΙ ΕΝΕΡΓΕΣ ΕΓΓΡΑΦΕΣ ΣΕ ΡΟΕΣ ΓΕΓΟΝΟΤΩΝ	160
ΕΙΚΟΝΑ 7-14. ΣΥΓΚΡΙΣΗ ΤΗΣ ΧΡΗΣΗΣ ΜΝΗΜΗΣ	161
ΕΙΚΟΝΑ 7-15. ΚΛΙΜΑΚΩΣΗ ΤΟΥ ΧΡΟΝΟΥ ΕΠΕΞΕΡΓΑΣΙΑΣ ΣΕ ΣΧΕΣΗ ΜΕ ΤΟΝ ΑΡΙΘΜΟ ΤΩΝ ΕΝΕΡΓΩΝ SANs	161

ΠΙΝΑΚΑΣ ΠΙΝΑΚΩΝ

ΠΙΝΑΚΑΣ 4-1. ΕΡΕΥΝΗΤΙΚΑ ΕΡΩΤΗΜΑΤΑ ΚΑΙ ΟΙ ΠΑΡΑΜΕΤΡΟΙ ΤΟΥΣ.....	73
ΠΙΝΑΚΑΣ 4-2. ΑΝΤΙΣΤΟΙΧΙΣΗ ΤΟΥ ΠΡΟΤΕΙΝΟΜΕΝΟΥ ΠΛΑΙΣΙΟΥ ΕΡΓΑΣΙΑΣ ΜΕ ΤΑ ΕΡΕΥΝΗΤΙΚΑ ΕΡΩΤΗΜΑΤΑ	87
ΠΙΝΑΚΑΣ 4-3. ΣΥΓΚΡΙΣΗ ΜΕ ΤΗ ΒΙΒΛΙΟΓΡΑΦΙΑ	88
ΠΙΝΑΚΑΣ 4-4. ΣΥΓΚΡΙΣΗ ΤΗΣ ΠΡΟΤΑΣΗΣ ΤΗΣ ΔΙΑΤΡΙΒΗΣ ΜΕ ΤΟ CEVICHE.....	89
ΠΙΝΑΚΑΣ 5-1. ΑΛΓΟΡΙΘΜΟΣ ΕΚΤΕΛΕΣΗΣ ΤΩΝ SANs ΣΕ ΨΕΥΔΟΚΩΔΙΚΑ	111
ΠΙΝΑΚΑΣ 5-2. ΣΥΝΟΠΤΙΚΗ ΣΥΓΚΡΙΣΗ ΚΑΝΟΝΩΝ ECA ΚΑΙ SANs.....	123
ΠΙΝΑΚΑΣ 5-3. ΣΥΝΟΠΤΙΚΗ ΣΥΓΚΡΙΣΗ BPMN2.0 ΚΑΙ SANs	125
ΠΙΝΑΚΑΣ 5-4. ΣΥΝΟΠΤΙΚΗ ΣΥΓΚΡΙΣΗ HTNs ΚΑΙ SANs	127
ΠΙΝΑΚΑΣ 5-5. ΣΥΝΟΠΤΙΚΗ ΣΥΓΚΡΙΣΗ ΔΕΝΔΡΩΝ ΣΥΜΠΕΡΙΦΟΡΩΝ ΚΑΙ SANs	129
ΠΙΝΑΚΑΣ 5-6. ΠΕΡΙΟΡΙΣΜΟΙ ΤΩΝ ΠΡΟΣΕΓΓΙΣΕΩΝ ΣΕ ΣΧΕΣΗ ΜΕ ΤΟ ΠΡΩΤΟ ΕΡΕΥΝΗΤΙΚΟ ΕΡΩΤΗΜΑ	130
ΠΙΝΑΚΑΣ 7-1. ΠΑΡΑΔΕΙΓΜΑ ΔΥΝΑΜΙΚΟΥ ΚΑΝΟΝΑ ECA ΣΤΟ DROOLS.....	159
ΠΙΝΑΚΑΣ 8-1. ΒΑΣΙΚΟΙ ΤΥΠΟΙ ΠΡΟΣΑΡΜΟΓΩΝ	182
ΠΙΝΑΚΑΣ 8-2. ΠΑΡΑΔΕΙΓΜΑΤΑ ΣΥΝΘΕΤΩΝ ΤΥΠΩΝ ΠΡΟΣΑΡΜΟΓΩΝ	182
ΠΙΝΑΚΑΣ 8-3. ΕΙΚΟΝΕΣ ΑΠΟ ΤΗΝ ΕΦΑΡΜΟΓΗ ΤΗΣ ASPECT 1 (“BEFORE”)	195
ΠΙΝΑΚΑΣ 8-4. ΕΙΚΟΝΕΣ ΑΠΟ ΤΗΝ ΕΦΑΡΜΟΓΗ ΤΗΣ ASPECT 2 (“AFTER”)	197
ΠΙΝΑΚΑΣ 8-5. ΕΙΚΟΝΕΣ ΑΠΟ ΤΗΝ ΕΦΑΡΜΟΓΗ ΤΗΣ ASPECT 3 (“AROUND”)	199
ΠΙΝΑΚΑΣ 10-1. ΣΥΓΚΡΙΣΗ ΤΩΝ ΜΕΣΩΝ ΧΡΟΝΩΝ ΕΠΕΞΕΡΓΑΣΙΑΣ ΓΕΓΟΝΟΤΩΝ ΤΩΝ ΔΥΟ ΜΕΘΟΔΩΝ, ΑΝΑ ΡΥΘΜΟ ΑΠΟΣΤΟΛΗΣ ΚΑΙ ΠΛΗΘΟΣ ΕΠΙΛΟΓΩΝ	245
ΠΙΝΑΚΑΣ 11-1. ΕΡΕΥΝΗΤΙΚΑ ΕΡΩΤΗΜΑΤΑ, ΠΑΡΑΜΕΤΡΟΙ ΚΑΙ ΠΡΟΤΑΣΕΙΣ ΤΗΣ ΔΙΑΤΡΙΒΗΣ.....	254

Περίληψη

Η προσαρμογή (*adaptation*) αποτελεί μια κρίσιμη ιδιότητα των επιχειρησιακών διαδικασιών (*business processes*) αφού επιτρέπει να τροποποιείται η λειτουργία τους ανάλογα με τις συνθήκες που επικρατούν, έτσι ώστε να συνεχίζουν να επιτυγχάνουν τους στόχους τους. Όμως τόσο η διαπίστωση της ανάγκης για προσαρμογή όσο και η επιλογή των πιο κατάλληλων αλλά και η εφαρμογή των αντίστοιχων τροποποιήσεων, είναι θέματα που παρουσιάζουν ιδιαίτερες προκλήσεις.

Στην παρούσα διδακτορική διατριβή εξετάζεται το πρόβλημα προσαρμογής διαδικασιών (*business process adaptation*) με αξιοποίηση γεγονότων (*events*) από το περιβάλλον ή και το εσωτερικό των διαδικασιών. Ειδικότερα εστιάζει στην απόκτηση «επίγνωσης της κατάστασης» (*situation awareness*) των διαδικασιών, δηλαδή την «κατανόηση» των συνθηκών του περιβάλλοντος και των αιτιών τους, με βάση τα λαμβανόμενα γεγονότα. Στη συνέχεια η επίγνωση αυτή οδηγεί την επιλογή και κατόπιν εφαρμογή κατάλληλων προσαρμογών στις διαδικασίες που το χρειάζονται.

Συγκεκριμένα, προτείνεται ένα πλαίσιο εργασίας το οποίο επιχειρεί να δώσει μια απ' άκρο εις άκρο απάντηση στο ζήτημα αυτό. Εισάγονται και παρουσιάζονται τα Δίκτυα Κατάστασης – Ενέργειας (*Situation–Action Networks*) ως ένα νέο μεθοδολογικό εργαλείο για ανίχνευση των αναγκών προσαρμογής από γεγονότα, και επιλογή των καταλληλότερων από ένα σύνολο εναλλακτικών. Ακόμη προτείνεται η χρήση μιας νέας μεθόδου (*FlexiBPMN2.0*) βασισμένης στον Θεματοστρεφή προγραμματισμό (*aspect-oriented programming*) για εφαρμογή των προσαρμογών σε διαδικασίες αποτυπωμένες με χρήση της γλώσσας μοντελοποίηση διαδικασιών BPMN 2.

Στο πλαίσιο της διατριβής πραγματοποιήθηκε βιβλιογραφική μελέτη στα γνωστικά πεδία της προσαρμογής διαδικασιών και εφαρμογών βασισμένων σε υπηρεσίες ιστού, στο πεδίο της επίγνωσης καταστάσεων καθώς της επεξεργασίας σύνθετων γεγονότων. Επίσης κατά περίπτωση εξετάστηκαν μέθοδοι και τεχνολογίες που θα μπορούσαν να παρέχουν απάντηση σε συγκεκριμένα ζητήματα που απασχόλησαν τη διατριβή.

Λέξεις Κλειδιά

Επιχειρησιακές διαδικασίες, BPMN 2.0, Επίγνωση Κατάστασης, Προσαρμογή Οδηγούμενη από Γεγονότα, Θεματοστρεφής Προσαρμογή

ABSTRACT

Adaptation is a significant property of *business processes* since it enables the modification of their operation, in accordance to the on-going environment conditions, in order to continue pursuing and achieving their business goals. However, identifying the need for adaptation or choosing the right adaptation and applying it onto a running business process are non-trivial and challenging tasks.

In this thesis the problem of *business process adaptation* using *events* from their internal or external environment, is investigated. Specifically, the focus is placed on acquiring *situation awareness* for business processes, which is the comprehension of the process environment conditions as well as their causes, based on the received events. Situation awareness subsequently drives the selection of suitable adaptation choices and their application onto the processes in need.

In this thesis we propose a framework that aspires to provide an end-to-end solution to the aforementioned issue. Specifically, *Situation–Action Networks (SANs)* are introduced as a new methodological tool for identifying adaptation needs using events, and selecting the right adaptations from a pool of alternative adaptations. Furthermore, we propose a method for adaptation application on BPMN 2.0 business processes (called *FlexiBPMN2.0*), which is based on principles and techniques of aspect–oriented programming.

A state-of-the-art survey has also been conducted in the scientific fields of business process adaptation, service-based applications adaptation, situation awareness and complex event processing. Furthermore, where necessary we have also examined methods and techniques that could provide inspiration or a solution to specific issues of this thesis.

Keywords

Business Processes, BPMN 2.0, Situation Awareness, Event-Driven Adaptation, Aspect-Oriented Adaptation

ΕΥΧΑΡΙΣΤΙΕΣ

Η παρούσα διατριβή είναι η κατάληξη μιας προσπάθειας που πραγματοποιήθηκε στο πλαίσιο του προγράμματος μεταπτυχιακών σπουδών της σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Ηλεκτρονικών Υπολογιστών του Εθνικού Μετσόβιου Πολυτεχνείου. Υπήρξε ένα ενδιαφέρον και συναρπαστικό "ταξίδι" που μου έδωσε την ευκαιρία να αναπτυχθώ επιστημονικά και προσωπικά, αλλά και να δημιουργήσω νέες φιλίες. Το ταξίδι αυτό δεν θα είχε πραγματοποιηθεί χωρίς τη συνδρομή πολλών αξιόλογων ανθρώπων.

Κατ' αρχήν δε θα μπορούσε να ολοκληρωθεί χωρίς τη διαρκή συνδρομή, καθοδήγηση και παρότρυνση του επιβλέποντός μου, Καθηγητή κ. Γ. Μέντζα. Του οφείλω ιδιαίτερες ευχαριστίες για τις ευκαιρίες που μου προσέφερε, την υπομονή και την πίστη του σε μένα. Ευχαριστώ ακόμη τον επίκουρο καθηγητή, κ. Δ. Αποστόλου για τις πολύτιμες συμβουλές, σχόλια και την εποικοδομητική κριτική του που συνέβαλλε στη βελτίωση της δουλειάς μου.

Θα ήθελα να ευχαριστήσω και τα άλλα δύο μέλη της τριμελούς συμβουλευτικής επιτροπής, τους Καθηγητές κ. Β. Ασημακόπουλο και κ. Δ. Ασκούνη, καθώς και τον Καθηγητή κ. Ι. Ψαρρά και τους Επίκουρους Καθηγητές Χ. Δούκα, Κ. Κουτσίκο και κ. Δ. Αποστόλου για την τιμή που μου έκαναν να συμμετάσχουν στην επιτροπή εξέτασης της διατριβής.

Ένα μεγάλο ευχαριστώ αναλογεί στους συναδέλφους μου Ευθύμιο Μπόθο, Χαράλαμπο Μαγγούτα καθώς και τα υπόλοιπα μέλη της ομάδας. Η συναναστροφή μαζί τους αποτέλεσε απ' την αρχή πηγή υποστήριξης και έμπνευσης για μένα. Ιδιαίτερη μνεία θα ήθελα να κάνω στους φίλους και συνεργάτες μου Γιάννη Βεργινάδη και Νίκο Παπαγεωργίου που μοιραστήκαμε κοινά ερευνητικά ενδιαφέροντα και προσπάθειες.

Επίσης, θα ήθελα να ευχαριστήσω τους φίλους μου Στάθη, Νίκο, Φάνη, Δημήτρη και Ειρήνη για την κατανόηση και την υπομονή που έδειξαν όσον καιρό έγραφα τη διατριβή και απουσίαζα από τις κοινές μας δραστηριότητες. Το ξέρω ότι με σκεφτόσασταν παιδιά. ☺

Ολοκληρώνοντας θα ήθελα να εκφράσω την ευγνωμοσύνη μου στους γονείς μου και τις αδερφές μου, για την αμέριστη συμπαράσταση, υπομονή και ηθική υποστήριξη που μου προσέφεραν όλα αυτά τα χρόνια. Τους οφείλω πολύ περισσότερα απ' ότι θα μπορούσα να εκφράσω. Σας ευχαριστώ όλους για την αγάπη σας!

Τη διατριβή αυτή την αφιερώνω στη μητέρα μου Δήμητρα και τον πατέρα μου Γεώργιο, που "έφυγε" πριν από λίγους μήνες.

Ιωάννης Γ. Πατινωτάκης
Ιούλιος 2015

1 Εισαγωγή

Το σύγχρονο κοινωνικοοικονομικό γίνεσθαι, όπως έχει διαμορφωθεί τις τελευταίες δεκαετίες, χαρακτηρίζεται από διεθνοποίηση των αγορών, ανταγωνιστικότητα, συνεχείς και ταχύτατες αλλαγές, διεύρυνση της γεωγραφικής επιρροής καθώς και αύξηση της πολυπλοκότητας των σχέσεων και αλληλεπιδράσεων. Το γεγονός αυτό συμπαράσχει αντίστοιχα σχεδόν όλες τις πτυχές της ζωής σε όλα τα επίπεδα (από το προσωπικό έως και διακρατικό). Ένας κύριος μοχλός πίσω από αυτήν την τάση είναι η αλματώδης πρόοδος της τεχνολογίας σε πολλαπλούς τομείς όπως οι επικοινωνίες, πληροφορική – υπολογιστές, μεταφορές και άλλα. Ακόμη, η προτυποποίηση πολλών σημαντικών αγαθών, υπηρεσιών, τεχνολογιών και διαδικασιών, από τους αντίστοιχους διεθνείς φορείς, διευκολύνει αυτήν την τάση. Εξάλλου, προς την κατεύθυνση αυτή συντείνουν διακρατικές συμφωνίες (όπως ήταν η Γενική Συμφωνία Δασμών και Εμπορίου GATT) αλλά και υπερεθνικοί σχηματισμοί (όπως η Ευρωπαϊκή Ένωση και ο Διεθνής Οργανισμός Εμπορίου) που αίρουν τα εμπόδια στην διακίνηση αγαθών, υπηρεσιών, κεφαλαίων και φυσικά ανθρώπων.

Το διαμορφούμενο αυτό και διαρκώς μεταβαλλόμενο περιβάλλον έχει αλλάξει άρδην το πλαίσιο μέσα στο οποίο «διαβιούν» και δραστηριοποιούνται οι σύγχρονες επιχειρήσεις, οργανισμοί καθώς και διοικήσεις κρατών, περιφερειών ή διακρατικών σχηματισμών. Έτσι τα παραδοσιακά μοντέλα διοίκησης και επιχειρείν που χρησιμοποιούνταν για πολλές δεκαετίες έχουν τεθεί σε αμφισβήτηση και τρόπον τινά προκαλούνται. Οι σύγχρονοι οργανισμοί και επιχειρήσεις έχουν πλέον να αντιμετωπίσουν προκλήσεις κι ανταγωνισμό από έναν ευρύτερο γεωγραφικό χώρο. Αυτό δημιουργεί απαιτήσεις για γρηγορότερη ανταπόκριση στις εκάστοτε συνθήκες και αλλαγές που συμβαίνουν, προκειμένου να διατηρήσουν ή και επαυξήσουν την ποιότητα των προϊόντων και υπηρεσιών που παράγουν ή παρέχουν, αφού πλέον έχουν να ανταγωνιστούν περισσότερους. Ακόμη απαιτείται να μπορούν να διαχειριστούν και εκμεταλλευτούν την ποικιλομορφία που παρουσιάζει το νέο περιβάλλον αλλά και να αντιμετωπίσουν τη συμπίεση του περιθωρίου κέρδους που προκύπτει από τον εντονότερο ανταγωνισμό. Παράλληλα επιχειρήσεις και οργανισμοί καλούνται να αξιοποιήσουν στο μέγιστο δυνατό βαθμό τις ευκαιρίες και δυνατότητες που

παρέχει ένα τέτοιο διεθνοποιημένο περιβάλλον αφού πλέον έχουν πρόσβαση σε ένα πολύ ευρύτερο αλλά και ετερογενές κοινό.

Για να μπορέσουν λοιπόν να συνεχίσουν να υπάρχουν αλλά και να λειτουργούν αποδοτικά, επιχειρήσεις και οργανισμοί (τουλάχιστον οι επιτυχημένοι από αυτούς) βρίσκονται σε μια συνεχή προσπάθεια προσαρμογής στο παραπάνω πλαίσιο. Για το σκοπό αυτό αξιοποιούν και εκμεταλλεύονται όλους τους πόρους και τα μέσα που έχουν στη διάθεσή τους ή που μπορούν να αποκτήσουν εφόσον κριθεί απαραίτητο. Πέραν των παραδοσιακών πόρων και μέσων όπως είναι το κεφάλαιο, οι πρώτες ύλες, ο εξοπλισμός, οι υπηρεσίες από τρίτους, το ανθρώπινο δυναμικό και άλλα, τα τελευταία χρόνια γίνεται ολοένα φανερότερη η σημασία εννοιών όπως η επιχειρησιακή γνώση (corporate knowledge) και οι επιχειρησιακές διαδικασίες (business processes) που ενυπάρχουν κατά τρόπο «άδηλο» μέσα σε μια επιχείρηση ή οργανισμό αλλά χρησιμοποιούνται και επηρεάζουν αποφασιστικά την αποτελεσματικότητα και την επιτυχία τους. Αυτά έχουν πλέον αναγνωριστεί ως σημαντικότερα άυλα περιουσιακά στοιχεία (Dicken and Gould, 2008; Drean, 2010; Peloquin, 2001) ενώ έχουν αναπτυχθεί και προταθεί μεθοδολογίες, πλαίσια και εργαλεία για τη διαχείρισή τους (Dumas et al., 2013; Frost, 2015; Ko, 2009; Panagacos, 2012).

1.1 Γενικό Πλαίσιο

Οι επιχειρησιακές διαδικασίες και οι διαχείρισή τους αποτελούν ιδιαίτερο πεδίο έρευνας και εφαρμογής (Business Process Management). Ως επιχειρησιακή διαδικασία¹ (Business Process) ορίζεται ένα σύνολο λογικά συνδεδεμένων δραστηριοτήτων (activities) και εργασιών (tasks) για την επίτευξη ενός στόχου (goal) ή την παραγωγή ενός αποτελέσματος (Arrián, 2015; businessdictionary.com, 2015). Διακρίνονται σε διοικητικές, λειτουργικές και υποστηρικτικές διαδικασίες (management, operational and supporting processes). Σχετίζεται δε με την έννοια της «ροής εργασιών» (workflow) που είναι η τρόπον τινά «μετακίνηση» υλικών, προϊόντων ή πληροφοριών και αναθέσεων από μια εργασία σε μια άλλη κι από τον ένα ρόλο στον άλλο.

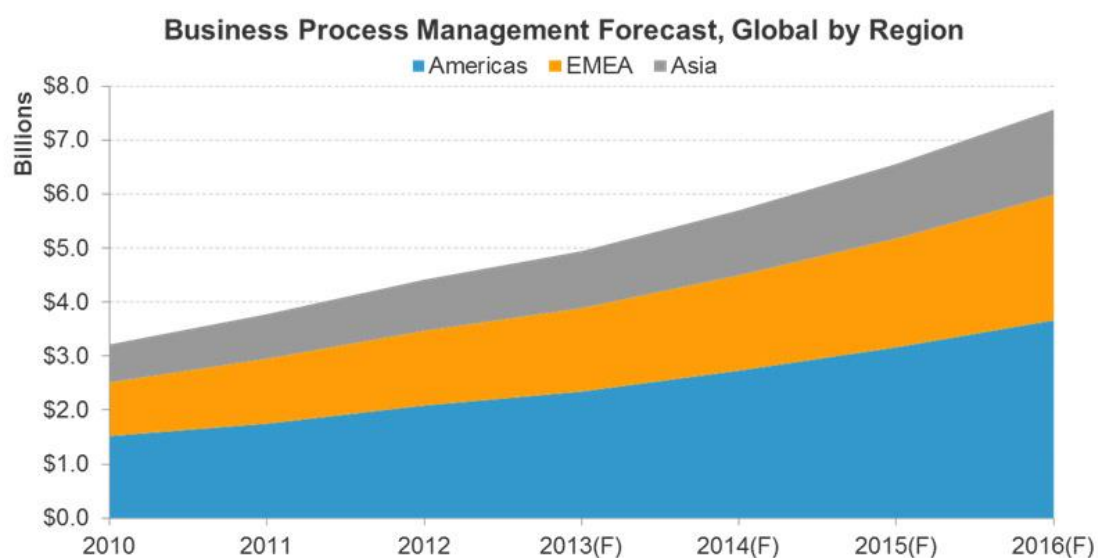
Επιχειρήσεις και οργανισμοί που ενεργά ασχολούνται με τη διαχείριση των διαδικασιών τους, επιχειρούν να καταγράφουν, τεκμηριώνουν, τυποποιούν και παρακολουθούν τη δημιουργία, εξέλιξη και εκτέλεση των διαδικασιών. Τέτοιες προσπάθειες διασφαλίζουν ότι οι διοικήσεις τους θα έχουν πάντα γνώση και έλεγχο του τρόπου λειτουργίας των μονάδων που διοικούν, ενώ με κατάλληλες παρεμβάσεις μπορούν να τον προσαρμόζουν στα εκάστοτε δεδομένα που παρουσιάζονται ή στους στόχους που θέτουν.

Η τυποποίηση των διαδικασιών συμβάλει αποφασιστικά στο να μπορεί να επαναληφθεί ένα επιτυχημένο αποτέλεσμα ή μοντέλο εργασίας ενώ επιπλέον επιτρέπει την εξειδίκευση, τη γρηγορότερη διεκπεραίωση – εκτέλεση των διαδικασιών, την ευκολότερη ή ταχύτερη εκμάθηση και φυσικά την οικονομικότερη λειτουργία τους. Προς την κατεύθυνση αυτή έχουν αναπτυχθεί και προταθεί συστήματα και εργαλεία που υποβοηθούν και εν μέρει

¹ Αφορά τόσο τις εμπορικές επιχειρήσεις όσο και μη εμπορικούς οργανισμούς αλλά και κυβερνήσεις κρατών ή διοικήσεις φορέων

αυτοματοποιούν την καταγραφή, διαχείριση και εκτέλεση των επιχειρησιακών διαδικασιών. Τέτοια εργαλεία μεγεθύνουν τα οφέλη της τυποποίησης αλλά παρέχουν και νέες δυνατότητες σχεδιασμού, παρακολούθησης (της εκτέλεσης) και διαχείρισης των διαδικασιών. Τυπικά παρέχουν μηχανισμούς για αυτόματη και αναλυτική καταγραφή των εισόδων, ενεργειών και αποτελεσμάτων των διαδικασιών από πολλαπλά σημεία και πηγές, την εφαρμογή τεχνικών επιχειρησιακής ευφυΐας (business intelligence), τεχνικών εξόρυξης δεδομένων και μοτίβων συμπεριφορών (data and pattern mining), προσομοίωσης και επαλήθευσης επιλογών και αποφάσεων, καθώς και της δημιουργίας αναφορών (reporting) και επισημάνσεων για κρίσιμες αλλαγές παραγόντων και συνθηκών που επηρεάζουν τις διαδικασίες.

Σύμφωνα με έρευνα της (Forrester, 2013), η αγορά λύσεων διαχείρισης επιχειρησιακών διαδικασιών (BPM market), παρότι υφίσταται για αρκετά χρόνια, συνεχίζει να παρουσιάζει ανάπτυξη (αν και με σημάδια κάμψης). Χαρακτηρίζεται δε από την ύπαρξη παρόχων λύσεων (vendors) μικρού αλλά και μεγάλου μεγέθους καθώς και από ευκαιρίες για περαιτέρω ανάπτυξη. Το μέγεθός της παγκοσμίως προβλέπεται ότι θα συνεχίσει να αυξάνεται από τα 3.2δισ USD το 2010 στα 7.5δισ USD το 2016, επιτυγχάνοντας σύνθετο ρυθμό ετήσιας ανάπτυξης 18.7% (CAGR)². Ο ρυθμός αυτός είναι σημαντικά υψηλότερος από το 11% της παγκόσμιας αγοράς λογισμικού στο σύνολό της.



Σχήμα 1-1. Πρόβλεψη ανάπτυξης της αγοράς BPM παγκοσμίως (Forrester, 2013)

Επιβεβαιώνοντας τα παραπάνω, η (Gartner, 2014) αναφέρει ότι πολλές τεχνολογίες BPM χρησιμοποιούνται πλέον ευρέως, κι ωστόσο συνεχώς εμφανίζονται νεότερες. Μια τέτοια αναδυόμενη τεχνολογία είναι η **Οδηγούμενη από Γεγονότα Διαχείριση Επιχειρησιακών Διαδικασιών (Event-Driven BPM)**³, η οποία αφορά άμεσα την παρούσα διατριβή. Μάλιστα

² Σύμφωνα με την MarketsandMarkets (<http://www.marketsandmarkets.com>) ο ρυθμός αυτός εκτιμάται σε 17,9% από το 2014 έως το 2019, ενώ η αύξηση της αγοράς από τα 4,71δισ USD το 2014 σε 10,73δισ USD το 2019 (marketsandmarkets.com, 2014).

³ Ως Event-Driven BPM ορίζεται το σχεδιαστικό υπόδειγμα (design paradigm) όπου η σειρά των εργασιών προσδιορίζεται δυναμικά εφαρμόζοντας κανόνες σε γεγονότα, καθώς αυτά συμβαίνουν.

την έχει συμπεριλάβει στον Κύκλο Υπερβολής (Hype Cycle⁴) για τη BPM, υπογραμμίζοντας έτσι το ενδιαφέρον της γι' αυτήν, ενώ εκτιμά ότι υπάρχουν σημαντικές δυνατότητες ανάπτυξης και τα προσδοκώμενα οφέλη αναμένει να είναι επίσης υψηλά. Η τεχνολογία τοποθετείται στην αρχική φάση (Innovation Trigger) του κύκλου υπερβολής, με αναμενόμενο χρόνο ευρείας υιοθέτησής της (mainstream adoption) από 5 έως 10 έτη. Η τρέχουσα διείσδυση της τεχνολογίας στην αγορά BPM κυμαίνεται από 1% έως 5%.

Σύμφωνα πάντα με την ίδια αναφορά (Gartner, 2014), τα αναμενόμενα οφέλη είναι η μεγαλύτερη ευελιξία στην αλλαγή των επιχειρησιακών απαιτήσεων, η γρηγορότερη ανταπόκριση σε προβλήματα, απειλές ή ευκαιρίες και η βελτιωμένη συνεργασία μεταξύ όσων συμμετέχουν στις ίδιες διαδικασίες. Τα περισσότερα εμπορικά προϊόντα λογισμικού για οδηγούμενη από γεγονότα διαχείριση διαδικασιών θεωρούνται ανώριμα και μη πλήρη, όπως και τα σχετικά πρότυπα. Χαρακτηρίζονται δε από υψηλή τιμή ενώ απαιτούν αρκετή προσπάθεια προσαρμογής στον εκάστοτε πελάτη – χρήστη. Μερικοί από τους σημαντικότερους παρόχους τέτοιων λύσεων είναι οι Appian, Aurea, Bosch Software Innovations, Fujitsu, IBM, Informatica, Oracle, Pegasystems, Red Hat, SAP, Software AG, Tibco Software, Vitria και Whitestein.

Κατά την εκτίμηση της (Gartner, 2014) η οδηγούμενη από γεγονότα διαχείριση διαδικασιών θα καταστεί μια από τις πιο σημαντικές και διαδεδομένες προσεγγίσεις υλοποίησης ευφυών επιχειρησιακών λύσεων ενώ θα αποτελέσει ένα βασικό κριτήριο επιλογής εμπορικών προϊόντων για ευφυή επιχειρησιακή λειτουργία. Μακροπρόθεσμα θα χρησιμοποιηθεί σε μεγάλα και περίπλοκα έργα διότι μπορεί να υποστηρίξει μια ευρεία γκάμα διαδικασιών από απλές έως πολυσύνθετες.

1.2 Κίνητρο

Όπως αναφέρθηκε παραπάνω, οι διαδικασίες μέσα στο σύγχρονο περιβάλλον δεν μπορεί να είναι στατικές και αμετάβλητες αλλά θα πρέπει να προσαρμόζονται στις εκάστοτε συνθήκες, απαιτήσεις αλλά και ευκαιρίες που εμφανίζονται. Η «προσαρμογή των διαδικασιών» (business process adaptation) επιτυγχάνεται ως επί το πλείστο με ανθρώπινη παρέμβαση, δηλαδή κάποιος κατάλληλος εμπειρογνώμονας (expert) καλείται να σχεδιάσει και υλοποιήσει τις απαραίτητες αλλαγές στις διαδικασίες. Στην «καθημερινότητα» οι διαδικασίες βρίσκονται συχνά αντιμέτωπες με καταστάσεις και συνθήκες που δεν έχουν προβλεφθεί επαρκώς ή δεν είναι δυνατό να προβλεφθούν, ανεξάρτητα από το πόσο αναλυτικά και σωστά είναι σχεδιασμένες. Τέτοιες περιπτώσεις είναι αρκετά συχνές στην πράξη και τις αντιμετωπίζουν συνήθως άνθρωποι (Lu et al. 2014). Οι αλλαγές που

Χρησιμοποιεί δε ένα συνδυασμό ενορχήστρωσης δομημένων διαδικασιών, διαχείρισης επιχειρησιακών κανόνων και επεξεργασίας σύνθετων γεγονότων.

⁴ Ο «κύκλος υπερβολής» (hype cycle) αποτελεί ένα μεθοδολογικό εργαλείο της εταιρείας συμβούλων πληροφορικής Gartner, για την εκτίμηση της ωριμότητας και αποδοχής μιας αναδυόμενης τεχνολογίας ή εφαρμογής. Επίσης αποτυπώνει το πως αυτή θα εξελιχθεί μέσα στο χρόνο, συμβάλλοντας στην αντικειμενικότερη εκτίμησή της, την αποφυγή υπερβολικών προσδοκιών και τη μείωση του κινδύνου που απορρέει από ενδεχόμενη επένδυση σε αυτήν.

[Πηγή: <http://www.gartner.com/technology/research/methodologies/hype-cycle.jsp>]

απαιτούνται είναι πολλές φορές μικρής έκτασης, με σύντομη χρονική διάρκεια ή εμφανίζονται με μεγάλη συχνότητα, έτσι ώστε να μην έχει νόημα ή να μην είναι εφικτός ο επανασχεδιασμός της διαδικασίας για την επίλυσή τους από κάποιον άνθρωπο. Συνήθως τέτοια φαινόμενα αντιμετωπίζονται κατά περίπτωση (ad hoc) από τους άμεσα εμπλεκόμενους στην εκτέλεση των διαδικασιών.

Όμως η ad hoc αντιμετώπιση τέτοιων φαινομένων μπορεί μεσο-μακροπρόθεσμα να αποδειχθεί επιζήμια, αφού τέτοιες παρεκκλίσεις από την κανονική διαδικασία συνήθως δεν τεκμηριώνονται επαρκώς ούτε αξιολογούνται κανονικά. Επίσης δεν μετατρέπονται σε κάποιας μορφής γνώση ή εμπειρία χρήσιμη και για άλλους μέσα στον οργανισμό, όπως για παράδειγμα σε μαθήματα προς μίμηση ή προς αποφυγή. Επιπλέον, οι μηχανογραφημένες και αυτοματοποιημένες επιχειρησιακές διαδικασίες δεν είναι εύκολα επιδεκτικές τέτοιων αλλαγών και παρεκκλίσεων κατά την εκτέλεσή τους. Αυτό δημιουργεί «δυσκαμψία» στην αντιμετώπιση έκτακτων περιπτώσεων και συμβάντων ενώ η αλλαγή μιας μηχανογραφημένης επιχειρησιακής διαδικασίας δεν μπορεί να θεωρηθεί ως τετριμμένη εργασία. Εξάλλου, όπως έχει ήδη τονιστεί αρκετές φορές από παλιά, η ευελιξία και η προσαρμοστικότητα αποτελεί ένα κρίσιμο χαρακτηριστικό των διαδικασιών που παρέχει στις επιχειρήσεις ή οργανισμούς τη δυνατότητα να προσαρμόζονται στις εκάστοτε συνθήκες (Heinl et al., 1999; Regev and Wegmann, 2005).

Τα γεγονότα και ειδικότερα η οδηγούμενη από γεγονότα διαχείριση διαδικασιών, μπορούν να αξιοποιηθούν για την αντιμετώπιση του προαναφερθέντος προβλήματος και έχουν ήδη χρησιμοποιηθεί σε αρκετές εργασίες για προσαρμογή εφαρμογών και διαδικασιών (δείτε στην ενότητα 1.3 και στα κεφάλαια 2 και 3). Μερικοί λόγοι είναι η σε σχεδόν πραγματικό χρόνο λήψη και επεξεργασία των απαραίτητων πληροφοριών, που επιτρέπει την ταχύτερη ανίχνευση και αντιμετώπιση τυχόν προβλημάτων, καθώς και η δυνατότητα συνδυασμού πληροφορίας από πολλαπλές πηγές, που επιτρέπει την καλύτερη κατανόηση των εκάστοτε συνθηκών. Ακόμη η ευελιξία που παρέχουν οι οδηγούμενες από γεγονότα αρχιτεκτονικές (event-driven architecture) προσδίδει πολλά πλεονεκτήματα στις αντίστοιχες υλοποιήσεις.

1.3 Σχετική έρευνα

Για τη συστηματικότερη αντιμετώπιση της ανάγκης προσαρμογής των διαδικασιών έχουν πραγματοποιηθεί αρκετές έρευνες που εξετάζουν το πρόβλημα από πολλές διαφορετικές οπτικές ενώ προτείνουν διάφορες λύσεις και πλαίσια εργασίας. Το σχετικό επιστημονικό πεδίο είναι αυτό της διαχείρισης επιχειρησιακών διαδικασιών (business process management) και ειδικότερα της προσαρμογής αυτών (business process adaptation). Η προσαρμογή (adaptation) ως ιδιότητα των πληροφοριακών συστημάτων έχει ιδιαίτερα απασχολήσει την έρευνα σχετικά με τις υπηρεσίες ιστού (web services), εφαρμογές βασισμένες σε υπηρεσίες, τη σύνθεση υπηρεσιών (service composition) και την υπηρεσιοστρεφή αρχιτεκτονική (Service-Oriented Architecture, SOA).

Ιδιαίτερο ενδιαφέρον παρουσιάζουν οι μέθοδοι που αξιοποιούν γεγονότα⁵ (events) προκειμένου να λαμβάνουν πληροφόρηση για διαδικασίες ή εφαρμογές και το περιβάλλον τους. Αυτό οφείλεται στο ότι η χρήση γεγονότων επιτρέπει την ταχύτερη λήψη, διάχυση, επεξεργασία και διανομή πληροφοριών, σε σχεδόν πραγματικό χρόνο, από την πηγή της πληροφορίας προς τους ενδιαφερόμενους αποδέκτες. Τεχνολογίες που στηρίζονται στη χρήση γεγονότων και της οδηγούμενης από γεγονότα αρχιτεκτονικής (event-driven architecture, EDA) επιτυγχάνουν γρηγορότερη επίγνωση και ανταπόκριση σε αλλαγές των συνθηκών καθώς και κλιμακούμενη δυνατότητα επεξεργασίας μεγάλου όγκου πληροφορίας.

Ενδιαφέρουσα κατά την άποψή μας είναι η εργασία των Schonenberg et al. (2008) για την κατηγοριοποίηση της δυνατότητας ευελιξίας των διαδικασιών σε ευελιξία (α) κατά τον σχεδιασμό (flexibility by design), (β) κατ' απόκλιση (flexibility by deviation), (γ) με ελλειπείς προδιαγραφές (flexibility by underspecification) και (δ) ευελιξία κατά τη μεταβολή (flexibility by change). Επίσης σημαντική θεωρούμε την εργασία που έγινε στο πλαίσιο του Ευρωπαϊκού Δικτύου Αριστείας S-Cube⁶ καθώς και την ταξινόμηση των διαφόρων ειδών προσαρμογής που μπορεί να εφαρμοστούν στις βασισμένες σε υπηρεσίες εφαρμογές (Service-Based Applications ή SBAs) (Bucchiarone et al., 2010; Lanese et al., 2010).

Πολλές διαφορετικές στρατηγικές έχουν προταθεί για δημιουργία ή επιλογή προσαρμογών. Μια τέτοια είναι η αξιοποίηση του πλαισίου για την επιλογή των προσαρμογών (context-driven adaptation). Παραδείγματα μεθόδων και συστημάτων που ακολουθούν αυτή την προσέγγιση έχουν προταθεί από τους Santos et al. (2007), Dai and Liu (2010), Moltchanov et al. (2009) για απ' άκρο εις άκρο σχεδιασμό συστημάτων με επίγνωση του πλαισίου για διανομή υπηρεσιών και περιεχομένου. Παρεμφερείς προσέγγιση είναι αυτή των Catarci et al. (2008) που στηρίζεται στη χρήση μιας οντολογίας για προσαρμογές σε διαδικασίες. Ακόμη, ενδιαφέρον παρουσιάζει και η πρόταση των David and Ledoux (2005) για ένα αυτό-προσαρμοζόμενο σύστημα βασισμένο στη χρήση κανόνων ECA και επίγνωσης του πλαισίου εκτέλεσης.

Μια άλλη στρατηγική για την προσαρμογή εφαρμογών στηρίζεται στη σύνθεση υπηρεσιών με χρήση προσαρμογέων (adaptation adapters), κανόνων προσαρμογών (adaptation rules) κι άλλων τεχνικών. Τέτοιες προσεγγίσεις προτείνονται από τους Küster et al. (2005), van der Aalst et al. (2009), Lanese et al. (2010), Spanoudakis et al. (2005), Spanoudakis and Zisman (2010), Hu et al. (2008). Οι Aydin et al. (2008) απ' την άλλη προτείνουν τη χρήση λογισμού γεγονότων (event calculus) για την αυτόματη δημιουργία και εκτέλεση συνθέσεων υπηρεσιών ιστού.

Ενδιαφέρον παρουσιάζουν και διάφορες προσεγγίσεις που υιοθετούν μεθόδους από το επιστημονικό πεδίο των συστημάτων προτάσεων (recommender systems). Οι πρώτοι που πρότειναν τη χρήση αλγορίθμων από το πεδίο αυτό ήταν οι Sreenath and Singh (2004), για

⁵ Γεγονός (event) είναι μια πληροφορία ή μήνυμα για κάτι που συνέβη ή που άλλαξε την κατάσταση ενός συστήματος ή του περιβάλλοντός αυτού. Δείτε επίσης τον ορισμό του (Mani Chandry, 2006)

⁶ <http://www.s-cube-network.eu/>

επιλογή υπηρεσιών με χρήση συνεργατικής διήθησης (collaborative filtering) και διήθηση βασισμένη στο περιεχόμενο (content-based filtering). Μεθόδους βασισμένες στη βαθμολόγηση υπηρεσιών με χρήση συνεργατικής διήθησης, χρησιμοποίησαν στις εργασίες τους οι Shao et al. (2007), Zheng et al. (2009), Rong et al. (2009), Toma et al. (2009) και Chen et al. (2010). Οι Adomavicius and Tuzhilin (2010) διακρίνουν τις μεθόδους προτάσεων, (1) σε εκείνες που πραγματοποιούν αναζητήσεις υπηρεσιών (ως προτάσεις προσαρμογών) βασισμένες στο πλαίσιο εκτέλεσης (context-sensitive) και (2) σε εκείνες που «εκμαιεύουν» προτιμήσεις των χρηστών από το πλαίσιο εκτέλεσης.

Επίσης ενδιαφέρουσες είναι και διάφορες προσεγγίσεις για προτάσεις προσαρμογών βασισμένων σε εμπειρίες των χρηστών (user experience) ή σε ιστορικά δεδομένα εκτέλεσης (past execution data). Τέτοιες είναι οι εργασίες των Stoitsev et al. (2007), Pesic and van der Aalst (2006), Schonenberg, Weber, et al. (2008) και Dorn et al. (2010) για συστήματα διαχείρισης διαδικασιών (process management systems) ή συστήματα διαχείρισης ροής εργασιών (workflow management systems). Άλλες εναλλακτικές προσεγγίσεις ακολουθούνται από τους Almeida et al. (2004) όπου προτείνεται η χρήση οντολογίας και Vanderfeesten et al. (2008) όπου χρησιμοποιούν μια στρατηγική επιλογής με βάση το ελάχιστο κόστος – τιμή κάποιας παραμέτρου.

Όσον τώρα αφορά τους τρόπους ανίχνευσης της ανάγκης προσαρμογής έχουν προταθεί στη βιβλιογραφία μέθοδοι και τεχνικές που αντιμετωπίζουν το θέμα χρησιμοποιώντας αρκετά διαφορετικές προσεγγίσεις και σε διαφορετικό εύρος. Μια τέτοια προσέγγιση είναι η αξιοποίηση μεθόδων επίγνωσης κατάστασης σύμφωνα με τις εργασίες των Endsley (1995), Endsley and Connors (2008), Jakobson et al. (2007) και Jakobson (2008). Ιδιαίτερο ακόμη ενδιαφέρον παρουσιάζουν προσεγγίσεις επίγνωσης κατάστασης με χρήση γεγονότων (Adi and Etzion, 2004), όπως είναι το σύστημα SARI (Schiefer and Seufert 2005; Schiefer et al. 2007; Schiefer et al. 2009), η προσέγγιση των Berkovsky and Eytani (2005) για εγγραφή σε περιεχόμενο με επίγνωση του περιβάλλοντος (context-aware subscription) καθώς και μια βασισμένη σε γεγονότα υπηρεσιοστρεφής αρχιτεκτονική (event-based SOA) για κινητά τηλέφωνα που πρότειναν οι Hinze, Michel and Eschner (2009).

Μια άλλη προσέγγιση αφορά την παρακολούθηση της εκτέλεσης εφαρμογών για τον εντοπισμό προβλημάτων, όπως είναι τα συστήματα VieDAME (Moser et al. 2008), Dynamo (Baresi, Guinea and Pasquale, 2007), το σύστημα των Erradi et al. (2006) για την παρακολούθηση της εφαρμογής πολιτικών QoS με χρήση κανόνων ECA και η αρχιτεκτονική DySOA (Siljee et al., 2005). Μία ακόμη ενδιαφέρουσα προσέγγιση είναι η χρήση μεθόδων δοκιμών χρόνου εκτέλεσης (online testing) όπως περιγράφεται για παράδειγμα στις εργασίες των Wang et al. (2004), Deussen et al. (2003), Chan et al. (2007) και το πλαίσιο προσαρμοστικών δοκιμών των Bai et al. (2007). Ακόμη έχουν προταθεί και υβριδικές μέθοδοι που συνδυάζουν τεχνικές παρακολούθησης και δοκιμών χρόνου εκτέλεσης όπως των Gehlert et al. (2011).

Από τη βιβλιογραφική έρευνα που πραγματοποιήθηκε διαπιστώθηκε ότι υπάρχει σημαντικός αριθμός εργασιών στα επιστημονικά πεδία που αφορούν τη διατριβή. Ωστόσο είναι λιγότερες εκείνες που παρέχουν μια συνολική θεώρηση και αντιμετώπιση του

προβλήματος που αυτή θέτει, ενώ οι λύσεις που προτείνουν έχουν περιορισμούς ή ακολουθούν συγκεκριμένες συμβάσεις. Διαπιστώνονται κενά στη μεθοδολογική αποτύπωση των καταστάσεων ενδιαφέροντος των διαδικασιών σε σχέση με τους στόχους τους και στην αξιοποίηση των τεχνολογιών που οδηγούν στην επίγνωση των καταστάσεων αυτών. Όσον αφορά τον τρόπο εφαρμογής των προσαρμογών σε διαδικασίες, διαπιστώνεται ότι οι περισσότερες προτάσεις που υπάρχουν στη βιβλιογραφία εστιάζουν κυρίως σε παλαιότερες γλώσσες, όπως η BPEL που εξειδικεύεται σε εφαρμογές υπηρεσιών ιστού, ενώ όσες ασχολούνται με διαδικασίες σε BPMN 2.0, το νέο de facto πρότυπο για αποτύπωση διαδικασιών, δεν είναι αρκετά ώριμες.

1.4 Εύρος της διατριβής και Ερευνητικός Στόχος

Βασικός στόχος της παρούσας διατριβής είναι η διερεύνηση και η πρόταση μιας λύσης στο πρόβλημα της οδηγούμενης από γεγονότα προσαρμογής επιχειρησιακών διαδικασιών. Φυσικά το ερώτημα αυτό έχει ήδη απασχολήσει την αντίστοιχη ερευνητική κοινότητα. Σε αντίθεση όμως με πολλές εργασίες στη βιβλιογραφία, η θέση μας είναι ότι η προσαρμογή πρέπει να οδηγείται από την «επίγνωση της κατάστασης» των διαδικασιών κι όχι απευθείας από γεγονότα (είτε είναι πρωτογενή είτε σύνθετα). Τα γεγονότα που λαμβάνονται πρέπει να αξιοποιούνται και να ερμηνεύονται σύμφωνα πάντα με κατάλληλα μοντέλα επίγνωσης κατάστασης. Ο λόγος είναι ότι η επίγνωση της κατάστασης επιτρέπει την κατανόηση των αιτιών που μια διαδικασία βρίσκεται αντιμέτωπη με συγκεκριμένες συνθήκες (τις οποίες καταγράφουν τα γεγονότα). Γνωρίζοντας ή μπορώντας να αντιληφθούμε τα αίτια, αφενός μας επιτρέπει να ερμηνεύουμε ρεαλιστικότερα και ορθότερα τα λαμβανόμενα γεγονότα. Αφετέρου καθίσταται πιο αποτελεσματική ολόκληρη η διαδικασία προσαρμογής, αφού πλέον μπορούν να αναγνωριστούν οι καταστάσεις και οι συνθήκες εκείνες που αποτελούν πρόβλημα και οδηγούν σε πραγματική ανάγκη για προσαρμογή. Ακόμη, η επιλογή των κατάλληλων προσαρμογών καθώς και η εφαρμογή τους στις διαδικασίες όπου χρειάζεται, επίσης μπορούν να «οδηγηθούν» από την επίγνωση της κατάστασης.

Στην παρούσα διατριβή εστιάζουμε το ενδιαφέρον μας σε συγκεκριμένες παραμέτρους του προβλήματος. Ειδικότερα, όσον αφορά τη φύση των διαδικασιών, ενδιαφερόμαστε για μηχανογραφημένες επιχειρησιακές διαδικασίες που αποτυπώνονται και εκτελούνται με τη βοήθεια της σημειογραφίας BPMN 2.0. Επιπλέον εστιάζουμε σε διαδικασίες όπου υπάρχει η δυνατότητα λήψης πληροφόρησης για την κατάστασή τους (έμμεσα ή άμεσα) με τη μορφή γεγονότων.

Όσον αφορά τις προσαρμογές το ενδιαφέρον μας εντοπίζεται σε διορθωτικές προσαρμογές (που αντιμετωπίζουν ένα υφιστάμενο ή επερχόμενο πρόβλημα) κι όχι τόσο σε βελτιωτικές (που αποσκοπούν στη βελτίωση των παραμέτρων εκτέλεσης των διαδικασιών). Ακόμη οριοθετούμε την έρευνά μας στην επιλογή προσαρμογών από ένα σύνολο διαθέσιμων εναλλακτικών ενώ δεν ασχολούμαστε με τη σύνθεση ή δημιουργία νέων. Τέλος επιδιώκουμε μια προσέγγιση που θα επιτρέπει την εφαρμογή των προσαρμογών στις διαδικασίες χωρίς να απαιτείται μια εκ των προτέρων προετοιμασία αυτών.

1.5 Προτεινόμενη προσέγγιση

Στη συνέχεια της διατριβής προτείνεται ένα πλαίσιο εργασίας που περιλαμβάνει μεθοδολογικά εργαλεία με στόχο (α) την αποτύπωση των καταστάσεων ενδιαφέροντος των επιχειρησιακών διαδικασιών, σε συνάρτηση με τους στόχους τους και σε συνδυασμό με γεγονότα που οδηγούν στην επίγνωσή τους. (β) Τη δυναμική και ευέλικτη αναζήτηση και επιλογή προσαρμογών από ένα σύνολο εναλλακτικών, που σχετίζονται με τις καταστάσεις των διαδικασιών που ανιχνεύονται. (γ) Την εφαρμογή των προσαρμογών στις διαδικασίες, δηλαδή την κατάλληλη τροποποίηση αυτών. Επίσης έχει αναπτυχθεί και ένα σύνολο εργαλείων λογισμικού που υλοποιούν το προτεινόμενο πλαίσιο εργασίας.

Η προσαρμογή των διαδικασιών είναι μια επαναλαμβανόμενη διεργασία ανίχνευσης τυχόν προβλημάτων ή αποκλίσεων ή άλλων συναφών «περιπτώσεων ενδιαφέροντος», εκτίμηση της ανάγκης για προσαρμογή, επιλογή μιας προσαρμογής από ένα σύνολο εναλλακτικών και πρόταση αυτής στον υπεύθυνο της διαδικασίας, και τέλος εφαρμογή αυτής με χρήση μιας νέας τεχνικής θεματοστρεφούς προγραμματισμού. Τα παραπάνω καλύπτουν το μεγαλύτερο μέρος του κύκλου ζωής μιας προσαρμογής, από την ανίχνευση καταστάσεων ενδιαφέροντος (εδώ προβλημάτων) έως την τροποποίηση της διαδικασίας (δείτε πιο αναλυτικά στο Κεφάλαιο 4 «Πρόταση της Διατριβής»).

Όσον αφορά τον στόχο (α) προτείνουμε μια καινοτόμα μέθοδο για επίγνωση της κατάστασης των διαδικασιών με αξιοποίηση γεγονότων (events), η οποία λαμβάνει υπόψη της τους στόχους και εν μέρει τη δομή των διαδικασιών. Αυτή είναι τα Δίκτυα Κατάστασης – Ενέργειας (ΔΚΕ) ή Situation – Action Networks (SANs), τα οποία προσομοιάζουν με δένδρα συμπεριφορών (δείτε ενότητα 5.1.4). Κάθε SAN αποτυπώνει έναν στόχο μιας διαδικασίας και επίσης την ανάλυσή τους σε υποστόχους. Κάθε στόχος – υποστόχος συσχετίζεται με ένα γεγονός (event) που σηματοδοτεί την εμφάνιση μιας κατάστασης όπου υπάρχει απόκλιση από αυτόν ή απαιτούνται ενέργειες προκειμένου αυτός να (επαν-)επιτευχθεί. Κάθε φορά που λαμβάνεται ένα τέτοιο γεγονός μπορούν να εκτελεστούν ορισμένες ενέργειες ως ανταπόκριση σε αυτό. Επίσης αν ο στόχος έχει υποστόχους τότε μπορεί να ζητηθεί επιπλέον η επίτευξη ή η εξέταση της επίτευξης των υποστόχων και ούτω καθεξής.

Οι ενέργειες των SANs αποσκοπούν εν τέλη στη διαπίστωση αλλαγής της κατάστασης μιας διαδικασίας (άρα επίγνωσης αυτής της αλλαγής) καθώς και στην επιλογή μιας αντίστοιχης προσαρμογής εφόσον αυτό απαιτείται. Η ανταπόκριση σε ένα γεγονός, δηλαδή η εκτέλεση ορισμένων ενεργειών ή η εξέταση της επίτευξης των υποστόχων, μπορεί είτε να προκαθορίζεται από τον εμπειρογνώμονα κατά τον σχεδιασμό της προσαρμογής, είτε να επιλέγεται δυναμικά κατά την εκτέλεση της διαδικασίας από ένα σύνολο εναλλακτικών, σύμφωνα με μια προκαθορισμένη μέθοδο. Η περίπτωση της δυναμικής επιλογής απαντά στο (β) στόχο του πλαισίου εργασίας. Ο τρόπος επιλογής που προτείνεται είναι με χρήση συνόλων εναλλακτικών προσαρμογών που καλούνται «δεξαμενές ενεργειών». Κάθε μία εναλλακτική προσαρμογή είναι κατάλληλη για χρήση κάτω από συγκεκριμένες (διαφορετικές) συνθήκες και καταστάσεις. Τα SANs επεκτείνονται με μια ειδική ενέργεια που καλείται «αφηρημένη ενέργεια» (abstract action) όπου κατά την εκτέλεσή της γίνεται η αναζήτηση μέσα σε μια δεξαμενή ενεργειών για τον εντοπισμό της εκάστοτε

καταλληλότερης εναλλακτικής. Τόσο η δεξαμενή ενεργειών, όσο και ο τρόπος αναζήτησης και επιλογής της εναλλακτικής και οι παράμετροι αυτού, προκαθορίζονται για κάθε μία αφηρημένη ενέργεια είτε κατά τον σχεδιασμό του SAN είτε κάποια στιγμή πριν την εκτέλεσή του. Οι αφηρημένες ενέργειες προσθέτουν σημαντική ευελιξία στα SANs.

Τέλος, ο (γ) στόχος αφορά την εφαρμογή των προσαρμογών σε μια διαδικασία, δηλαδή την κατάλληλη τροποποίηση αυτής σύμφωνα με τις προδιαγραφές της προσαρμογής που επιλέγεται. Το προτεινόμενο πλαίσιο εργασίας περιλαμβάνει μια καινοτόμα τεχνική προσαρμογής που στηρίζεται στις αρχές και μεθόδους του θεματοστρεφούς προγραμματισμού (aspect-oriented programming). Η τεχνική αυτή καλείται FlexiBPMN2.0 και επεκτείνει προγενέστερη εργασία που έχει προταθεί στη βιβλιογραφία, την AO4BPMN. Αναλυτικότερα, κάθε προσαρμογή προδιαγράφει ένα ή περισσότερα σημεία ή τμήματα (μιας διαδικασίας) που τροποποιεί καθώς και (προαιρετικά) ποιες συνθήκες πρέπει να ισχύουν για την εφαρμογή αυτής. Τα σημεία ή τμήματα αυτά μαζί με τις αντίστοιχες συνθήκες εφαρμογής καλούνται σημεία τομής (pointcuts). Εφόσον διαπιστωθεί ότι μια προσαρμογή πρέπει να εφαρμοστεί, τότε το αντίστοιχο σημείο τομής αντικαθίσταται πλήρως από ένα άλλο (εναλλακτικό) τμήμα διαδικασίας, το οποίο εκτελείται στη θέση του. Αυτό καλείται «συμβουλή» (advice). Μια συμβουλή μπορεί αν απαιτηθεί να εκτελέσει και το αρχικό σημείο τομής ως μέρος της. Οι όροι σημεία τομής και συμβουλές είναι ορολογία δανεισμένη από τον θεματοστρεφή προγραμματισμό ενώ ο συνδυασμός σημείων τομής και της συμβουλής που τα αντικαθιστά καλείται «θέμα» (aspect). Στην προτεινόμενη τεχνική FlexiBPMN2.0 ένα θέμα αποτυπώνει μια προσαρμογή μιας διαδικασίας αφού προσδιορίζει τόσο τα σημεία ή τμήματα που τροποποιούνται, τις συνθήκες εφαρμογής τους, όσο και το νέο τμήμα που τα αντικαθιστά.

1.6 Καινοτομίες – Συνεισφορά

Τα SANs είναι μια καινοτομία της διατριβής, η οποία συνδυάζει τα δένδρα συμπεριφορών (που χρησιμοποιούνται κυρίως στην τεχνητή νοημοσύνη παιχνιδιών Η/Υ) με ιδέες από τους κανόνες ECA (Dittrich et al., 1995) για την επίτευξη και αποτύπωση της οδηγούμενης από γεγονότα επίγνωσης της κατάστασης διαδικασιών. Τα SANs προδιαγράφουν τον τρόπο λήψης γεγονότων από τις αντίστοιχες πηγές, καθώς και τον τρόπο εγγραφής και διαγραφής από αυτές. Προσδιορίζουν ακόμη τις ενέργειες που πρέπει να εκτελεστούν ως ανταπόκριση στη λήψη ενός γεγονότος. Επίσης περιλαμβάνουν και ένα βήμα ελέγχου συνθηκών του περιβάλλοντος εκτέλεσης, το οποίο λειτουργεί ως φίλτρο προαιρετικής ή υπό συνθήκη εκτέλεσης. Τα SANs μπορούν να χρησιμοποιηθούν ως μοντέλα για την αποτύπωση του τρόπου επίγνωσης κατάστασης των διαδικασιών από γεγονότα, τα οποία (μοντέλα) λαμβάνουν υπόψη τους στόχους και τη δομή των διαδικασιών. Μπορούν όμως να χρησιμοποιηθούν και για την σε πραγματικό χρόνο επίγνωση της κατάστασής τους και αντίδρασης στις αλλαγές των καταστάσεων αυτών. Στο πλαίσιο της παρούσας διατριβής ενδιαφερόμαστε για την ανίχνευση καταστάσεων που δημιουργούν την ανάγκη για προσαρμογή καθώς και την επιλογή κατάλληλων προσαρμογών για την αντιμετώπιση των παραπάνω καταστάσεων.

Η επόμενη καινοτομία της διατριβής εντοπίζεται στον τρόπο επιλογής των προσαρμογών από ένα σύνολο εναλλακτικών. Συγκεκριμένα οι δεξαμενές ενεργειών αποτελούν ένα πλαίσιο εργασίας για επιλογή εναλλακτικών, το οποίο είναι ανεξάρτητο της μεθόδου επιλογής (και των παραμέτρων της) αλλά και της φύσης των ίδιων των εναλλακτικών. Το πλαίσιο αυτό προβλέπει τη χρήση αφηρημένων ενεργειών (abstract actions) στα SANS, όπου σε κάθε μία από αυτές θα πρέπει να προσδιορίζεται (πριν την εκτέλεσή της) η δεξαμενή ενεργειών (ως σύνολο εναλλακτικών), η μέθοδος αναζήτησης και επιλογής και οι παράμετροί της καθώς και ένας τρόπος επίλυσης (resolution policy) για την περίπτωση που επιλεγούν περισσότερες από μία εναλλακτικές. Στη συγκεκριμένη διατριβή περιγράφονται ενδεικτικά δύο πολυκριτηριακές μέθοδοι αναζήτησης και επιλογής (SMART και LOWA), όπου η μεν πρώτη κάνει χρήση αριθμητικών δεδομένων ενώ η δεύτερη χρησιμοποιεί γλωσσικούς όρους, προκειμένου να εκτιμήσουν την καταλληλότητα μιας εναλλακτικής.

Η τρίτη καινοτομία αφορά σε μία νέα μέθοδο, την FlexiBPMN2.0, για την αποτύπωση και εφαρμογή προσαρμογών σε διαδικασίες και την τροποποίησή τους. Η καινοτομία εστιάζεται στην επέκταση μιας προϋπάρχουσας εργασίας (AO4BPMN) έτσι ώστε να μπορεί να εφαρμόζεται σε διαδικασίες σε BPMN 2.0. Ακόμη περιγράφεται ο τρόπος αποτύπωσης και η δομή μιας προσαρμογής καθώς και η λογική αρχιτεκτονική του αντίστοιχου προσαρμοστικού συστήματος εκτέλεσης. Το σύστημα αυτό προβλέπει την εκπομπή γεγονότων σε κάθε βήμα της εκτέλεσης μιας διαδικασίας αλλά και τη λήψη γεγονότων που ενημερώνουν για την επιλογή μιας προσαρμογής, την οποία στη συνέχεια εφαρμόζει στις εκτελούμενες διαδικασίες.

Τέλος αναφέρουμε ότι καινοτόμα θεωρούμε και τη συνολική προσέγγιση του προβλήματος αφού προτείνει ένα νέο συνδυασμό μεθόδων για αποτύπωση και επίτευξη οδηγούμενης από γεγονότα επίγνωσης καταστάσεων (SANS), επιλογή προσαρμογών με ένα παραμετρικό πλαίσιο εργασίας (δεξαμενές ενεργειών – αφηρημένες ενέργειες SANS) και εφαρμογή αυτών με χρήση μια νέας θεματοστρεφούς τεχνικής (FlexiBPMN2.0). Τα παραπάνω είναι τα μεθοδολογικά εργαλεία ενός πλαισίου εργασίας για την προσαρμογή διαδικασιών, το οποίο καλύπτει το μεγαλύτερο μέρος του κύκλου ζωής προσαρμογών (δείτε ενότητα 4.2.1).

Σημειώνουμε ότι και ο κύκλος ζωής των προσαρμογών αποτελεί μια πρωτοτυπία της παρούσας διατριβής, (α) ως μια εξειδίκευση του κύκλου OODA (Greene, 2007; Osinga, 2007; Richards, 2004) στο πεδίο των προσαρμοστικών εφαρμογών ή του «κύκλου ζωής των προσαρμοστικών εφαρμογών βασισμένων σε υπηρεσίες» (Bucchiarone et al., 2010) ή ακόμη του «κύκλου ζωής διαχείρισης επιχειρησιακών διαδικασιών» (ter Hofstede et al., 2010). (β) Επειδή προβλέπει ρητά την πρόταση των προσαρμογών που επιλέγονται στον υπεύθυνο της διαδικασίας για έγκριση πριν την εφαρμογή αυτών, αναγνωρίζοντας έτσι τη σημασία του βήματος αυτού. Οι υλοποιήσεις των παραπάνω μεθοδολογικών εργαλείων, μαζί με λογισμικό για την εγγραφή και λήψη γεγονότων και την επεξεργασία σύνθετων γεγονότων, απαρτίζουν την πλατφόρμα προτάσεως προσαρμογών, η οποία υλοποιεί το προτεινόμενο πλαίσιο εργασίας.

1.7 Δομή διατριβής

Η υπόλοιπη διατριβή έχει δομηθεί ως ακολούθως. Το πρώτο κεφάλαιο είναι η παρούσα εισαγωγή. Στα επόμενα δύο κεφάλαια παρουσιάζονται τα ευρήματα βιβλιογραφικής έρευνας που πραγματοποιήθηκε σε επιστημονικά πεδία σχετικά με την οδηγούμενη από γεγονότα επίγνωση κατάστασης (κεφάλαιο 2) και την προσαρμογή επιχειρησιακών διαδικασιών (κεφάλαιο 3). Στο κεφάλαιο 4 επιχειρείται η διαμόρφωση του προβλήματος και η διατύπωση των σχετικών ερευνητικών ερωτημάτων που απαντώνται από τη διατριβή. Στο κεφάλαιο 5 παρουσιάζεται η πρώτη καινοτομία, δηλαδή τα Δίκτυα Κατάστασης – Ενέργειας και επεξηγείται η μέθοδος, η σημειογραφία της και τα μέρη αυτής. Ο μηχανισμός εκτέλεσης των SANs παρουσιάζεται στο κεφάλαιο 6 ενώ μια αξιολόγηση αυτού δίνεται στο κεφάλαιο 7. Στη συνέχεια παρουσιάζεται η μέθοδος FlexiBPMN2.0 για θεματοστρεφή προσαρμογή διαδικασιών αποτυπωμένων σε BPMN 2 καθώς και το σύστημα προσαρμοστικής εκτέλεσης διαδικασιών. Αυτή είναι η δεύτερη καινοτομία της διατριβής. Στο κεφάλαιο 9 περιγράφεται η επέκταση των SANs με τις «δεξαμενές ενεργειών» και τις «αφηρημένες ενέργειες» (τρίτης καινοτομίας της διατριβής) και δίνονται δύο ενδεικτικές πολυκριτηριακές μέθοδοι αναζήτησης και επιλογής προσαρμογών. Στο κεφάλαιο 10 επιχειρείται μια αξιολόγηση της υλοποίησης των δεξαμενών ενεργειών στο πλαίσιο της υλοποίησης των SANs. Το κεφάλαιο 11 αποτελεί τον επίλογο της διατριβής όπου παρουσιάζονται τα συμπεράσματα αυτής και επισημαίνονται οι δυνατότητες για περαιτέρω έρευνα. Η διατριβή ολοκληρώνεται με τον κατάλογο των δημοσιεύσεων και ανακοινώσεων που πραγματοποιήθηκαν στο πλαίσιο της εκπόνησής της. Ακόμη παρατίθενται οι μεταφράσεις των όρους που χρησιμοποιήθηκαν συχνότερα ή που θεωρούνται σημαντικοί καθώς και ένας πλήρης κατάλογος με τη βιβλιογραφία που χρησιμοποιήθηκε. Τέλος συμπεριλαμβάνονται δύο παραρτήματα, δύο με παραδείγματα που αποσαφηνίζουν τη χρήση των SANs και των δεξαμενών ενεργειών και ένα με την προγραμματιστική διεπαφή (API) της υλοποίησης του μηχανισμού των SANs σε Java.

1.8 Σχέση με τις δημοσιεύσεις

Μεγάλο μέρος της έρευνας που διεξήχθη στο πλαίσιο της διατριβής έχει παρουσιαστεί σε επιστημονικά συνέδρια ή δημοσιευτεί στον επιστημονικό τύπο. Στη συνέχεια δίνεται μια αντιστοίχιση μεταξύ των δημοσιεύσεων και της έρευνας όπως αυτή παρουσιάζεται στα κεφάλαια της διατριβής. Ο πλήρης κατάλογος των δημοσιεύσεων και ανακοινώσεων του Ι. Πατινωτάκη δίνεται στο τέλος της διατριβής, στο παράρτημα «Δημοσιεύσεις και Ανακοινώσεις».

- Τα SANs που παρουσιάζονται στο κεφάλαιο 5, περιγράφονται στα [C9] “A Goal Driven Dynamic Event Subscription Approach” και [C1] “Context Management in Event Marketplaces” ενώ αξιολόγησή τους (κεφάλαιο 7) έχει δημοσιευτεί στο [J1] “Dynamic Event Subscriptions in Distributed Event Based Architectures”.

- Η επέκταση των SANs με τις δεξαμενές ενεργειών και τις αφηρημένες ενέργειες που παρουσιάζονται στο κεφάλαιο 9, έχουν δημοσιευτεί στα [C10] “Service Adaptation Recommender in the Event Marketplace: Conceptual View”, [C6] “Collaborative Process Flexibility Using Multi-Criteria Decision Making” και [C5] “A Framework for Situation-Aware Adaptation of Service-Based Applications”.
- Τέλος η μέθοδος FlexiBPMN2.0 που εισάγεται στο κεφάλαιο 8, έχει δημοσιευτεί στο [C7] “An Aspect Oriented Approach for Implementing Situational Driven Adaptation of BPMN2.0 Workflows”.

1.9 Σχέση με Ερευνητικά Έργα

Η παρούσα διδακτορική διατριβή έχει εν μέρει υποστηριχθεί από την Ευρωπαϊκή Επιτροπή μέσω του ερευνητικού έργου των Τεχνολογιών της Πληροφορίας και Επικοινωνίας (Information and Communication Technologies) PLAY⁷ (Pushing dynamic and ubiquitous interaction between services Leveraged in the Future Internet by Applying complex event processing, ICT-FP7- 258659) και σε μικρότερο βαθμό από το έργο των Τεχνολογιών της Κοινωνίας της Πληροφορίας (Information Society Technologies) Laboranova⁸ (Collaboration Environment for Strategic Innovation, IST-FP6-35262).

Έργο PLAY

Το όραμα του PLAY είναι να αλλάξει ριζικά το Διαδίκτυο του μέλλοντος (Future Internet) καθιστώντας το ικανό να έχει επίγνωση των καταστάσεων του περιβάλλοντος, διαμέσου οδηγούμενων από γεγονότα υπηρεσιών, ικανών να προσαρμόζονται στις αλλαγές αυτού. Στόχος του ήταν η ανάπτυξη και δοκιμή μιας αξιόπιστης, ελαστικής, υπηρεσιοστρεφούς, «ομόσπονδης» αρχιτεκτονικής (elastic, federated SOA) για οδηγούμενη από γεγονότα αλληλεπίδραση, μέσα σε κατανεμημένα και ετερογενή συστήματα και περιβάλλοντα. Μια τέτοια αρχιτεκτονική επιτρέπει την ανταλλαγή πληροφοριών πλαισίου (contextual) μεταξύ ετερογενών υπηρεσιών, παρέχοντας έτσι τη δυνατότητα για βελτιστοποιημένη και προσωποποιημένη εκτέλεση αυτών, επιτυγχάνοντας έτσι την οδηγούμενη από τις καταστάσεις προσαρμοστικότητα (situation-driven adaptivity).

Μεγάλο μέρος της έρευνας που περιλαμβάνεται στην παρούσα διατριβή αναπτύχθηκε στο πλαίσιο του PLAY. Το μέρος αυτό περιλαμβάνει την ανάπτυξη μεθόδων για οδηγούμενη από γεγονότα επίγνωση καταστάσεων καθώς και την υλοποίηση αντίστοιχων εργαλείων λογισμικού. Επίσης περιλαμβάνει την ανάπτυξη μεθόδων και υλοποίηση εργαλείων για πρόταση προσαρμογών που αφορούν εκτελούμενες επιχειρησιακές διαδικασίες.

⁷ <http://www.play-project.eu/>

⁸ <http://www.laboranova.com/>

Έργο Laboranova

Ο κύριος στόχος του Laboranova ήταν να βοηθήσει τους «εργάτες της γνώσης» ώστε με ένα συστηματικό τρόπο να μοιράζονται, βελτιώνουν και αξιολογούν ιδέες μεταξύ ομάδων, επιχειρήσεων και δικτύων, αυξάνοντας έτσι την καινοτομική παραγωγή των μονάδων αυτών. Η καινοτομία, η παραγωγή ιδεών και η αξιολόγησή τους είναι στον πυρήνα του έργου. Στο πλαίσιο του Laboranova είχαμε την ευκαιρία να συνεργαστούμε με αρκετά πανεπιστήμια όπως το Polytechnic University of Catalonia στην Ισπανία και το University of Nottingham στο Ηνωμένο Βασίλειο και με σχολές όπως το ESADE στην Ισπανία και το INSEAD στη Γαλλία.

Ένα μέρος της έρευνας που περιλαμβάνεται στην παρούσα διατριβή αναπτύχθηκε στο πλαίσιο του Laboranova. Το μέρος αυτό περιλαμβάνει το σχεδιασμό και υλοποίηση των αλγορίθμων υπολογισμού του σταθμισμένου μέσου όρου κριτηρίων τόσο αριθμητικής όσο και γλωσσικής μορφής (μέθοδοι SMART και LOWA αντίστοιχα) (ενότητα 9.4).

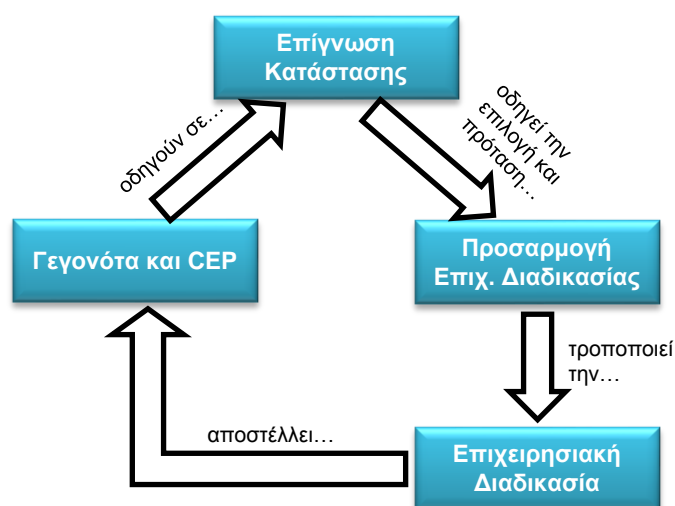
2 Επίγνωση Κατάστασης Οδηγούμενη από Γεγονότα

Στα δύο επόμενα κεφάλαια (2 και 3) παρουσιάζονται συνοπτικά αρκετές εργασίες που έχουν προταθεί στη βιβλιογραφία, οι οποίες άπτονται θεμάτων σχετικών με το αντικείμενο της διατριβής ή προτείνουν ενδιαφέρουσες λύσεις και μεθόδους στο πρόβλημα που αυτή πραγματεύεται. Για τη συστηματικότερη και πληρέστερη αναζήτηση στη βιβλιογραφία χρησιμοποιήθηκε το πλαίσιο έρευνας που περιγράφεται στην επόμενη ενότητα. Στο παρόν κεφάλαιο περιλαμβάνονται εργασίες που αφορούν την επίτευξη επίγνωσης κατάστασης και πλαισίου (situation and context awareness) ενώ στο επόμενο παρουσιάζονται έρευνες σχετικές με την προσαρμογή συστημάτων και διαδικασιών.

Αναλυτικότερα, οι εργασίες που συμπεριλαμβάνονται στο κεφάλαιο αυτό, αφορούν κυρίως στην αναγνώριση και κατανόηση των συνθηκών του περιβάλλοντος, βήμα απαραίτητο προκειμένου να καταστεί δυνατή η ανίχνευση των αναγκών προσαρμογής των επιχειρησιακών διαδικασιών και εφαρμογών. Κατ' αρχήν δίνεται μια εισαγωγή στην επεξεργασία σύνθετων γεγονότων (complex event processing). Κατόπιν παρουσιάζονται η «επίγνωση πλαισίου» και η «επίγνωση κατάστασης» ως ικανότητες συστημάτων και διαδικασιών, και γίνεται μνεία σε σχετικές εργασίες της βιβλιογραφίας. Ιδιαίτερη αναφορά γίνεται σε τεχνικές εκτίμησης και αναγνώρισης των καταστάσεων. Τέλος, παρουσιάζονται ορισμένες καινοτόμες προσεγγίσεις για οδηγούμενη από γεγονότα επίγνωση καταστάσεων και αντίδραση σε αυτές.

2.1 Πλαίσιο της βιβλιογραφικής έρευνας

Όπως αναφέρθηκε ήδη από την εισαγωγή και περιγράφεται διεξοδικότερα στο κεφάλαιο 4, βασικές έννοιες που αφορούν τη διατριβή είναι αυτές (α) του γεγονότος και της επεξεργασίας γεγονότων, (β) της ικανότητας για επίγνωση της κατάστασης συστημάτων ή επιχειρησιακών διαδικασιών καθώς και (γ) της προσαρμογής διαδικασιών. Οι δε σχέσεις μεταξύ των εννοιών αυτών, όπως νοούνται στο πλαίσιο της διατριβής, απεικονίζονται στο ακόλουθο διάγραμμα.



Σχήμα 2-1. Πλαίσιο της βιβλιογραφικής έρευνας

Δηλαδή με βάση τα γεγονότα που αφορούν ένα σύστημα ή μια επιχειρησιακή διαδικασία επιτυγχάνεται η επιθυμητή επίγνωση της κατάστασης στην οποία βρίσκονται. Με βάση την κατάσταση αυτή και εφόσον κριθεί αναγκαίο, επιλέγονται και προτείνονται κατάλληλες προσαρμογές. Οι προσαρμογές εφαρμόζονται στη συνέχεια στις διαδικασίες προκειμένου να τις τροποποιήσουν. Τέλος, οι διαδικασίες, κατά την εκτέλεσή τους, αποστέλλουν γεγο-

νότα που μπορούν να αξιοποιηθούν για τη συνέχιση της επίγνωσης κατάστασης (ως βρόχος ανατροφοδότησης).

Όλες οι παραπάνω έννοιες – στοιχεία αποτελούν αντικείμενο της βιβλιογραφικής έρευνας της διατριβής. Συγκεκριμένα, τα επιστημονικά πεδία που μας ενδιαφέρουν είναι αυτά της επεξεργασίας γεγονότων και ειδικότερα της σύνθετης επεξεργασίας γεγονότων (CEP), της επίγνωσης κατάστασης και πλαισίου, και της προσαρμογής επιχειρησιακών διαδικασιών. Ιδιαίτερο ενδιαφέρον παρουσιάζουν οι εργασίες που άπτονται περισσότερων του ενός πεδία. Οι περιπτώσεις αυτές υπονοούνται από τα βέλη (σχέσεις) του παραπάνω σχήματος. Αναλυτικότερα, αυτά είναι η επίγνωση κατάστασης από γεγονότα ή CEP, η οδηγούμενη από την (επίγνωση) κατάσταση προσαρμογή, καθώς και η εφαρμογή προσαρμογών στις επιχειρησιακές διαδικασίες (ειδικότερα των διαδικασιών σε BPMN 2). Εξυπακούεται ότι τα παραπάνω δίνουν τις γενικές κατευθύνσεις της έρευνας και σε καμία περίπτωση δεν είναι περιοριστικά.

2.2 Επεξεργασία Σύνθετων Γεγονότων

Στην πληροφορική ένα **γεγονός (event)** μπορεί να οριστεί ως μια αμετάβλητη καταγραφή (immutable record) ενός συμβάντος, όπως μια ενέργεια που εκτελέστηκε ή μια αλλαγή της κατάστασης του περιβάλλοντος⁹, η οποία έχει νόημα ή ενδιαφέρει κάποια εφαρμογή. Αυτή μπορεί να αφορά μια επιχειρησιακή ή λειτουργική οντότητα, μια δραστηριότητα (business / operational activity / entity) αλλά μπορεί να είναι και ένα τεχνικής φύσεως μήνυμα όπως κάποιο σφάλμα της υποδομής (infrastructure fault) ή αλλαγή στην διαθεσιμότητα των πόρων.

⁹ An event is an immutable record of a past occurrence of an action or state change. Event properties capture the state information for an event. [Πηγή: http://www.espertech.com/esper/release-5.2.0/esper-reference/html/event_representation.html]

Πρακτικά ένα γεγονός είναι ένα μήνυμα που περιγράφει ένα πραγματικό συμβάν, το οποίο καταγράφηκε από έναν αισθητήρα. Οι διάφορες παράμετροί του περιέχουν πληροφορίες σχετικές με το συμβάν αυτό. Αναλόγως με την εφαρμογή τα γεγονότα μπορούν να αναπαρασταθούν σε μορφές (formats) όπως απλό κείμενο, XML ή RDF, και να μεταδοθούν μέσω διαφόρων καναλιών, όπως μέσω δικτύου, SMS ή email. Τυπικά, οι πληροφορίες που περιέχονται σε ένα γεγονός είναι μόνο αυτές που απαιτούνται για να καταγραφεί μια μεταβολή κατάστασης κι όχι το σύνολο των πληροφοριών της σχετικής οντότητας.

Τα γεγονότα διακρίνονται σε διάφορους τύπους ανάλογα με το είδος των πληροφοριών που περιέχουν και των συμβάντων που καταγράφουν. Μια ακολουθία από διαδοχικά γεγονότα καλείται **ροή γεγονότων (event flow ή event stream)**. Γεγονότα που ανήκουν στην ίδια ροή είναι συνήθως ίδιου τύπου, διαθέτουν κοινή σημασιολογία, κοινό τρόπο αναπαράστασης και μετάδοσης. Η επεξεργασία ροών γεγονότων (Event Stream Processing) είναι μια μέθοδος ανάλυσης ροών γεγονότων και η εξαγωγή συμπερασμάτων από αυτές.

Η **Επεξεργασία Σύνθετων Γεγονότων ή Complex Event Processing (CEP)**, απ' την άλλη, πρόκειται για μια τεχνολογία που επιτρέπει τον συνδυασμό γεγονότων από πολλαπλές ροές και εντοπίζει μέσα σε αυτά μοτίβα που υποδηλώνουν καταστάσεις συνθετότερες από εκείνες των αρχικών γεγονότων. Στόχος της είναι να αναγνωρίζει τα μοτίβα αυτά και να αντιδρά το συντομότερο δυνατό. Δύο βασικές έννοιες της επεξεργασίας σύνθετων γεγονότων είναι αυτές της ομαδοποίησης (aggregation) των απαραίτητων απλών γεγονότων, και της αιτιότητας (causality) όπου η εμφάνιση ενός γεγονότος εξαρτάται από την εμφάνιση κάποιου άλλου (Luckham, 2001).

Τυπικά, τα πρωτογενή γεγονότα μεταφέρουν χαμηλού επιπέδου πληροφορία, όπως για παράδειγμα μετρήσεις και καταγραφές συμβάντων από αισθητήρες. Αντίθετα, τα σύνθετα γεγονότα εκφράζουν καταστάσεις που χαρακτηρίζονται από πολλαπλές παραμέτρους και οι οποίες αντιστοιχούν σε συνδυασμούς πολλών απλών γεγονότων. Τα σύνθετα γεγονότα έχουν συνήθως υψηλότερη προστιθέμενη αξία για τις εφαρμογές και τους καταναλωτές τους απ' ότι τα πρωτογενή γεγονότα από τα οποία προέρχονται. Άρα μπορούμε να πούμε ότι ανήκουν σε υψηλό σημασιολογικό επίπεδο απ' ότι τα πρωτογενή γεγονότα.

Για την αναγνώριση σύνθετων γεγονότων από τις ροές γεγονότων χρησιμοποιούνται διάφορες τεχνικές επεξεργασίας (Etzion and Niblett, 2010; Sybase, 2010). Ένας τρόπος για να επιτευχθεί αυτό είναι με αναζήτηση στις ροές εκ των προτέρων γνωστών μοτίβων γεγονότων που καλούνται μοτίβα ή «πρότυπα σύνθετων γεγονότων» (complex event patterns, CEPats). Το δε εξειδικευμένο λογισμικό που χρησιμοποιείται για την ανίχνευση των προτύπων μέσα στις ροές γεγονότων καλείται μηχανισμός επεξεργασίας σύνθετων γεγονότων (CEP Engine).

Η επεξεργασία σύνθετων γεγονότων (CEP) έχει εφαρμοστεί για τον έλεγχο βιομηχανικών διαδικασιών και επιχειρησιακών δραστηριοτήτων ενώ έχει χρησιμοποιηθεί και σε λύσεις διαχείρισης πληροφοριακής υποδομής (IT management solutions). Συστήματα διαχείρισης δικτύων λαμβάνουν γεγονότα από τη δικτυακή υποδομή και ενεργούν με βάση ορισμένους δεδομένους κανόνες. Παραδείγματα τέτοιων γεγονότων είναι οι ειδοποιήσεις (alerts) για «χαμηλή μνήμη» (memory low), για απόρριψη πακέτων (packets dropped), υπερβολική

χρήση CPU (server CPU usage too high), επιδείνωση των επιδόσεων μιας βάσης δεδομένων (database server performance) και άλλα. Ακόμη, ο συνδυασμός αυτοματοποιημένων διαδικασιών (process automation) με την επεξεργασία σύνθετων γεγονότων επιτρέπει την πλήρη αξιοποίηση της «ελαστικότητας» (elasticity) που παρέχουν λύσεις τύπου υπολογιστικής νέφους (cloud computing) και εικονοποίησης (virtualization).

2.3 Πλαίσιο και Επίγνωση Πλαισίου

Ο Dey (2001) ορίζει το **πλαίσιο (context)** ως «οποιαδήποτε πληροφορία μπορεί να χρησιμοποιηθεί για να χαρακτηρίσει την κατάσταση μιας οντότητας», η οποία μπορεί να είναι κάποιο πρόσωπο, τοποθεσία ή αντικείμενο που σχετίζεται ή αλληλεπιδρά με την εφαρμογή ή τον χρήστη. Οι Ye et al. (2012) διακρίνουν δύο είδη πλαισίων: τα πρωτεύοντα και δευτερεύοντα. Ως **πρωτεύον πλαίσιο (primary context)** ορίζουν το σύνολο των πρωτογενών δεδομένων που λαμβάνονται από πραγματικούς και λογικούς αισθητήρες (real and virtual sensors). Ως **δευτερεύον πλαίσιο (secondary context)** θεωρούν τις πληροφορίες που «συμπεραίνονται» ή παράγονται από την συνεπεξεργασία ροών πρωτογενών δεδομένων (πρωτευόντων πλαισίων). Οι da Cunha Mattos et al. (2014) αναφέρουν ότι κάθε ενέργεια που εκτελείται ή συμβάν «ανήκει» ή «υπάγεται» σε ένα πλαίσιο. Το πλαίσιο αποτελεί μια κοινή περιγραφή της γνώσης που ενυπάρχει σε μια φυσική, κοινωνική, ιστορική ή άλλη περίπτωση, μέσα στην οποία λαμβάνουν χώρα οι ενέργειες ή τα συμβάντα. Η γνώση αυτή δεν αποτελεί μέρος τους αλλά θέτει περιορισμούς στην εκτέλεση των ενεργειών ή την ερμηνεία των συμβάντων (Brézillon, 1999). Οι Harter et al. (1999) και Moran and Dourish (2001) έδωσαν επιπλέον ορισμούς οι οποίοι λαμβάνουν υπόψη και παράγοντες όπως τα χαρακτηριστικά συσκευών, χρηστών, προφίλ και προτιμήσεων (Feng et al., 2009).

Στη βιβλιογραφία έχουν προταθεί διάφορες κατηγοριοποιήσεις των πλαισίων αναλόγως με την οπτική γωνία του κάθε ερευνητή. Οι Dey and Abowd (2000) τα διακρίνουν σε πλαίσια τοποθεσίας, ταυτότητας, δραστηριότητας και χρόνου (location, identity, activity and time contexts). Οι Kaltz et al. (2005) τα διαχωρίζουν σε πλαίσια χρηστών, ρόλων, διαδικασιών, τοποθεσίας, χρόνου και συσκευής για να συμπεριλάβουν μια μεγαλύτερη γκάμα σεναρίων. Μολαταύτα σημειώνουν ότι η βέλτιστη κατηγοριοποίηση εξαρτάται σε σημαντικό βαθμό από την εκάστοτε εφαρμογή ή σενάριο χρήσης. Η Bauer (2012) συνέκρινε δεκατρία μεταμοντέλα πλαισίων και τα κατηγοριοποίησε ανάλογα με το βαθμό αφάιρεσης (abstraction level) που επιτυγχάνουν δεδομένου ενός πλαισίου. Οι Zainol and Nakata (2010) πρότειναν μια οντολογία πλαισίων που απαρτίζεται από τρεις βασικές κλάσεις: το εξωτερικό πλαίσιο, πλαίσιο της διεπαφής και το εσωτερικό πλαίσιο.

Οι Vieira et al. (2011) ορίζουν την έννοια του στοιχείου πλαισίου (context element) και τη διακρίνουν το πλαίσιο (context). Ένα στοιχείο πλαισίου είναι ουσιαστικά μια πληροφορία που χαρακτηρίζει μια παράμετρο ή ιδιότητα μιας οντότητας. Το πλαίσιο απ' την άλλη είναι ένα σύνολο από στοιχεία, τα οποία είναι απαραίτητα για την εκτέλεση ορισμένων ενεργειών από ένα χρήστη ή σύστημα. Σύμφωνα με τους ίδιους, τα στοιχεία του πλαισίου είναι σταθερά και γνωστά ήδη από το σχεδιασμό του συστήματος. Αντίθετα το πλαίσιο

είναι δυναμικό και δημιουργείται κατά την εκτέλεση του συστήματος όταν συμβαίνει κάποια αλληλεπίδραση με αυτό.

Οι Santos et al. (2007) προτείνουν ένα προσανατολισμένο σε υπηρεσίες ενδιάμεσο λογισμικό (middleware) για εφαρμογές ενήμερες για το πλαίσιο τους (service-oriented middleware for context-aware applications). Προκειμένου να παρέχει αυτή τη δυνατότητα, το ενδιάμεσο λογισμικό συμπεριλαμβάνει δύο υπηρεσίες: (α) την υπηρεσία διαχείρισης πλαισίου (Context Management Service), το οποίο παρέχει ένα σύνολο από διεπαφές προς διάφορα υποσυστήματα καθώς και ένα μοντέλο δεδομένων για την αναπαράσταση του πλαισίου, και (β) την υπηρεσία ανίχνευσης και ειδοποιήσεων (Awareness & Notification Service) που παρέχει ένα μηχανισμό κανόνων για την παρακολούθηση των συνθηκών του πλαισίου και λαμβάνει ειδοποιήσεις όταν εμφανίζονται συγκεκριμένες συνθήκες σε αυτό.

Όσον αφορά τα πλαίσια επιχειρησιακών διαδικασιών, οι Saidani and Nurcan (2007) παρουσίασαν μια προσέγγιση για μοντελοποίηση επιχειρησιακών διαδικασιών, η οποία υποστηρίζει την περιγραφή του πλαισίου. Σύμφωνα με τις ίδιες η δυνατότητα για ενσωμάτωση της γνώσης του πλαισίου (context related knowledge) επιτρέπει την πιο ευέλικτη και αποτελεσματική διαχείριση των διαδικασιών και την εκφραστικότερη αποτύπωση των επιχειρησιακών κανόνων. Η προσέγγιση περιλαμβάνει τέσσερα βήματα: την εκμαίευση, κατηγοριοποίηση, προσαρμογή και μέτρηση του πλαισίου για το εκάστοτε στιγμιότυπο της διαδικασίας.

Τέλος, οι Rosemann et al. (2008) προτείνουν το πλαίσιο εργασίας Onion, για την καλύτερη κατανόηση διαφορετικών τύπων πλαισίων και των επιπτώσεών τους στις επιχειρησιακές διαδικασίες. Το πλαίσιο εργασίας διακρίνει τέσσερις τύπους πλαισίων: άμεσα, εσωτερικά, εξωτερικά και περιβαλλοντικά. Προτείνουν ακόμη μια προσέγγιση για μοντελοποίηση διαδικασιών προσανατολισμένη σε στόχους (goal-oriented), όπου το πλαίσιο μπορεί να γίνει κατανοητό, να ταξινομηθεί και να ενσωματωθεί.

Οι Soffer et al. (2010) παρουσίασαν μια αυτοματοποιημένη προσέγγιση εμπειρικής μάθησης για τη βελτίωση των επιχειρησιακών διαδικασιών με την πάροδο του χρόνου. Σύμφωνα με αυτή συνδυάζονται τρεις πλευρές των επιχειρησιακών διαδικασιών: τα μονοπάτια εκτέλεσης, το πλαίσιο και οι στόχοι. Η μάθηση πραγματοποιείται σε κύκλους (learning cycles) όπου προβλέπεται και μια φάση αναγνώρισης του σχετικού πλαισίου, καθώς και χρήσης του για τη βελτίωση του μοντέλου της διαδικασίας. Κατά την εκτέλεση της διαδικασίας χρησιμοποιείται το βελτιωμένο μοντέλο ενώ τα αποτελέσματα που παράγονται αποθηκεύονται για τον επόμενο κύκλο μάθησης.

2.3.1 Επίγνωση Πλαισίου (Context Awareness)

Ως επίγνωση πλαισίου νοείται η ικανότητα συστημάτων να ανιχνεύουν το περιβάλλον τους προκειμένου να μπορούν να αντιδρούν σε αυτό. Ο όρος προέρχεται από το πεδίο της διάχυτης υπολογιστικής (pervasive computing) και εισήχθη από τους Schilit and Theimer (1994) για την ανάπτυξη λογισμικού που προσαρμόζεται ανάλογα με την τοποθεσία χρήσης του, τη συλλογή πληροφοριών για ανθρώπους και αντικείμενα που γειτνιάζουν, καθώς και τις αλλαγές που συμβαίνουν σε αυτά. Αφορά την αξιοποίηση του πλαισίου για την

προσαρμογή της συμπεριφοράς συστημάτων στο περιβάλλον τους (Feng et al., 2009). Η επίγνωση πλαισίου αποτελεί ένα σημαντικό επιστημονικό πεδίο ειδικά σε σχέση με την έρευνα σε κινητές συσκευές και υπηρεσίες.

Ορισμένες ενδιαφέρουσες επισκοπήσεις της βιβλιογραφίας επάνω στο πεδίο αυτό έχουν δημοσιευτεί από τους Perera et al. (2014), da Cunha Mattos et al. (2014), Verbert et al. (2012), Bettini et al. (2010), Baldauf et al. (2007) και Bolchini et al. (2007). Επισκοπήσεις που εξειδικεύονται στην επίγνωση πλαισίου για συστήματα κινητών υπηρεσιών έχουν δοθεί από τους Yurur et al. (2014), del Carmen Rodríguez-Hernández and Ilarri (2014), Makris et al. (2013), Ganti et al. (2011) Lane et al. (2010) και G. Chen and Kotz (2000).

2.4 Κατάσταση και Επίγνωση Κατάστασης

Ιστορικά ο πρώτος τυπικός (formal) ορισμός της **Κατάστασης (Situation)** δόθηκε από τους McCarthy and Hayes (1969) με τον Λογισμό Καταστάσεων (Situation Calculus), όπου χρησιμοποίησαν εκφράσεις Λογικής Πρώτης Τάξεως (First-Order Logic) για να ορίσουν μια κατάσταση ως ένα στιγμιότυπο (snapshot) των συνολικών συνθηκών που επικρατούν σε έναν «κόσμο» (world) σε μια συγκεκριμένη χρονική στιγμή. Όμως η εξέταση μιας κατάστασης ως το σύνολο των συνθηκών ενός κόσμου ήταν υπολογιστικά μη αποδοτική, γι' αυτό οι Pirri and Reiter (1999) όρισαν την κατάσταση ως μια ακολουθία ενεργειών, επιτρέποντας έτσι τον υπολογισμό των τρεχουσών συνθηκών αν γνωρίζουμε τις αρχικές και το πως η ακολουθία των ενεργειών τις μεταβάλλει.

Οι Ye et al. (2012), ως κατάσταση ορίζουν μια αφαίρεση (abstraction) συμβάντων του πραγματικού κόσμου, η οποία προκύπτει από το αντίστοιχο πλαίσιο και τη διαμόρφωση των σχέσεων ενδιαφέροντος μέσα σε αυτό, καθώς και μια υπόθεση (hypothesis) για το πώς το πλαίσιο σχετίζεται με τους παράγοντες εκείνους που ενδιαφέρουν τις εφαρμογές. Η ανίχνευση καταστάσεων απαιτεί την από κοινού επεξεργασία αρκετών πλαισίων, της γνώσης του σχετικού πεδίου (domain knowledge), καθώς και χωρικά και χρονικά μοντέλα των αναμενόμενων συμπεριφορών του φαινομένου ενδιαφέροντος.

Οι Bettini et al. (2010) ορίζουν την κατάσταση (situation) ως μια «σημασιολογική αφαίρεση» (semantic abstraction) απλών, χαμηλού επιπέδου δεδομένων (low-level data). Χρησιμοποιούν δε την έννοια αυτή για να ομαδοποιήσουν και μοντελοποιήσουν τις συνθήκες και τους περιορισμούς που απαιτείται να αναγνωριστούν προκειμένου να διαπιστωθεί η ανάγκη εκτέλεσης συγκεκριμένων ενεργειών da Cunha Mattos et al. (2014). Επίσης μπορεί να οριστεί ως μια «εξωτερική σημασιολογική ερμηνεία» των πρωτογενών δεδομένων. Με τον όρο «ερμηνεία» εννοούμε ότι η κατάσταση δίνει νόημα στα δεδομένα ενώ με τον όρο «εξωτερική» εννοούμε ότι η ερμηνεία είναι από την οπτική των εφαρμογών κι όχι των πηγών δεδομένων (π.χ. αισθητήρες). Το «σημασιολογική» εννοεί ότι η ερμηνεία δίνεται με βάση δομές και σχέσεις μεταξύ δεδομένων (ίδιου ή διαφορετικού τύπου) (Marrella et al., 2014).

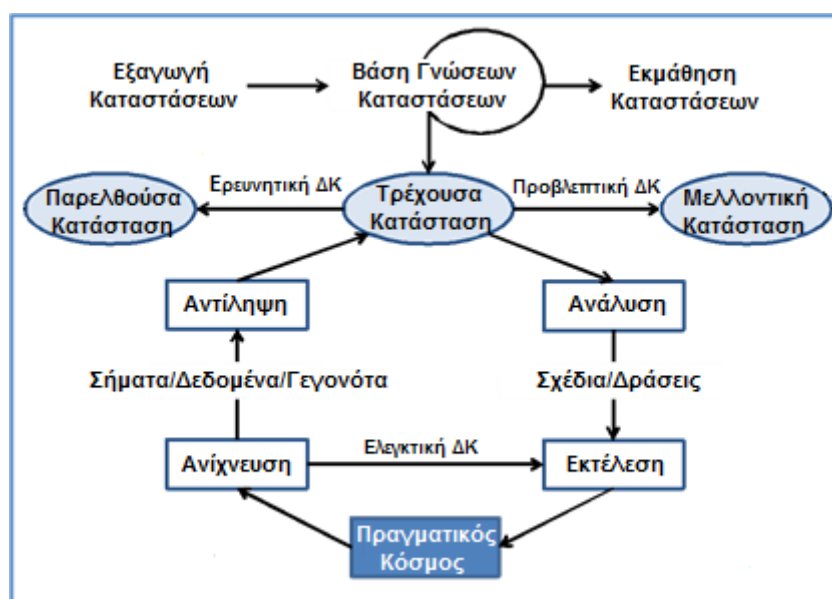
Είναι φανερό ότι η κατάσταση είναι μια έννοια σχετική. Τα ίδια πρωτογενή δεδομένα μπορούν να ερμηνευθούν ως διαφορετικές καταστάσεις, ανάλογα με τις ανάγκες της εκάστοτε εφαρμογής. Επίσης, μια κατάσταση μπορεί να είναι μια απλή διαμόρφωση μιας

οντότητας (π.χ. η πόρτα είναι ανοικτή) ή μια ενέργεια που συμβαίνει στο περιβάλλον (π.χ. οδήγηση). Μεταξύ των καταστάσεων μπορούν να οριστούν σχέσεις, συμπεριλαμβανομένων και των αναφερόμενων ακολούθως (Marrella et al., 2014) :

- **Γενίκευση.** Μια κατάσταση μπορεί να θεωρηθεί ως γενικότερη μιας άλλης αν η εμφάνιση της δεύτερης υπονοεί και την πρώτη.
- **Σύνθεση.** Μια κατάσταση μπορεί να αναλυθεί σε απλούστερες καταστάσεις.
- **Εξάρτηση.** Μια κατάσταση εξαρτάται από μια άλλη αν η εμφάνιση της πρώτης καθορίζεται από την εμφάνιση της δεύτερης. Οι Choujaa and Dulay (2009) θεωρούν ότι μια σχέση εξάρτησης μπορεί να είναι βραχυπρόθεσμη ή μακροπρόθεσμη.
- **Αντίθεση.** Δύο καταστάσεις είναι αμοιβαία αποκλειόμενες αν δεν μπορούν να συνυπάρξουν την ίδια χρονική περίοδο, στον ίδιο τόπο και για το ίδιο θέμα.
- **Χρονική Αλληλουχία.** Μια κατάσταση μπορεί να εμφανιστεί πριν ή μετά ή ταυτόχρονα με μια άλλη.

2.4.1 Διαχείριση Καταστάσεων (Situation Management)

Η έννοια της κατάστασης αποτελεί μια σημαντική παράμετρο σε γνωστικά πεδία όπου απαιτείται αυτόματη και συνεχής αναγνώριση πολύπλοκων και συχνά απρόβλεπτων, δυναμικών καταστάσεων, καθώς και η αντίδραση σε αυτές. Κατά συνέπεια αναζητούνται αποτελεσματικές μέθοδοι για την αναγνώριση, πρόβλεψη, τεκμηρίωση και έλεγχο των καταστάσεων. Οι λειτουργίες αυτές καλούνται Διαχείριση Κατάστασης (Situation Management) (Jakobson et al., 2007), η οποία θεωρείται ως μια συνεργατική, οδηγούμενη από στόχους διαδικασία για (α) ανίχνευση και συλλογή πληροφοριών, (β) αναγνώριση και επίγνωση καταστάσεων, (γ) ανάλυση παρελθουσών καταστάσεων και πρόβλεψη μελλοντικών, και (δ) τεκμηρίωση, σχεδιασμό και υλοποίηση ενεργειών ούτως ώστε να επιτευχθεί η επιθυμητή κατάσταση-στόχος, μέσα όμως σε ένα πλαίσιο προκαθορισμένων περιορισμών (pre-defined constraints).



Σχήμα 2-2. Ο κύκλος ζωής της Διαχείρισης Κατάστασης (Jakobson et al., 2007)

Σύμφωνα με τον Jakobson (2008) υπάρχουν τρεις τύποι διαχείρισης καταστάσεων: ο (δι)ερευνητικός (investigative), ο ελεγκτικός (control) και ο προβλεπτικός (predictive). Ο ερευνητικός τύπος διαχείρισης καταστάσεων επικεντρώνεται στην αναδρομική (retrospective) ανάλυση των αιτιακών συνθηκών που εξηγούν το γιατί μια συγκεκριμένη κατάσταση συνέβη. Ο ελεγκτικός τύπος αποβλέπει στην διατήρηση μιας κατάστασης ή στην μεταβολή της προς μια επιθυμητή, ενώ ο προβλεπτικός τύπος αποσκοπεί στην πρόβλεψη πιθανών μελλοντικών καταστάσεων. Για παράδειγμα, ο εντοπισμός της αιτίας αποτυχίας μετάδοσης ενός πακέτου δεδομένων μέσω ενός τηλεπικοινωνιακού δικτύου αποτελεί τον στόχο του ερευνητικού τύπου διαχείρισης καταστάσεων. Η διατήρηση της υψηλής ανταποκρισιμότητας μιας εφαρμογής νέφους, με την προσθήκη επιπλέον υπολογιστικών πόρων (CPU cores), θα ήταν στόχος του ελεγκτικού τύπου. Η πρόβλεψη μιας πιθανής «επίθεσης» σε ένα στοιχείο ενός δημόσιου τηλεπικοινωνιακού δικτύου αποτελεί παράδειγμα του προβλεπτικού τύπου.

Επιμέρους παράγοντες της διαχείρισης καταστάσεων που αξίζει να αναφερθούν είναι η επίγνωση καταστάσεων (situation awareness), η επίλυση καταστάσεων (situation resolution) καθώς και η εξαγωγή γνώσης και η εκμάθηση από αυτές. Η επίγνωση κατάστασης βασίζεται στην ανίχνευση (sensing) και αντίληψη (perception) των συνθηκών ενός συστήματος και αποσκοπεί στην κατανόηση της τρέχουσας κατάστασής του. Η επίλυση καταστάσεων αποβλέπει στον σχεδιασμό (planning) και εκτέλεση (execution) ενεργειών προς επίτευξη ή διατήρηση μιας συγκεκριμένης κατάστασης-στόχου. Με αυτούς τους δύο παράγοντες δημιουργείται ένας βρόχος ελέγχου κατάστασης (situation control loop). Τέλος η εξαγωγή γνώσης και η εκμάθηση είναι η κύρια πηγή απόκτησης γνώσης (knowledge acquisition) που απαιτείται σε άλλες διαδικασίες.

2.4.2 Επίγνωση Κατάστασης (Situation Awareness)

Ως επίγνωση κατάστασης νοείται η ικανότητα των συστημάτων να αντιλαμβάνονται τα στοιχεία του περιβάλλοντος, ειδικά σε σχέση με το χρόνο και την τοποθεσία τους, να κατανοούν το νόημά τους και να προβλέπουν την εξέλιξή τους στο εγγύς μέλλον. Η επίγνωση κατάστασης αφορά τη «γνώση» του τι συμβαίνει στο περιβάλλον προκειμένου να κατανοηθεί ο τρόπος με τον οποίο πληροφορίες, γεγονότα και ενέργειες επηρεάζουν τους στόχους του συστήματος, τόσο άμεσα όσο και στο εγγύς μέλλον.

Παρόλο που έχουν προταθεί αρκετοί ορισμοί, αυτός της Mica Endsley είναι ο ευρύτερα αποδεκτός και εφαρμόσιμος σε πολλαπλά πεδία. Σύμφωνα με τον ορισμό αυτόν, ως επίγνωση κατάστασης ορίζεται «η αντίληψη (perception) των στοιχείων του περιβάλλοντος μέσα σε ένα συγκεκριμένο χρονικό και χωρικό πλαίσιο, η κατανόηση (comprehension) του νοήματός τους και η προβολή/προέκτασή (projection) τους στο εγγύς μέλλον¹⁰» (Endsley and Connors, 2008; Endsley, 1995).

¹⁰ "Situation Awareness is the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future" (Endsley & Connors, 2008), (Endsley, 1995)

Σχετικά πρόσφατα προτάθηκαν διάφορες μέθοδοι για αναπαράσταση καταστάσεων, οι οποίες καλύπτουν ένα ευρύ φάσμα, όπως προσεγγίσεις βασισμένες σε γεγονότα (Cardell-Oliver and Liu, 2010), προσεγγίσεις βασισμένες σε οντολογίες και αντικειμενοστραφείς προσεγγίσεις. Η προσέγγιση SAWA (Situation Awareness Assistant) (Kokar et al., 2009) βασίζεται σε μια οντολογία για την επίγνωση καταστάσεων. Η οντολογία που προτείνουν οι Yau et al. (2005) διακρίνει τις καταστάσεις σε απλές (atomic) και σύνθετες (composite), ενώ η οντολογία core SAW από τους Baumgartner et al. (2010) αναπαριστά μια κατάσταση ως μια συλλογή από «αντικείμενα κατάστασης» (situation objects) που περιλαμβάνουν διάφορες οντότητες, συσχετίσεις καθώς και άλλες καταστάσεις. Επίσης εισάγει την έννοια των βασικών χωρο-χρονικών συσχετίσεων μεταξύ των παρατηρούμενων και των πραγματικών αντικειμένων.

Άλλες εργασίες προτείνουν τη χρήση ευφυών πρακτόρων (intelligent agents) για τη μοντελοποίηση της επίγνωσης κατάστασης. Για παράδειγμα οι Thangarajah et al. (2006) υιοθετούν την έννοια της κατάστασης προκειμένου να επιτύχουν σημασιολογικά επαρκείς προδιαγραφές για τη λήψη αποφάσεων από ευφυείς πράκτορες. Οι Urlings et al. (2003) χρησιμοποιούν πράκτορες BDI (Beliefs–Desires–Intentions) για την επαύξηση της επίγνωσης κατάστασης σε στρατιωτικά περιβάλλοντα. Οι So and Sonenberg (2004) ενσωματώνουν την έννοια της επίγνωσης κατάστασης σε ένα υπολογιστικό μοντέλο για πράκτορες με προβλεπτική συμπεριφορά (proactive behavior).

Οι Ye et al. (2012) υπογραμμίζουν ότι η επίγνωση πλαισίου και η επίγνωση κατάστασης είναι δύο διακριτές έννοιες. Η επίγνωση κατάστασης εμπλουτίζει την έννοια του πλαισίου συμπεριλαμβάνοντας τη διάσταση του χρόνου καθώς και άλλες διαστάσεις (K. Lee et al., 2009). Για παράδειγμα μια κατάσταση μπορεί να συμβαίνει μόνο σε συγκεκριμένο χρόνο μέσα σε μια μέρα, μπορεί να έχει συγκεκριμένη διάρκεια ή συχνότητα μέσα σε μια περίοδο ή να συμβαίνει με συγκεκριμένη σειρά (Ye et al. 2012). Σύμφωνα με τους Feng et al. (2009) η επίγνωση κατάστασης εστιάζει στην αποτύπωση του περιβάλλοντος του χρήστη ή εφαρμογής, προκειμένου να βοηθήσει στην κατανόηση της τρέχουσας κατάστασης. Η επίγνωση πλαισίου απ' την άλλη, αφορά την αξιοποίηση του πλαισίου του χρήστη προκειμένου να τον βοηθήσει να έχει μια πιο αποδοτική αλληλεπίδραση με το σύστημα, προσαρμόζοντας τη συμπεριφορά του συστήματος.

Κλείνοντας αναφέρουμε ότι ένα από τα βασικά προβλήματα στην επεξεργασία σύνθετων γεγονότων (CEP) είναι ο εντοπισμός και η αναφορά καταστάσεων έτσι ώστε να είναι δυνατή η αντίδραση σε αυτές. Σε αυτό το πλαίσιο, η κατάσταση ορίζεται συχνά ως η εμφάνιση ενός (σύνθετου) γεγονότος που ενδεχομένως απαιτεί κάποια αντίδραση (Adi and Etzion, 2004). Σε πεδία όπου χαρακτηρίζονται από την απαίτηση ανίχνευσης καταστάσεων σε πραγματικό χρόνο, γίνεται εμφανής και επιτακτική η ανάγκη για αυτόματα και συνεχή αναγνώριση και αντίδραση σε αυτές.

Περισσότερες εργασίες επάνω στην επίγνωση καταστάσεων μπορεί να αναζητηθούν σε βιβλιογραφικές επισκοπήσεις όπως των Ye et al. (2012) και Endsley (2000).

2.4.3 Εκτίμηση Καταστάσεων (Situation Assessment)

Η Endsley (1995) υποστηρίζει ότι η επίγνωση κατάστασης, ως κατοχή γνώσης, διακρίνεται από τη διαδικασία που χρησιμοποιείται για την απόκτηση της γνώσης αυτής. Η διαδικασία αυτή ονομάζεται εκτίμηση κατάστασης και μπορεί να διαφέρει σημαντικά από περίπτωση σε περίπτωση και από πλαίσιο σε πλαίσιο. Σημειώνεται ότι η επίγνωση κατάστασης δεν παράγεται απλά από την εκτίμηση κατάστασης αλλά μπορεί και αντίστροφα να ενεργοποιήσει και «οδηγήσει» τη διαδικασία αυτή κατά έναν επαναληπτικό τρόπο (Endsley, 2000). Οι τεχνικές εκτίμησης καταστάσεων μπορούν να κατηγοριοποιηθούν με βάση διάφορα κριτήρια (Ye et al., 2012), όπως :

- το αν απαιτούν επίβλεψη (supervised or unsupervised),
- αν είναι στατικές (αποσκοπούν στην αποτύπωση χαρακτηριστικών των πρωτογενών δεδομένων όταν συντρέχει μια κατάσταση) ή ακολουθιακές (αποσκοπούν στην αποτύπωση των χρονικών συσχετισμών μεταξύ πρωτογενών δεδομένων αλλά και μεταξύ καταστάσεων),
- το επίπεδο αφαίρεσης που επιτυγχάνουν (level of abstraction),
- την ικανότητα διαχείρισης δεδομένων με εγγενή αβεβαιότητα ή χρήση «χονδρικών ή υπεραπλουστευμένων» κανόνων που έχουν οριστεί με ad hoc τρόπο,
- τους χρονικούς συσχετισμούς των καταστάσεων,
- την πολυπλοκότητα,
- την ικανότητα ενσωμάτωσης και παραγωγής γνώσης,
- την απαιτούμενη προσπάθεια ανάπτυξης (development effort),
- τους τύπους των πρωτογενών δεδομένων.

Οι Ye et al. (2012) διακρίνουν τις τεχνικές εκτίμησης καταστάσεων σε δύο κατηγορίες: τις βασισμένες στην προδιαγραφή των καταστάσεων (specification-based techniques) και τις βασισμένες σε μηχανική μάθηση (machine learning-based techniques). Πολλές από αυτές προέρχονται από το χώρο της τεχνητής νοημοσύνης.

Τυπικά, οι βασισμένες σε προδιαγραφές προσεγγίσεις (specification-based approaches) απαιτούν την εκ των προτέρων (a priori) ανάπτυξη ενός μοντέλου καταστάσεων από έναν εμπειρογνώμονα, προτού καταστεί δυνατό να χρησιμοποιηθούν σε κάποια εφαρμογή και να εφαρμοστούν στην επεξεργασία των δεδομένα. Παραδείγματα τέτοιων είναι τεχνικές που στηρίζονται σε κανόνες λογικής πρώτης τάξεως (Gu, Wang, et al., 2004; Loke, 2010), στον λογικό προγραμματισμό (logic programming) (Henricksen and Indulska 2006; Gu, Wang, et al., 2004; Gu et al., 2005), προσεγγίσεις με χωροχρονική λογική (spatial and temporal logic) (Cook et al. 2009), προσεγγίσεις με ασαφή λογική (fuzzy logic) (Delir Haghghi et al., 2008; Anagnostopoulos et al., 2007) αλλά και προσεγγίσεις βασισμένες στη θεωρία Dempster–Shafer (DST) (McKeever et al., 2009; Sentz and Ferson, 2002; Zhang et al., 2009). Ιδιαίτερο ενδιαφέρον παρουσιάζουν τεχνικές βασισμένες σε οντολογίες όπου υπάρχει σημαντικό έργο (Costa et al. 2006; Gu, Wang, et al. 2004; H. Chen et al. 2003; Matheus et al. 2003; Yau and Liu 2006; Weißenberg et al. 2006; Smimov et al. 2007; Baumgartner and Retschitzegger 2007; Bettini et al. 2010; L. Chen et al. 2009; Klein et al. 2007).

Οι βασισμένες σε μηχανική μάθηση προσεγγίσεις (machine learning-based approaches) απαιτούν ένα σύνολο δεδομένων από το περιβάλλον εφαρμογής τους προκειμένου να «αναπτύξουν» το μοντέλο επίγνωσης κατάστασης που θα χρησιμοποιούν. Ένα από τα κυριότερα σημεία υπεροχής των τεχνικών μηχανικής μάθησης σε σχέση με τις προσεγγίσεις που βασίζονται στην προδιαγραφή των καταστάσεων είναι η ικανότητά τους να αποκαλύπτουν μοτίβα και συσχετίσεις μεταξύ των δεδομένων. Μπορούν να εντοπίζουν και «μαθαίνουν» τις σχέσεις μεταξύ συνδυασμών περιγραφικών ή αριθμητικών δεδομένων και των αντίστοιχων καταστάσεων.

Παραδείγματα τέτοιων είναι οι τεχνικές που βασίζονται στη χρήση του μοντέλου Naïve Bayes (Patterson et al. 2003; Tapia et al. 2004; van Kasteren and Krose 2007; Bao and Intille 2004; J. Han and Kamber 2006; Muhlenbrock et al. 2004), στη χρήση μπεϋζιανών δικτύων (Bayesian networks) (J. Ye et al. 2007; Gu, Pung and Zhang, 2004; Ding and Peng 2004; Abdelsalam and Ebrahim 2004; Truong, Lee and Lee 2005), τεχνικές που βασίζονται σε Λανθάνοντα Μοντέλα Markov (Hidden Markov models) (Atallah and Yang, 2009; Clarkson and Pentland, 1999; Duong et al., 2005; Hasan et al., 2008; Minnen et al., 2005; van Kasteren et al., 2008; Wojek et al., 2006) και στις βελτιώσεις τους (Bui et al., 2002; Fine and Singer, 1998; Nguyen et al., 2005), προσεγγίσεις βασισμένες σε Δένδρα Αποφάσεων (Decision trees) (Bao and Intille, 2004; Cheung and Vogel, 2005; Karantonis et al., 2006; Mathie et al., 2004), σε Νευρωνικά Δίκτυα (Neural networks) (Yang et al. 2008) καθώς και σε Μηχανές Διανυσμάτων Υποστήριξης (Support vector machines) (Kanda et al., 2008; Patel et al., 2007)

Οι προσεγγίσεις που βασίζονται στην προδιαγραφή των καταστάσεων απαιτούν την αποτύπωση των ορισμών των καταστάσεων και των κανόνων επαγωγής (συλλογισμού) από κάποιον εμπειρογνώμονα. Έτσι βασίζονται στη γνώση του πεδίου που αυτός τους ενσωματώνει. Η ενσωμάτωση των χωροχρονικών συσχετισμών και η διαχείριση της αβεβαιότητας μπορούν να επιτευχθούν σε ορισμένες μόνο περιπτώσεις (Ye et al. 2012). Από την άλλη, οι προσεγγίσεις που βασίζονται σε μηχανική μάθηση αίρουν την ανάγκη για γνώση του πεδίου αλλά απαιτούν μεγάλο όγκο δεδομένων εκπαίδευσης (training data) από το περιβάλλον εφαρμογής, προκειμένου να αναπτύξουν το αντίστοιχο μοντέλο. Η αβεβαιότητα αντιμετωπίζεται εγγενώς, γεγονός που αποτελεί ένα πλεονέκτημα των τεχνικών αυτών. Επίσης ορισμένες από αυτές μπορούν να διαχειριστούν και τους χρονικούς συσχετισμούς των δεδομένων. Ορισμένα μειονεκτήματά των μεθόδων αυτών είναι ότι η συλλογιστική με την οποία παράγουν τα μοντέλα δεν γίνεται εύκολα ξεκάθαρη και κατανοητή από τους χρήστες (Ye et al. 2012) ενώ οι περισσότερες από αυτές απαιτούν εποπτευόμενη μάθηση (supervised learning) γεγονός που θέτει περιορισμούς στη χρήση τους σε πραγματικές εφαρμογές όπου οι απαιτήσεις για κλιμάκωση, εφαρμοσιμότητα, ιδιωτικότητα και προσαρμοσιμότητα είναι αρκετά υψηλές (Gu, Chen, et al., 2010).

Περισσότερες πληροφορίες μπορούν να αναζητηθούν σε αντίστοιχες βιβλιογραφικές επισκοπήσεις όπως αυτή των Ye et al. (2012).

2.5 Καινοτόμες προσεγγίσεις για οδηγούμενη από γεγονότα επίγνωση καταστάσεων

Ακολούθως παρουσιάζουμε ορισμένες εργασίες που έχουν προταθεί στη βιβλιογραφία και εισηγούνται καινοτόμες προσεγγίσεις που αφορούν τον εντοπισμό διαφόρων καταστάσεων ενδιαφέροντος και αντίδραση σε αυτές, την παραγωγή προτάσεων για εγγραφή (subscription) σε περιεχόμενο και ροές γεγονότων καθώς και τη δυναμική δημιουργία ροών γεγονότων με βάση την επικρατούσα κατάσταση. Ακόμη αφορούν μια οδηγούμενη από γεγονότα και προσανατολισμένη σε υπηρεσίες αρχιτεκτονική για «κινητές υπηρεσίες» (mobile services).

Το σύστημα SARI (Schiefer and Seufert, 2005; Schiefer et al., 2007; Schiefer et al., 2009) παρέχει ένα πλαίσιο εργασίας για διαχείριση που βασίζεται σε γεγονότα κανόνων (event-based rule management framework). Επιτρέπει δε τη μοντελοποίηση επιχειρησιακών καταστάσεων και εξαιρέσεων (exceptions) με τη χρήση κανόνων αντίδρασης και αντίδρασης (sense and respond rules). Οι κανόνες αυτοί, τους οποίους δημιουργούν και διαχειρίζονται οι χρήστες μέσω ενός γραφικού περιβάλλοντος, επιτρέπουν την αντίδραση σε επιχειρησιακές καταστάσεις και την αποστολή απαντητικών γεγονότων (respond events). Οι επιχειρησιακές καταστάσεις ορίζονται από συνθήκες γεγονότων (event conditions) και πρότυπα γεγονότων (event patterns) και μπορούν να συνδυάζονται με οποιονδήποτε τρόπο. Κατά την λειτουργία του συστήματος οι κανόνες εξετάζονται από «υπηρεσίες γεγονότων» που εκτελούνται ταυτόχρονα σε πολλούς υπολογιστές, και οι οποίες μπορούν να ενσωματωθούν σε εφαρμογές υπηρεσιών. Το σύστημα επιτρέπει τον συσχετισμό γεγονότων με χρήση συνόλων συσχέτισης (correlation sets) για τον ορισμό πολύπλοκων σχέσεων μεταξύ γεγονότων και επιχειρησιακών ενεργειών. Το SARI μπορεί επιπλέον να αντιδρά τόσο με την παραγωγή νέων γεγονότων όσο και με τη δημιουργία ή ενημέρωση διαφόρων επιχειρησιακών αντικειμένων. Παρότι όμως οι κανόνες αντίδρασης και αντίδρασης μπορούν να χρησιμοποιηθούν σε διάφορα σενάρια λήψης αποφάσεων, το SARI δεν υποστηρίζει εγγενώς την παροχή προτάσεων για την προσαρμογή επιχειρησιακών διαδικασιών.

Οι Laliwala et al. (2006) εστίασαν στην παροχή προτάσεων για εγγραφή σε ροές γεγονότων με βάση το θέμα, συσχετίζοντας τις προτιμήσεις, το πλαίσιο και την τοποθεσία του χρήστη με τις σημασιολογικές περιγραφές των ροών γεγονότων. Η συσχέτιση αυτή επιτυγχάνεται με χρήση μιας κατάλληλης οντολογίας με βάση την οποία συσχετίζονται έννοιες που αφορούν τις προτιμήσεις του χρήστη, το πλαίσιό του και την τοποθεσία του, με έννοιες που χρησιμοποιούνται για την περιγραφή των γεγονότων. Ακολουθώντας μια παρεμφερή προσέγγιση, οι Berkovsky and Eytani (2005) περιέγραψαν συνοπτικά μια προσέγγιση για εγγραφή σε περιεχόμενο με σημασιολογική επίγνωση του πλαισίου (semantic context-aware subscription). Δεδομένης μιας οντολογίας με πιθανά πλαίσια χρηστών, η προσέγγιση αυτή αντιστοιχίζει συγκεκριμένα ενδιαφέροντα των χρηστών σε μια κατάλληλη δομή δεδομένων. Τα σχετικά γεγονότα επίσης αντιστοιχίζονται στην ίδια δομή δεδομένων. Όταν το πλαίσιο ενός χρήστη αλλάξει τότε δημιουργείται δυναμικά μια νέα εγγραφή. Οι Rodriguez-Dominguez et al. (2010) παρουσιάζουν ένα μοντέλο επανακαθοριζόμενων γεγονότων (redefinable events), τα οποία χρησιμοποιούνται για να υποστηρίξουν τη

δυναμική αναδιαμόρφωση (reconfiguration) των επικοινωνιών σε «πανταχού παρόντα» (ubiquitous) συστήματα. Στο μοντέλο αυτό τα γεγονότα αποτελούνται από «κόμβους», οι οποίοι δομούνται ως τριάδες της μορφής «αναγνωριστικό-τύπος-τιμή» και αναπαριστούν οποιαδήποτε πληροφορία. Προκειμένου να αντιμετωπιστεί η δυναμική φύση των «πανταχού παρόντων» (ubiquitous) συστημάτων, τα γεγονότα προσαρμόζονται στις αλλαγές που συμβαίνουν στο σύστημα (όπως η προσθήκη νέων δικτυακών συσκευών) επαναπροσδιορίζοντας το σύνολο των κόμβων γεγονότων κατά την εκτέλεση του συστήματος. Έτσι είναι δυνατό να προκύψουν δυναμικά νέες ροές γεγονότων (event streams) με βάση την διαρκώς μεταβαλλόμενη δομή των κόμβων γεγονότων και τις αντίστοιχες σημασιολογικές περιγραφές των κόμβων. Αυτή η προσέγγιση επιτρέπει εμμέσως την δυναμική εγγραφή σε ροές γεγονότων, διότι επιτρέπει στους χρήστες – συνδρομητές (subscribers) να λαμβάνουν ειδοποιήσεις για νέα είδη γεγονότων που δεν υπήρχαν κατά την αρχική εγγραφή. Αυτό συμβαίνει αν οι σημασιολογικές περιγραφές των νέων γεγονότων ταιριάζουν με την αντίστοιχη σημασιολογική περιγραφή της εγγραφής.

Οι Hinze, Michel, et al. (2009) προτείνουν μία αρχιτεκτονική βασισμένη σε γεγονότα και προσανατολισμένη σε υπηρεσίες (event-based service-oriented architecture) για τη συνεργασία κινητών υπηρεσιών (mobile services). Η εν λόγω αρχιτεκτονική επιτρέπει στις υπηρεσίες να αλληλεπιδρούν μεταξύ τους με χρήση ενός βασισμένου σε γεγονότα ενδιάμεσου λογισμικού (event-based middleware), όπου τα γεγονότα προκαλούν τη μετάδοση πληροφοριών. Όταν ξεκινά μια υπηρεσία συνδέεται σε έναν μεσάζοντα (broker), γνωστοποιεί (advertises) τις πληροφορίες που παρέχει και εγγράφεται για λήψη άλλων πληροφοριών. Ο μεσάζον (broker) διατηρεί αυτές τις πληροφορίες σε έναν κατάλογο και είναι υπεύθυνος για την εγγραφή (registration) και διαγραφή (deregistration) των υπηρεσιών, καθώς και την αποστολή (publish) και διήθηση (filtering) των γεγονότων. Οι υπηρεσίες παρέχουν μια περιγραφή τους, τη γνωστοποίηση (advertisement) των παρεχόμενων πληροφοριών, τις απαιτούμενες λειτουργικές συνθήκες και τους κανόνες εγγραφής. Οι κανόνες αυτοί επιτρέπουν σε μια υπηρεσία να θέτει προτεραιότητες σε συγκεκριμένους τύπους γεγονότων σε σχέση με άλλους, ή να επιλέγει μεταξύ πολλών παρόχων πληροφορίας. Η αποτίμηση ενός τέτοιου κανόνα οδηγεί σε εγγραφή σε μια ροή γεγονότων.

3 Προσαρμογή Επιχειρησιακών Διαδικασιών

Η «προσαρμογή» (adaptation) ως ένα από τα βασικά φαινόμενα της βιολογίας είναι η εξελικτική διαδικασία κατά την οποία ένας πληθυσμός σταδιακά αποκτά χαρακτηριστικά καταλληλότερα σε σχέση με το περιβάλλον που διαβιώνει¹¹ (Martin, 2010; Williams, 1966). Η έννοια όμως της προσαρμογής έχει χρησιμοποιηθεί εκτεταμένα και σε άλλα γνωστικά αντικείμενα, συμπεριλαμβανομένης της επιστήμης των υπολογιστών. Θεωρείται δε ως μία από τις πιο επιθυμητές ικανότητες των σύγχρονων, δυναμικών και κατανεμημένων συστημάτων.

Σύμφωνα με το S-Cube, το πιο γνωστό Ευρωπαϊκό Δίκτυο Αριστείας¹² στο χώρο των υπηρεσιών, «προσαρμογή είναι η διαδικασία τροποποίησης μιας εφαρμογής προκειμένου αυτή να ικανοποιεί νέες απαιτήσεις ή να ανταποκρίνεται σε νέες καταστάσεις, τις οποίες υπαγορεύει το περιβάλλον, με βάση προδιαγεγραμμένες στρατηγικές» (S-Cube, 2008b). Οι τροποποιήσεις μπορεί να αφορούν κάποιο τμήμα της εφαρμογής ή μόνο μια συγκεκριμένη μονάδα ή υπηρεσία αυτής. Ακόμη μπορεί να αφορούν τις αλληλεπιδράσεις και σχέσεις μεταξύ των μερών της εφαρμογής. Η έννοια της προσαρμογής έχει μελετηθεί εκτενώς καθότι θεωρείται ως μία πολύ επιθυμητή ιδιότητα των συστημάτων που δραστηριοποιούνται σε δυναμικά και κατανεμημένα περιβάλλοντα.

3.1 Ευελιξία Συστημάτων και Διαδικασιών (Flexibility)

Προκειμένου ένα πληροφοριακό σύστημα να είναι αποτελεσματικό πρέπει να είναι προσαρμοστικό και ευέλικτο, δηλαδή να μπορεί να ανταπεξέρχεται ως ένα βαθμό στην μεταβλητότητα των απαιτήσεων που απορρέουν από τις ανάγκες του εκάστοτε χρήστη ή οργανισμού που το χρησιμοποιεί. Σύμφωνα με το S-Cube, ευελιξία είναι η ικανότητα ενός συστήματος ή υπηρεσίας να παρουσιάζει μια αποδεκτή συμπεριφορά σε μη φυσιολογικές ή

¹¹ “Adaptation as one of the basic phenomena of biology is the evolutionary process whereby a population becomes better suited to its habitat” (E. Martin, 2010; Williams, 1966).

¹² Από την ιστοσελίδα του S-Cube: “S-Cube, the Software Services and Systems Network, ... integrated research communities to address Service-Oriented computing research challenges. S-Cube established a Network of Excellence enabling Europe to lead in the fields of Service and Software Architectures, Infrastructures and Engineering.” [Πηγή: <http://www.s-cube-network.eu/about-s-cube>]

μη αναμενόμενες συνθήκες, ή όταν το πλαίσιο του μεταβάλλεται (S-Cube, 2008a). Ακόμη η ευελιξία μπορεί να θεωρηθεί ως μια κατάσταση «ισορροπίας ανάμεσα στην αλλαγή και τη σταθερότητα που διασφαλίζει τη διατήρηση της ταυτότητας μιας διαδικασίας¹³» (Regev et al., 2007; Regev and Wegmann, 2006).

Οι Gebauer and Schober (2006) παρουσίασαν ένα θεωρητικό μοντέλο που αναλύει τις παραμέτρους δύο γενικών στρατηγικών για ευελιξία των πληροφοριακών συστημάτων : της «ευελιξίας χρήσης» (flexibility-to-use) που αφορά χαρακτηριστικά των συστημάτων που παρέχονται κατά την εκτέλεση, και της «ευελιξίας αλλαγής» (flexibility-to-change) που αφορά χαρακτηριστικά που αποτελούν «επιλογές αναβάθμισης» των συστημάτων σε μεταγενέστερο όμως χρόνο. Οι Kumar and Stylianou (2013) έδειξαν ότι η οργανωσιακή ευελιξία (organizational flexibility) σχετίζεται με την ευελιξία της πληροφοριακής υποδομής. Χρησιμοποιώντας πραγματικά παραδείγματα, οι συγγραφείς προτείνουν ένα εμπειρικό πλαίσιο εργασίας για την καλύτερη κατανόηση και διαχείριση της ευελιξίας των πληροφοριακών συστημάτων (IS flexibility), η οποία αναπτύχθηκε χρησιμοποιώντας μεθόδους «Θεμελιωμένης Θεωρίας¹⁴» και ανάλυσης περιεχομένου. Επεκτείνοντας μια προγενέστερη θεωρία για την ευελιξία πληροφοριακών συστημάτων, οι Schober and Gebauer (2011) αποτίμησαν την αξία της ευελιξίας, εφαρμόζοντας Ανάλυση Δένδρων Αποφάσεων (Decision Tree Analysis ή DTA), Ανάλυση Πραγματικών Δικαιωμάτων (Real Option Analysis ή ROA), και Εκτίμηση Σαφούς Κινδύνου (explicit risk assessment) βασιζόμενοι σε πειράματα προσομοιώσεων.

Ως ευελιξία των διαδικασιών (process flexibility) θεωρείται η ικανότητα να ανταπεξέρχονται στις προβλέψιμες αλλά και τις απρόβλεπτες μεταβολές, αλλάζοντας ή προσαρμόζοντας εκείνα τα μέρη τους που επηρεάζονται, διατηρώντας όμως παράλληλα την απαιτούμενη διάταξη των μερών εκείνων που δεν επηρεάζονται από τις μεταβολές (Schonenberg, Mans, et al., 2008). Η ευελιξία των διαδικασιών αφορά τόσο το τι πρέπει να μην αλλάξει σε μια διαδικασία, όσο και το τι πρέπει να αλλάξει (Regev and Wegmann, 2005).

Κατά τη διάρκεια ζωής των διαδικασιών απαιτούνται διαφορετικά είδη ευελιξίας. Υπάρχουν αρκετές μελέτες περί ευελιξίας των διαδικασιών σε σχέση με τους παράγοντες που την απαιτούν και τους τρόπους επίτευξής της μέσα στις επιχειρησιακές διαδικασίες (Snowdon et al., 2007; Soffer, 2005; Regev et al., 2006; Carlsen et al., 1997; Heintz et al., 1999; Kumar and Narasipuram, 2006; Webera et al., 2008). Βασιζόμενοι σε μια εκτενή επισκόπηση της βιβλιογραφίας και της παρεχόμενης ευελιξίας από υφιστάμενα εργαλεία, οι Schonenberg, Mans, et al. (2008) προτείνουν την ακόλουθη κατηγοριοποίηση των τύπων ευελιξίας:

- (α) **Ευελιξία κατά τον σχεδιασμό (flexibility by design)**. Αφορά τη δυνατότητα ενσωμάτωσης, κατά το σχεδιασμό, εναλλακτικών «διαδρομών» εκτέλεσης (execution paths) μέσα σε ένα μοντέλο διαδικασίας. Έτσι η επιλογή της καταλληλό-

¹³ "Flexibility is effectively a balance between change and stability that ensures that the identity of the process is retained" (Regev et al. 2007; Regev and Wegmann 2006)

¹⁴ Η Θεμελιωμένη Θεωρία είναι μια συστηματική ποιοτική (qualitative) μεθοδολογία των κοινωνικών επιστημών που δίνει έμφαση στη δημιουργία θεωριών από τα υπάρχοντα δεδομένα, κατά τη διάρκεια της διεξαγωγής των ερευνών. (Faggiolani, 2011; P. Y. Martin & Turner, 1986)

τερης διαδρομής γίνεται κατά το χρόνο εκτέλεσης της διαδικασίας και για κάθε στιγμιότυπο ξεχωριστά.

- (β) **Ευελιξία κατ' απόκλιση (flexibility by deviation)**. Αφορά τη δυνατότητα ενός στιγμιότυπου διαδικασίας να αποκλίνει από την προκαθορισμένη (σύμφωνα με το μοντέλο της διαδικασίας) διαδρομή εκτέλεσης, χωρίς ωστόσο να επηρεάζει το ίδιο το μοντέλο. Η απόκλιση περιλαμβάνει μόνο αλλαγές στη σειρά εκτέλεσης των εργασιών ενός συγκεκριμένου στιγμιότυπου, ενώ δεν επιτρέπει αλλαγές στο μοντέλο αυτής ή στις εργασίες που την απαρτίζουν.
- (γ) **Ευελιξία με ελλειπίες προδιαγραφές (flexibility by underspecification)**. Αφορά την ικανότητα εκτέλεσης ενός μη πλήρως προκαθορισμένου μοντέλου διαδικασίας, δηλαδή ενός μοντέλου που δεν περιέχει επαρκείς πληροφορίες για την άμεση εκτέλεσή του. Αυτό το είδος ευελιξίας δεν απαιτεί την αλλαγή του μοντέλου κατά τον χρόνο εκτέλεσης, παρά μόνο τη συμπλήρωσή του με συγκεκριμένες (concrete) υλοποιήσεις των ελλειπών ή των μη προσδιορισμένων μερών του.
- (δ) **Ευελιξία κατά την μεταβολή (flexibility by change)**. Αφορά την ικανότητα τροποποίησης του μοντέλου διαδικασίας κατά τον χρόνο εκτέλεσης, έτσι ώστε είτε ορισμένα είτε όλα τα εκτελούμενα στιγμιότυπα της διαδικασίας να μεταπίπτουν στο νέο (τροποποιημένο) μοντέλο της διαδικασίας.

3.2 Κατηγοριοποίηση Προσαρμογών

Έχουν προταθεί διάφορες κατηγοριοποιήσεις των προσαρμογών, οι οποίες εστιάζουν σε διαφορετικά χαρακτηριστικά, ανάλογα με την οπτική και τις ανάγκες του κάθε ερευνητή. Ως προς την αιτία, μια προσαρμογή μπορεί να είναι είτε διορθωτική (correction), είτε τροποποιητική (customization), είτε βελτιωτική (optimization) (Erradi et al., 2006). Η πρώτη κατηγορία αφορά περιπτώσεις όπου προέκυψε κάποιο πρόβλημα κατά τη λειτουργία της εφαρμογής ενώ η δεύτερη περιπτώσεις που μεταβλήθηκε το πλαίσιο της εφαρμογής, πιθανώς λόγω νέων απαιτήσεων των χρηστών. Η τρίτη κατηγορία προσαρμογών αφορά περιπτώσεις όπου αναγνωρίστηκε κάποια ευκαιρία για βελτίωση. Ακόμη, ως προς τη φύση τους οι προσαρμογές μπορεί να είναι δυναμικές ή στατικές (dynamic or static), χειροκίνητες ή αυτόματες (manual or automatic) και προβλεπτικές ή αντιδραστικές (proactive or retroactive) (Courbis and Finkelstein, 2005). Ανάλογα με το εύρος εφαρμογής τους μπορεί να θεωρηθούν κάθετες (vertical), όπου μεταβάλλονται συγκεκριμένα βήματα ή λειτουργίες χωρίς να επηρεάζεται η δομή των διαδικασιών, ή οριζόντιες (horizontal), όπου μεταβάλλεται και η δομή των διαδικασιών (Parazoglou, 2008). Πιο πρόσφατα οι Lu, Li and Zhang (2014) διαχώρισαν τις αιτίες των προσαρμογών σε επιχειρησιακές (business reasons) και τεχνικές (technical reasons) και προχώρησαν σε περαιτέρω ανάλυση αυτών. Επιπλέον πρότειναν και κατηγοριοποιήσεις ανάλογα με τον τρόπο υλοποίησης της προσαρμογής (single action, single plan, multiple parallel plans), το είδος της αλλαγής (change of structure of a process, change of contracts with customers, change execution of a process), και τον βαθμό εμπλοκής του ανθρώπου στη διαδικασία (human administrator, human administrator with significant help from computer systems, computer systems with minimal help from human administrator).

Στο S-Cube επισημαίνεται ότι οι εξελικτικές και προσαρμοστικές μέθοδοι και εργαλεία, είναι κομβικής σημασίας για τη δημιουργία εφαρμογών βασισμένων σε υπηρεσίες (SBAs). Στην ορολογία του S-Cube, ο όρος «εξέλιξη» (evolution) αναφέρεται στις μόνιμες αλλαγές ενός συστήματος, όπως αλλαγές στις απαιτήσεις, προδιαγραφές, μοντέλα και άλλα, κατά το χρόνο του σχεδιασμού (π.χ. επανασχεδίαση ή/και επανασύνθεση μιας εφαρμογής). Αντίθετα η «προσαρμογή» αναφέρεται σε αλλαγές σε κάποιο συγκεκριμένο στιγμιότυπο ενός συστήματος κατά το χρόνο εκτέλεσης (π.χ. επανεκτέλεση μιας διαθέσιμης υπηρεσίας ή η αντικατάσταση μιας ακατάλληλης). Στο S-Cube (2009) προτείνεται μια κατηγοριοποίηση των προσαρμογών με βάση το Γιατί (Why), Ποιος (Who), Τι (What), και Πως (How) γίνεται η προσαρμογή υπηρεσιών.

Η πρώτη διάσταση (Γιατί, Why) εξετάζει το σκοπό της προσαρμογής και διακρίνει τις εξής περιπτώσεις. Τελειοποιητικές Προσαρμογές (Perfective Adaptations), δηλαδή βελτιώσεις μιας εφαρμογής ακόμη κι αν αυτή εκτελείται σωστά. Διορθωτικές Προσαρμογές (Corrective Adaptations) που αφορούν την αφαίρεση ελαττωματικών συμπεριφορών και την αντικατάστασή τους με άλλες που παρέχουν ισοδύναμη λειτουργικότητα. Προσαρμοστικές Προσαρμογές (Adaptive Adaptations) που εξετάζουν τροποποιήσεις των εφαρμογών ως ανταπόκριση σε αλλαγές που επηρεάζουν το περιβάλλον τους. Αποτρεπτικές Προσαρμογές (Preventive Adaptations) που εστιάζουν στην αποτροπή μελλοντικών σφαλμάτων ή προβλημάτων σχετικών με τις μη λειτουργικές παραμέτρους (όπως ο χρόνος απόκρισης ή η διαθεσιμότητα) προτού αυτά ανακύψουν. Τέλος, Επεκτατικές Προσαρμογές (Extending Adaptations) που αφορούν επεκτάσεις των εφαρμογών με την προσθήκη νέων λειτουργιών.

Η δεύτερη διάσταση (Ποιος, Who) χαρακτηρίζει ένα πρόβλημα προσαρμογής από την οπτική γωνία των διαφόρων φορέων (actors), ανθρώπων ή λογισμικού, που εμπλέκονται στη διαδικασία, εξετάζοντας το υποκείμενο της προσαρμογής (διάσταση Τι), το θέμα της (π.χ. μοντέλο ποιότητας, λειτουργικότητα, παράμετροι Human-Computer Interaction και άλλα) καθώς και το εύρος αυτής (προσωρινή ή μόνιμη). Ένας φορέας μπορεί να είναι είτε ο αιτών την προσαρμογή (Adaptation Requestor), είτε ο σχεδιαστής της προσαρμογής (Adaptation Designer) ή αυτός που την εκκινεί (Adaptation Initiator) ή τέλος ο εκτελών την προσαρμογή (Adaptation Executor). Η διάσταση Τι (What) αφορά το υποκείμενο που δέχεται την προσαρμογή, ήτοι ένα στιγμιότυπο ή μια κλάση εφαρμογών (SBA instance ή SBA class), το πλαίσιο (context) αυτής ή οι μηχανισμοί της.

Τέλος η διάσταση Πως (How) εξετάζει τους τρόπους επίτευξης της προσαρμογής, μέσω συγκεκριμένων στρατηγικών προσαρμογής (adaptation strategies). Σύμφωνα με τους Bucchiarone et al. (2010) αυτές οι στρατηγικές είναι οι εξής: Αντικατάσταση υπηρεσίας (Service substitution), Επανεκτέλεση υπηρεσίας (Re-execution), Επανα-διαπραγμάτευση (Re-negotiation), Επανα-σύνθεση της εφαρμογής (Re-composition), Αντιστάθμιση (Compensation), Εξέλιξη των εναυσμάτων προσαρμογής (trigger evolution) με εισαγωγή σημείων ελέγχου (για σφάλματα, εξαιρέσεις) στη ροή εργασιών της εφαρμογής ικανών να ενεργοποιήσουν την προσαρμογή της, Καταγραφή και Επικαιροποίηση των πληροφοριών των προσαρμογών και τέλος Αποτυχία προσαρμογής. Επιπλέον αναφέρονται στην ποικιλία των αναγκών για προσαρμογή και στους παράγοντες από τους οποίους εξαρτώνται οι

στρατηγικές προσαρμογής (Lanese et al., 2010; Bucchiarone et al., 2010). Σύμφωνα με αυτή τη διάκριση, όρισαν τις ακόλουθες σχεδιαστικές προσεγγίσεις:

- **Ενσωματωμένες προσαρμογές (Built-in adaptation).** Οι ανάγκες που οδηγούν σε προσαρμογή καθώς και οι δυνατές προσαρμογές, είναι συγκεκριμένες και γνωστές εκ των προτέρων. Έτσι αυτές ενσωματώνονται στην εφαρμογή κατά το χρόνο σχεδιασμού αυτής.
- **Αφαιρετικές Προσαρμογές (Abstraction-based adaptation).** Οι ανάγκες για προσαρμογή είναι συγκεκριμένες αλλά οι πιθανές καταστάσεις που ενεργοποιούν τη διαδικασία προσαρμογής δεν είναι εκ των προτέρων γνωστές. Σε τέτοιες περιπτώσεις, σύμφωνα με τους Bucchiarone et al. (2010) και S-Cube (2008b), είναι σύνηθες να ορίζεται ένα αφαιρετικό μοντέλο (abstract model) των εφαρμογών καθώς και μια γενική στρατηγική προσαρμογής (adaptation strategy) που συγκεκριμενοποιείται κατά το χρόνο εκτέλεσης.
- **Δυναμικές προσαρμογές (Dynamic adaptation).** Οι ανάγκες για προσαρμογή μπορεί να εμφανιστούν κατά την εκτέλεση ενώ δεν είναι γνωστές κατά το χρόνο σχεδιασμού. Σε αυτές τις περιπτώσεις είναι απαραίτητο να παρέχονται εξειδικευμένοι μηχανισμοί που επιλέγουν και ενεργοποιούν τις κατάλληλες στρατηγικές προσαρμογής, ανάλογα με την εκάστοτε κατάσταση και έναυσμα της διαδικασίας προσαρμογής (adaptation trigger).

Στις ενότητες που ακολουθούν δίνεται μια επισκόπηση ορισμένων ευρέως γνωστών ερευνητικών εργασιών που παρουσιάζουν τεχνικές, μοντέλα και ολοκληρωμένες λύσεις που αφορούν δύο βασικές διαστάσεις της προσαρμογής εφαρμογών βασισμένων σε υπηρεσίες (SBA adaptation). Αυτές είναι: (i) κάτω από ποιες συνθήκες πρέπει να συμβεί προσαρμογή και (ii) πως πρέπει να συμβεί η προσαρμογή (ήτοι υλοποίηση της προσαρμογής).

3.3 Εναύσματα προσαρμογών σε Εφαρμογές Υπηρεσιών

Στις υποενότητες που ακολουθούν παρουσιάζονται αρκετές εργασίες που προτείνουν λύσεις και προσεγγίσεις, οι οποίες μεταξύ άλλων αναφέρονται στον τρόπο που ξεκινά μια προσαρμογή. Δηλαδή απαντούν έμμεσα ή πιο άμεσα στα ερωτήματα: (1) πότε υπάρχει ανάγκη για προσαρμογή και (2) με ποιο τρόπο γίνεται αυτή η ανάγκη αντιληπτή. Οι δύο πρώτες υποενότητες εστιάζουν σε προσεγγίσεις που αφορούν κυρίως εναύσματα (triggers) προερχόμενα από την εκτέλεση των διαδικασιών ή των εφαρμογών. Αντίθετα, η τελευταία υποενότητα εστιάζει σε προσεγγίσεις όπου τα εναύσματα είναι οι αλλαγές του πλαισίου.

3.3.1 Παρακολούθηση Εφαρμογών για την Εκκίνηση Προσαρμογών

Πολλές προσεγγίσεις προσαρμογών προκειμένου να ανιχνεύουν τα γεγονότα και τις καταστάσεις που υποδεικνύουν κάποια ανάγκη (για παράδειγμα για διατήρηση υψηλής ποιότητας υπηρεσία), αξιοποιούν διάφορες τεχνικές παρακολούθησης (monitoring). Ειδικότερα, η παρακολούθηση της ποιότητας των υπηρεσιών (QoS monitoring) είναι μια κρίσιμη παράμετρος για την απόφαση προσαρμογής τους ή μη. Η διαδικασία αυτή δεν μπορεί να θεωρηθεί ως απλή, ειδικά σε περιπτώσεις εφαρμογών που χαρακτηρίζονται από

σημαντική διασπορά των υπηρεσιών που τις απαρτίζουν. Αναλόγως με τους στόχους κάθε προσέγγισης προσαρμογής παρακολουθούνται διαφορετικοί τύποι πληροφοριών και γεγονότων και χρησιμοποιούνται διαφορετικές τεχνικές. Έτσι λαμβάνονται πληροφορίες σχετικές με την πρόοδο της εκτέλεσης των εφαρμογών.

Οι Moser et al. (2008) παρουσιάζουν το σύστημα VieDAME που επιτρέπει την παρακολούθηση διεργασιών BPEL (BPEL processes) ως προς ορισμένες παραμέτρους ποιότητας υπηρεσίας (QoS parameters) και την αντικατάσταση των χρησιμοποιούμενων υπηρεσιών με βάση διάφορες στρατηγικές αντικατάστασης (replacement strategies). Συγκεκριμένα, αξιοποιούν στατιστικά δεδομένα προηγούμενων εκτελέσεων των διεργασιών και εκτιμούν το συνολικό χρόνο εκτέλεσης της κάθε μίας. Το σύστημα VieDAME παρέχει πληροφορίες για τη βελτίωση των επιχειρησιακών διαδικασιών όσον αφορά τη χρήση και εκτέλεση υπηρεσιών. Έχουν αναπτύξει έναν εξειδικευμένο μηχανισμό για την αποθήκευση των γεγονότων (μηνύματα SOAP) που ανταλλάσσονται με τις διάφορες υπηρεσίες κατά τη χρήση/εκτέλεση τους από τις διεργασίες BPEL. Κατόπιν αυτά τα στοιχεία χρησιμοποιούνται για τον υπολογισμό των τιμών των παραμέτρων ποιότητας υπηρεσίας (QoS) όπως είναι ο μέσος χρόνος απόκρισης, η ακρίβεια και η διαθεσιμότητα των διαφόρων λειτουργιών.

Σε αρκετές άλλες προσεγγίσεις (Baresi et al., 2004; Baresi, Guinea and Pasquale, 2007; Erradi et al., 2006) τα εναύσματα για προσαρμογή είναι τα διάφορα μηνύματα «αποτυχίας». Τυπικές περιπτώσεις τέτοιων μηνυμάτων είναι οι εξαιρέσεις των εφαρμογών (exceptions), τα προβλήματα του δικτύου και η μη διαθεσιμότητα των υπηρεσιών (Baresi et al., 2004). Ακόμη, «αποτυχία» θεωρείται και η μη συμμόρφωση ορισμένων παραμέτρων με τις απαιτούμενες ή αναμενόμενες τιμές (ή εύρη τιμών). Οι Baresi, Guinea and Pasquale (2007) παρουσίασαν μια λύση για αυτό-ιώμενες (self-healing) BPEL συνθέσεις υπηρεσιών (BPEL compositions) που καλείται Dynamo. Πρόκειται για μία λύση βασισμένη σε *ισχυρισμούς*¹⁵ (assertions) που παρέχει γλώσσες ειδικού σκοπού (WSCoL και WSReL) για τον ορισμό των δραστηριοτήτων παρακολούθησης και ανάνηψης (recovery) από σφάλματα. Οι χρησιμοποιούμενοι ισχυρισμοί, προϋποθέσεις, αποτελέσματα και σταθερές (assertions, pre-conditions, post-conditions και invariants αντίστοιχα), ορίζουν περιορισμούς στη λειτουργικότητα και τις παραμέτρους ποιότητας (QoS) των συνθέσεων υπηρεσιών (δηλαδή των εφαρμογών) και του πλαισίου τους. Οι Spanoudakis et al. (2005) χρησιμοποιούν παραμέτρους με τη μορφή σύνθετων συμπεριφορικών απαιτήσεων (complex behavioural requirements) εκπεφρασμένων με χρήση λογισμού γεγονότων (event calculus). Το κεντρικό μέρος της αρχιτεκτονικής τους είναι ο SCS manager (Service Centric System manager), ο οποίος είναι επιφορτισμένος να επιβλέπει την παρακολούθηση των λειτουργικών και ποιοτικών απαιτήσεων των υπηρεσιών. Επίσης αιτείται την αναζήτηση και εύρεση άλλων εναλλακτικών υπηρεσιών για την αντικατάσταση μη διαθέσιμων ή δυσλειτουργούντων υφιστάμενων υπηρεσιών μιας εφαρμογής. Οι προδιαγραφές των απαιτήσεων, για τις οποίες

¹⁵ Οι ισχυρισμοί (assertions) είναι λογικές εκφράσεις (Boolean expressions) που τοποθετούνται μέσα σε προγράμματα και πρέπει να είναι πάντα αληθείς, εκτός κι αν το πρόγραμμα περιέχει σφάλματα (bugs). Σε μια τέτοια περίπτωση, το πρόγραμμα τυπικά διακόπτεται ή εγείρεται μία εξαίρεση (exception).

δεν υπήρξε συμμόρφωση, χρησιμοποιούνται για τη δημιουργία ερωτημάτων αναζήτησης εναλλακτικών υπηρεσιών που θα αντικαταστήσουν τις μη λειτουργούσες υφιστάμενες.

Οι Erradi et al. (2006) εκφράζουν τις αναμενόμενες τιμές των παραμέτρων ποιότητας υπηρεσίας (QoS) ως πολιτικές ποιότητας (QoS policies) με τη μορφή κανόνων Γεγονότος–Συνθήκης–Ενέργειας¹⁶ (Event–Condition–Action rules ή ECA rules). Όταν ανιχνεύεται μια απόκλιση από τις αναμενόμενες τιμές των παραμέτρων ποιότητας (QoS) τότε εκκινείται μια διαδικασία προσαρμογής που τροποποιεί την εφαρμογή. Χρησιμοποιείται δε μια εξειδικευμένη υπηρεσία για τη συνεχή παρακολούθηση των αλληλεπιδράσεων μεταξύ των υπηρεσιών που απαρτίζουν την εφαρμογή, προκειμένου να ελέγχει ότι εφαρμόζονται οι προκαθορισμένες πολιτικές ποιότητας και επίσης να ανιχνεύει τις όποιες αλλαγές κατάστασης όπως είναι τα σφάλματα. Οι πολιτικές ποιότητας καθορίζουν την επιθυμητή συμπεριφορά του συστήματος με τη μορφή προϋποθέσεων – αποτελεσμάτων (pre-conditions – post-conditions). Αυτές εκφράζουν περιορισμούς στα ανταλλάσσόμενα μηνύματα και τις οριακές τιμές των απαιτούμενων παραμέτρων ποιότητας (QoS guarantees), όπως αυτές ορίζονται σε προκαθορισμένες Συμφωνίες Επιπέδου Υπηρεσίας (Service Level Agreements ή SLAs). Η υπηρεσία παρακολούθησης λαμβάνει τα μηνύματα σφαλμάτων των υπηρεσιών που εκκλήθησαν (invoked services), σύμφωνα με τις διεπαφές που υποδεικνύονται από τις WSDL¹⁷ περιγραφές τους. Με μια παρόμοια προσέγγιση οι Siljee et al. (2005) προτείνουν την DySOA (Dynamic Service-Oriented Architecture), μια αρχιτεκτονική που επεκτείνει τις υπηρεσιο-κεντρικές (service-centric) εφαρμογές για να τις καταστήσει αυτό-προσαρμοζόμενες (self-adaptive). Χρησιμοποιούν τεχνικές παρακολούθησης για τον εντοπισμό και συλλογή πληροφοριών που αφορούν ένα προκαθορισμένο σύνολο παραμέτρων ποιότητας υπηρεσίας (QoS parameters) (όπως ο χρόνος απόκρισης, η συχνότητα σφαλμάτων, η διαθεσιμότητα), χαρακτηριστικών της υποδομής (φόρτος, εύρος ζώνης (bandwidth)) και ακόμη το περιβάλλον εκτέλεσης. Οι συλλεχθείσες πληροφορίες συγκρίνονται με τις αναμενόμενες τιμές που ορίζονται ως συναρτήσεις των παραπάνω παραμέτρων και σε περίπτωση απόκλισης επανα-ρυθμίζεται η εφαρμογή.

Όλες οι προηγούμενες προσεγγίσεις παρακολούθησης ενεργοποιούν «εκ των υστέρων» (reactive approach) τις αντίστοιχες διαδικασίες προσαρμογής, δηλαδή η τροποποίηση της εφαρμογής συμβαίνει μόνο αφού έχει συμβεί κάποιο κρίσιμο γεγονός ή εμφανιστεί ένα πρόβλημα. Μέχρι όμως να υλοποιηθεί μια προσαρμογή, ενδέχεται αρκετά στιγμιότυπα διαδικασιών ή εφαρμογών να έχουν εκτελεστεί με «λάθος» τρόπο. Αυτό μπορεί να αποσοβηθεί χρησιμοποιώντας τεχνικές δοκιμών χρόνου εκτέλεσης (online testing), οι οποίες παρουσιάζονται στην επόμενη ενότητα. Με αυτές καθίσταται δυνατός ο εντοπισμός μη φυσιολογικών καταστάσεων που όμως δεν έχουν οδηγήσει ακόμη στη δημιουργία προβλήματος (και άρα δεν αντιμετωπίζονται με τις παραπάνω μεθόδους). Οι δοκιμές

¹⁶ Οι κανόνες Γεγονότος-Συνθήκης-Ενέργειας (Event-Condition-Action ή ECA rules) πρόκειται για εκφράσεις που κωδικοποιούν τι ενέργειες θα πρέπει να γίνουν όταν ληφθεί ένα συγκεκριμένο γεγονός και ισχύει μια συγκεκριμένη λογική συνθήκη

¹⁷ Web Services Description Language (WSDL)

χρόνου εκτέλεσης επιτρέπουν την πραγματοποίηση των απαραίτητων «προληπτικών» προσαρμογών προτού ανακύψει ένα πραγματικό πρόβλημα.

3.3.2 Δοκιμές Παρεμβολής και Δοκιμές Χρόνου Εκτέλεσης για την Εκκίνηση Προσαρμογών

Η δυναμική φύση των προσανατολισμένων σε υπηρεσίες αρχιτεκτονικών (service-oriented architectures) θέτει νέες προκλήσεις στη δοκιμή και επικύρωση των συστημάτων. Η διαχείριση μεγάλων και κατακεκομμένων συστημάτων, σε συνδυασμό με την ολοένα αυξανόμενη πολυπλοκότητα των υποστηρικτικών τεχνολογιών, αποτελεί μια σημαντική πρόκληση. Σφάλματα μπορεί να συμβούν οπουδήποτε σε ένα κατακεκομμένο σύστημα. Συνεπώς τεχνικές που αξιολογούν την εξέλιξη της συμπεριφοράς των συστημάτων μέσα στο χρόνο θεωρούνται μεγάλης σημασίας. Οι δοκιμές χρόνου εκτέλεσης (online testing) αποτελούν ένα σημαντικό εργαλείο που μπορεί να αποκαλύπτει σφάλματα και αποκλίσεις στην ποιότητα των υπηρεσιών από τα αναμενόμενα επίπεδα, εκτελώντας περιοδικά τις υπηρεσίες ή τις εφαρμογές. Οι προσεγγίσεις που κατά κύριο λόγο χρησιμοποιούνται σήμερα για έλεγχο και επικύρωση εφαρμογών και υπηρεσιών, εστιάζουν κυρίως στις δοκιμές κατά το σχεδιασμό και την ανάπτυξη και αντιστοιχούν στις δοκιμές των παραδοσιακών συστημάτων λογισμικού (Gehlert et al., 2011).

Οι Wang et al. (2004) τονίζουν τη σημασία των δοκιμών χρόνου εκτέλεσης για εφαρμογές βασισμένες στο Web και αναλύουν τις ειδικότερες ανάγκες τους σε σχέση με τις δοκιμές κατά το χρόνο σχεδιασμού. Επίσης προτείνουν μεθόδους για τη μείωση των παρεμβολών (απ' τις δοκιμές) στην κανονική χρήση των υπηρεσιών καθώς και τη βελτίωση της αποδοτικότητάς τους. Οι συγγραφείς θεωρούν την καταγραφή πληροφοριών ως τη βάση για τις δοκιμές του χρόνου εκτέλεσης. Οι Deussen et al. (2003) προτείνουν μια πλατφόρμα για επικύρωση εφαρμογών σε πραγματικό χρόνο (online validation platform), η οποία περιλαμβάνει κι ένα υποσύστημα για δοκιμές χρόνου εκτέλεσης. Επίσης χρησιμοποιούν τη σημειογραφία TTCN-3¹⁸ (Testing and Test Control Notation Version 3) για να υλοποιήσουν το σύστημά τους σε περιβάλλοντα Ενεργών Δικτύων (Active Networks).

Ένα μεταμορφικό (metamorphic¹⁹) είδος δοκιμών χρόνου εκτέλεσης προτείνουν οι Chan et al. (2007), το οποίο χρησιμοποιεί «μαντεία δοκιμών²⁰» (test oracles) που δημιουργούνται κατά το σχεδιασμό των εφαρμογών αλλά για χρήση κατά την εκτέλεση. Συγκεκριμένα, κατά τις δοκιμές στη φάση του σχεδιασμού προσδιορίζεται ένα σύνολο επιτυχών σεναρίων

¹⁸ <http://www.ttcn-3.org/>

¹⁹ Οι μεταμορφικές δοκιμές (Metamorphic Testing, MT) προτάθηκαν από τους T. Y. Chen et al. (1998) και είναι μια τεχνική δοκιμής λογισμικού που μετριάζει το πρόβλημα της έλλειψης μαντειών δοκιμών. Μεταλλάσσουν ένα σενάριο δοκιμών με άγνωστο αναμενόμενο αποτέλεσμα σε ένα "επόμενο" (follow-up) για το οποίο μπορεί να υπολογιστεί το αποτέλεσμα. Για παράδειγμα το αναμενόμενο αποτέλεσμα της συνάρτησης $\sin(x)$ με ακρίβεια 100 ψηφίων δεν είναι γνωστό. Χρειάζεται λοιπόν μια μεταμορφική σχέση (metamorphic relation, MR) για τη συνάρτηση, π.χ. " $\sin(x) = \sin(\pi-x)$ ". Έτσι το ελεγχόμενο πρόγραμμα έχει το ίδιο αναμενόμενο αποτέλεσμα τόσο για το αρχικό σενάριο δοκιμών όσο και το "επόμενο" του.

²⁰ Τα μαντεία δοκιμών είναι μηχανισμοί που χρησιμοποιούνται κατά τη δοκιμή λογισμικού για την εξακρίβωση του αν κάποια δοκιμή ήταν επιτυχής ή μη. Αυτό γίνεται συνήθως συγκρίνοντας το αποτέλεσμα του υπό δοκιμή λογισμικού με ότι το μαντείο θεωρεί ως σωστό αποτέλεσμα για μια δεδομένη περίπτωση δοκιμής (test case).

δοκιμών (test cases) κι από αυτά προκύπτουν οι αντίστοιχες δοκιμές χρόνου εκτέλεσης. Οι δοκιμές αυτές εκτελούνται από μεταμορφικές υπηρεσίες (metamorphic services) που περικλείουν τις υπό έλεγχο υπηρεσίες καθώς και τις υλοποιήσεις των αντίστοιχων μεταμορφικών σχέσεων (metamorphic relations). Έτσι κάθε σφάλμα που αποκαλύπτεται από τις μεταμορφικές δοκιμές (metamorphic testing) οφείλεται σε αποτυχίες που διαπιστώθηκαν κατά το χρόνο εκτέλεσης.

Οι Bai et al. (2007) προτείνουν ένα προσαρμοστικό πλαίσιο δοκιμών (adaptive testing framework), το οποίο μπορεί να μαθαίνει και να βελτιώνει συνεχώς τις ενσωματωμένες στρατηγικές δοκιμών που διαθέτει, χρησιμοποιώντας έναν μηχανισμό «παραθύρου» (windowing mechanism). Οι δοκιμές εκτελούνται κατά τη λειτουργία μιας εφαρμογής και μπορούν να προσαρμοστούν στις αλλαγές τόσο του περιβάλλοντός της όσο και της ίδιας.

Μια άλλη προσέγγιση που σχετίζεται με τις δοκιμές χρόνου εκτέλεσης είναι οι δοκιμές παρεμβολής (regression tests), οι οποίες στοχεύουν στον έλεγχο του αν οι αλλαγές σε ένα σύστημα έχουν αρνητικές επιπτώσεις στην υφιστάμενη λειτουργικότητά του. Μια τυπική διαδικασία είναι η επανεκτέλεση προηγούμενων περιπτώσεων δοκιμών (test cases). Οι Ruth et al. (2007) παρουσιάζουν ένα πλαίσιο εργασίας (framework) που καθιστά εφικτή την εκτέλεση ενός υποσυνόλου μόνο των περιπτώσεων δοκιμών για την επικύρωση Web services, με χρήση της τεχνικής Ασφαλούς Επιλογής Δοκιμών Παρεμβολής (Safe Regression Test Selection ή Safe RTS). Επίσης οι di Penta et al. (2007) προτείνουν μια τεχνική δοκιμών παρεμβολής για Web services που εξετάζει αν οι τιμές των λειτουργικών και μη λειτουργικών παραμέτρων διατηρούνται στα αναμενόμενα επίπεδα με το πέρασμα του χρόνου.

Σύμφωνα με τους Gehlert et al. (2011), όλες οι παραπάνω προσεγγίσεις δεν απαντούν στο ερώτημα πως αξιοποιούνται τα αποτελέσματα των δοκιμών για την πραγματοποίηση προσαρμογών στις εφαρμογές. Σύμφωνα με τους ίδιους, οι βασισμένες σε υπηρεσίες εφαρμογές (SBAs) δύνανται να προσαρμοστούν δυναμικά για διάφορους λόγους όπως οι εξής: (1) επιδίωξη καλύτερης ικανοποίησης των απαιτήσεων των χρηστών, που καλείται «τελειοποιητική προσαρμογή», και (2) επιδιόρθωση και αποτροπή σφαλμάτων, που καλείται «διορθωτική προσαρμογή». Αυτή είναι η πρώτη εργασία που λαμβάνει υπόψη της και τους δύο αυτούς στόχους προσαρμογών και επιχειρεί να αποσοβήσει καταστάσεις με αλληλοσυγκρουόμενες επιδιώξεις. Παρουσιάζουν ακόμη ένα πλαίσιο εργασίας για την ενσωμάτωση των τελειοποιητικών και των διορθωτικών προσαρμογών, η οποία χρησιμοποιεί μεθόδους της τεχνολογίας απαιτήσεων (requirements engineering) για την ενεργοποίηση προσαρμογών του πρώτου είδους και τεχνικές δοκιμών χρόνου εκτέλεσης (online testing) για την ενεργοποίηση προσαρμογών του δεύτερου είδους. Όσον αφορά τη χρήση τεχνικών της τεχνολογίας απαιτήσεων (για τις τελειοποιητικές προσαρμογές), με αυτές επιχειρούν να διασφαλίσουν ότι οι υπηρεσίες παρέχουν τις απαιτούμενες από την εφαρμογή λειτουργικότητες. Αν όμως εμφανιστούν υπηρεσίες νεότερες και καλύτερες από τις ήδη χρησιμοποιούμενες τότε προσαρμόζεται η ροή εργασιών (workflow) της εφαρμογής και οι υφιστάμενες υπηρεσίες αντικαθίστανται από τις νέες. Για να το επιτύχουν αυτό χρησιμοποίησαν την προσέγγιση Tropos (Bresciani et al., 2004; Castro et al., 2002), η οποία διαθέτει ένα φορμαλισμό που επιτρέπει την πραγματοποίηση συλλογισμών (reasoning) για

την επίτευξη των στόχων ενός μοντέλου στόχων, και την ανίχνευση έτσι των επιθυμητών προσαρμογών. Όσον τώρα αφορά τις δοκιμές χρόνου εκτέλεσης (online testing) για τις διορθωτικές προσαρμογές, προτείνουν τη χρήση του πλαισίου εργασίας PROSA (PRO-active Self-Adaptation framework) (Hielscher et al., 2008). Το PROSA εκμεταλλεύεται τεχνικές δοκιμών χρόνου εκτέλεσης για να ενεργοποιήσει προληπτικές (proactive) προσαρμογές. Οι κυριότερες δραστηριότητες που επιτελεί το PROSA είναι η αρχικοποίηση των δοκιμών, η παραγωγή ή επιλογή των περιπτώσεων δοκιμών (test cases), η εκτέλεση των δοκιμών και ο εντοπισμός της πλέον κατάλληλης προσαρμογής.

Ένα μειονέκτημα της εργασίας τους είναι ότι περιορίζεται στην «αντικατάσταση μεμονωμένων υπηρεσιών» ως τον μόνο μηχανισμό για την πραγματοποίηση προσαρμογών σε εφαρμογές (SBAs). Οι προσαρμογές λαμβάνουν χώρα μεταβάλλοντας τις συνδέσεις (bindings) των υπηρεσιών στη ροή εργασιών της εφαρμογής (workflow), χρησιμοποιώντας τεχνικές δυναμικής σύνδεσης (dynamic binding). Η τροποποίηση των δομών ελέγχου και των ροών δεδομένων της εφαρμογής δεν αντιμετωπίζεται σε αυτή την εργασία. Εξετάζουν ωστόσο το αν από τις αλλαγές στην εφαρμογή θα επωφεληθούν όλα ή ορισμένα μόνο από τα ήδη εκτελούμενα στιγμιότυπα της ή αν μόνο τα μελλοντικά στιγμιότυπα (μελλοντικές εκτελέσεις) της εφαρμογής θα μπορέσουν να χρησιμοποιήσουν τις αλλαγές.

3.3.3 Προσαρμογή οδηγούμενη από το Πλαίσιο

Οι οδηγούμενες από το πλαίσιο προσαρμογές (context-driven adaptation) αποτελεί το αντικείμενο πολλών ερευνητικών εργασιών στη βιβλιογραφία, οι οποίες προτείνουν λύσεις τόσο σε θέματα σχεδιασμού όσο και σε θέματα υλοποίησης.

Οι Dai and Liu (2010) βασιζόμενοι στο μοντέλο των Grassi and Sindico (2007) ανέπτυξαν ένα πλαίσιο εργασίας (framework) για υπηρεσίες, ικανό να προσαρμόζεται και αντιδρά στις αλλαγές του πλαισίου. Χρησιμοποιεί τεχνικές ενημέρωσης για το πλαίσιο και αξιοποιεί μια βασισμένη στο OSGi, προσανατολισμένη σε υπηρεσίες πλατφόρμα ανοιχτού κώδικα.

Σε μια άλλη κατεύθυνση οι Moltchanov et al. (2009) παρουσιάζουν την εργασία τους στο έργο C-Cast²¹ (IST project) για «απ' άκρο εις άκρο» (end-to-end) σχεδιασμό συστημάτων παροχής υπηρεσιών και περιεχομένου με ικανότητα επίγνωσης του πλαισίου. Συγκεκριμένα παρουσιάζουν μια «αρθρωτή» (modular) και με πολλαπλά πεδία εφαρμογής (multi-domain) πλατφόρμα ανίχνευσης του πλαισίου, ενώ επιπλέον δείχνουν πως το ζητούμενο πλαίσιο μπορεί να εξαχθεί από διάφορες πηγές δεδομένων με εφαρμογή μηχανισμών επαγωγής (reasoning mechanisms). Τα αρχικά δεδομένα μπορεί να προέρχονται από εικονικούς (virtual), λογικούς (logical) ή πραγματικούς (physical) αισθητήρες που καταγράφουν το περιβάλλον του χρήστη, από πληροφορίες για το προφίλ του καθώς και από πηγές όπως διάφορες βάσεις δεδομένων (π.χ. ημερολόγιο, βιβλίο διευθύνσεων και άλλα). Έπειτα με βάση το εξαχθέν πλαίσιο είναι δυνατή η προσαρμογή της παροχής υπηρεσιών και του περιεχομένου έτσι ώστε να ελαττωθεί η ανάγκη για ανατροφοδότηση (feedback) και να βελτιωθεί η συνολική εμπειρία του χρήστη.

²¹ <http://www.ict-ccast.eu/>

Οι Catarci et al. (2008), στο πλαίσιο του Ευρωπαϊκού έργου Workpad²² (IST project), ανέπτυξαν ένα σύστημα διαχείρισης προσαρμοστικών διαδικασιών, το οποίο χρησιμοποιεί την οντολογία W4H για τη μοντελοποίηση του πλαισίου. Το σύστημα αυτό χρησιμοποιεί μοντέλα διαδικασιών στις οποίες κάθε εργασία (task) σχετίζεται με ένα σύνολο συνθηκών που πρέπει να ισχύουν προκειμένου αυτή να εκτελεστεί. Μια εξειδικευμένη υπηρεσία (context service) παρακολουθεί τις εσωτερικές και εξωτερικές συνθήκες και προσαρμόζει τις διαδικασίες έτσι ώστε να αντιμετωπίζονται οι όποιες αποκλίσεις.

Πέραν των παραπάνω, έχουν αναπτυχθεί μια σειρά από γλώσσες προγραμματισμού εξειδικευμένες σε συγκεκριμένα πεδία εφαρμογής (domain-specific languages). Αυτές επιτρέπουν στους προγραμματιστές να αναπτύσσουν συστήματα που πραγματοποιούν συλλογισμούς υψηλού σημασιολογικού επιπέδου. Για παράδειγμα οι Munnelly et al. (2007) προτείνουν μια γλώσσα που παρέχει δομές για την μοντελοποίηση των προσαρμογών της συμπεριφοράς των εφαρμογών, ως αποτέλεσμα αλλαγών στο πλαίσιο. Με μια προσέγγιση που βασίζεται στη χρήση ενδιάμεσου λογισμικού (middleware), οι David and Ledoux (2005) παρουσιάζουν την πλατφόρμα Wildcat για αυτό-προσαρμοζόμενα συστήματα που χρησιμοποιούν το μοντέλο ECA καθώς και γεγονότα που προέρχονται από μια υπηρεσία με επίγνωση του πλαισίου.

3.4 Προσαρμογή Διαδικασιών και Εφαρμογών

Στη συνέχεια παρουσιάζονται ορισμένες προσεγγίσεις για προσαρμογή επιχειρησιακών διαδικασιών και εφαρμογών, οι οποίες έχουν προταθεί στη βιβλιογραφία. Σημειώνουμε ότι οι περισσότερες από αυτές αφορούν διαδικασίες που βασίζονται στη χρήση υπηρεσιών.

Οι Geebelen et al. (2010) παρουσίασαν ένα πλαίσιο εργασίας όπου οι ροές εργασιών σχεδιάζονται ως πρότυπα (templates) ενώ οι δραστηριότητες που περιλαμβάνονται ορίζονται σε ένα υψηλό μόνο επίπεδο. Οι υλοποιήσεις των δραστηριοτήτων μοντελοποιούνται ως θέματα και αποθηκεύονται σε μια «βιβλιοθήκη» απ' όπου επιλέγονται σύμφωνα με την εκάστοτε πολιτική προσαρμογών.

Οι Xiao et al. (2011) προτείνουν ένα πλαίσιο εργασίας για τη δυναμική προσαρμογή διαδικασιών που βασίζεται σε περιορισμούς (constraint-based framework). Η προσέγγιση χρησιμοποιεί τμήματα διαδικασιών (process fragments) για την επίτευξη συγκεκριμένων εργασιών. Αυτά ενσωματώνονται δυναμικά στις κύριες διαδικασίες, σε καθορισμένα (από το σχεδιασμό) σημεία, τα οποία καλούνται σημεία μεταβλητότητας. Με χρήση περιορισμών μπορεί να προσδιοριστεί ποια τμήματα είναι καταλληλότερα για συγκεκριμένες εργασίες και να χρησιμοποιηθούν στη σύνθεση της τελικής διαδικασίας. Όταν οι περιορισμοί αλλάζουν τότε το σύστημα επιλέγει διαφορετικά τμήματα για αντικατάσταση των προηγούμενων. Ωστόσο αυτό γίνεται για τις νέες μόνο εκτελέσεις των διαδικασιών.

Οι Charfi et al. (2009) επεκτείνουν τη γλώσσα BPEL με τη χρήση πρόσθετων (plug-ins) για τη δημιουργία αυτοπροσαρμοζόμενων διαδικασιών. Κάθε πρόσθετο αποτελείται από θέματα

²² <http://www.dis.uniroma1.it/~workpad/>

παρακολούθησης (monitoring aspects), τα οποία παρακολουθούν το σύστημα ή διαδικασία για να διαπιστώσουν τυχόν ανάγκη για προσαρμογή, και από θέματα προσαρμογής (adaptation aspects), τα οποία πραγματοποιούν την απαραίτητη προσαρμογή. Κατά την εκτέλεση των διαδικασιών, όταν ένα θέμα παρακολούθησης ανιχνεύσει μια ανάγκη για προσαρμογή τότε, χρησιμοποιώντας την τεχνική AO4BPEL, ενσωματώνει το αντίστοιχο θέμα προσαρμογής στην διαδικασία. Τα θέματα προσαρμογής είναι δυνατό να ενσωματωθούν στη διαδικασία κατά την εκτέλεση.

Οι Canfora et al. (2008) προτείνουν ένα πλαίσιο εργασίας όπου οι διαδικασίες ορίζονται ως αφηρημένες ροές εργασιών χωρίς να προσδιορίζεται από ποια υπηρεσία υλοποιείται η κάθε δραστηριότητά τους. Κατά το χρόνο εκτέλεσης των διαδικασιών, οι δραστηριότητες αντιστοιχίζονται με υπηρεσίες που λαμβάνονται από μια «δεξαμενή υπηρεσιών» με βάση παραμέτρους ποιότητας υπηρεσίας (QoS metrics). Αν διαπιστωθεί παραβίαση μιας συμφωνίας επιπέδου υπηρεσίας (SLA) ή επιδείνωση κάποιας παραμέτρου ποιότητας υπηρεσίας (QoS parameter) για κάποια δραστηριότητα, τότε αυτή αντιστοιχίζεται εκ νέου σε άλλη ισοδύναμη υπηρεσία που συμμορφώνεται με τη συμφωνία επιπέδου υπηρεσίας.

Οι Bastida et al. (2008) παρουσίασαν μια προσέγγιση για σύνθεση υπηρεσιών με επίγνωση πλαισίου, χρησιμοποιώντας μια μεθοδολογία έξι βημάτων όπου ορίζεται ένα εκτελέσιμο μοντέλο υπηρεσιών. Η διαδικασία περιλαμβάνει ένα σύνολο μεταβλητών σημείων, τα οποία συνδέονται με συγκεκριμένες υπηρεσίες που επιλέγονται με βάση κανόνες ECA. Οι προσαρμογές εκφράζονται σε μια γλώσσα που έχουν δημιουργήσει γι' αυτόν το σκοπό, η οποία συσχετίζει ενέργειες ανα-προγραμματισμού με πληροφορίες του πλαισίου. Όταν κατά την εκτέλεση ληφθεί ένα γεγονός που σηματοδοτεί την εμφάνιση μιας συνθήκης στο πλαίσιο που σχετίζεται με κάποια ενέργεια ανα-προγραμματισμού τότε το αντίστοιχο (μεταβλητό) σημείο της διαδικασίας συνδέεται σε μία άλλη υπηρεσία.

Οι Rahman et al. (2008) προτείνουν μια προσέγγιση που χρησιμοποιεί κανόνες ECA για τη δημιουργία «συμβάσεων» (contracts) μεταξύ των υπηρεσιών που συμμετέχουν στην εκτέλεση των διαδικασιών. Κατά την εκτέλεση, η προσαρμογή επιτυγχάνεται με αλλαγή των κανόνων ECA των συμβάσεων και την ανάθεση νέων υπηρεσιών ιστού για την εκτέλεση των δραστηριοτήτων (όποτε χρειάζεται). Είναι ακόμη δυνατό να προσαρμοστεί η συμπεριφορά μεμονωμένων υπηρεσιών με εφαρμογή κατάλληλων θεμάτων (aspects).

Οι Colombo et al. (2006) παρουσίασαν μια πλατφόρμα που ενσωματώνει ένα μηχανισμό εκτέλεσης διαδικασιών σε BPEL με το σύστημα Drools για κανόνες ECA. Προτείνουν επίσης μια γλώσσα σύνθεσης υπηρεσιών όπου περιγράφεται η διαδικασία (με BPEL) και ορίζονται οι κανόνες ECA. Κατά την εκτέλεση ελέγχονται οι κανόνες και γίνεται αντιστοίχιση μεταξύ των δραστηριοτήτων των διαδικασιών με υπηρεσίες ιστού που τις υλοποιούν. Κάθε βήμα των διαδικασιών δημιουργεί γεγονότα, τα οποία μέσω των κανόνων συσχετίζονται με τη λογική της διαδικασίας, και έτσι παρακολουθούνται διάφορες παράμετροι εκτέλεσης των διαδικασιών. Η προσαρμογή επιτυγχάνεται με την αντιστοίχιση των δραστηριοτήτων σε ένα proxy, ο οποίος με βάση την κατάσταση της εκάστοτε διαδικασίας προωθεί τα «αιτήματα» εκτέλεσης των δραστηριοτήτων στις κατάλληλες υπηρεσίες.

3.5 Αναζήτηση Υπηρεσιών, Κανόνες Προσαρμογής και Προσαρμογείς

Μια άλλη κατηγορία ερευνητικών εργασιών απαντά στο ερώτημα της προσαρμογής εφαρμογών με χρήση μεθόδων αναζήτησης (εναλλακτικών) υπηρεσιών, μεθόδων σύνθεσης υπηρεσιών, με χρήση προσαρμογών υπηρεσιών (service adapters) ή με χρήση κανόνων προσαρμογής (adaptation rules). Μια τέτοια προσπάθεια είναι η εργασία των Küster et al. (2005) που αναγνωρίζει την ανάγκη για αυτοματοποιημένη σύνθεση υπηρεσιών με χρήση εναλλακτικών προσεγγίσεων. Οι λύσεις που προτείνονται περιλαμβάνουν: (i) τη σειριακή σύνθεση υπηρεσιών που επιλέγονται με αναζήτηση σε ένα γράφο (graph search) ή τη χρήση επαγωγής (forward chaining) ή αναγωγής (backward chaining) ή τη χρήση προσεγγιστικού σχεδιασμού με παρεμβολή²³ (estimated-regression planning), (ii) τη σύνθεση εφαρμογών με βάση τη ζητούμενη συμπεριφορά (behavior-based composition) ή με βάση τις διαθέσιμες λειτουργικές μονάδες (component-based composition), και (iii) τη σύνθεση με βάση τη συλλεχθείσα κατά την αναζήτηση υπηρεσιών ή υπό συνθήκη εφαρμοζόμενων σχεδίων (conditional plans) γνώση.

Σε μια άλλη κατεύθυνση, στόχος των van der Aalst et al. (2009) είναι να παρέχουν ορισμένες βασικές έννοιες σχετικές με τις αλληλεπιδράσεις μεταξύ των υπηρεσιών. Ειδικότερα παρουσιάζουν μια προσέγγιση για την παροχή, αντικατάσταση και βελτίωση υπηρεσιών καθώς και τη δημιουργία προσαρμογών υπηρεσιών. Χρησιμοποιούν τα Open Nets²⁴ ως το βασικό εργαλείο για την περιγραφή και τυποποίηση των υπηρεσιών. Τα Open Nets αποτελούν μια βελτίωση των Petri Nets και μπορούν να μεταφράζονται σε διαδικασίες BPEL. Οι συνθέσεις υπηρεσιών (service compositions) βασίζονται σε ένα σύνολο από μοτίβα αλληλεπιδράσεων υπηρεσιών (service interaction patterns), τα οποία δημιουργούν συσχετίσεις μεταξύ των στιγμιότυπων των υπηρεσιών καθώς και ανταλλαγές μηνυμάτων. Με τη χρήση των μοτίβων, οι van der Aalst et al. (2009) παρουσιάζουν μια προσέγγιση βελτίωσης των (σύνθετων) υπηρεσιών με εφαρμογή κανόνων μετασχηματισμού καθώς και χρήση προσαρμογών υπηρεσιών (service adapters).

Οι Lanese et al. (2010) προτείνουν μια προσέγγιση δυναμικών προσαρμογών που βασίζεται στο συνδυασμό «σημείων» προσαρμογής (adaptation hooks) και κανόνων προσαρμογής (adaptation rules). Τα σημεία προσαρμογής παρέχονται από την ίδια την εφαρμογή και καθορίζουν σε ποια σημεία της μπορεί να γίνουν προσαρμογές. Οι κανόνες προσαρμογής είναι εξωτερικοί και καθορίζουν το πότε και το πως πρέπει να γίνει η προσαρμογή. Παρέχονται δε από ένα διαχειριστή προσαρμογών (adaptation manager) και συμπεριλαμβάνουν περιγραφή της ενέργειας (activity) που πρέπει να προσαρμοστεί, μια συνθήκη που πρέπει να ισχύει για να εφαρμοστεί η προσαρμογή, τον νέο κώδικα της προσαρμοζόμενης ενέργειας καθώς και ένα σύνολο αναγκαίων μεταβλητών καθώς και μη

²³ Estimated-regression planning: Ο προσεγγιστικός σχεδιασμός με παρεμβολή χρησιμοποιεί αναγωγική ανάλυση (backward analysis) της δυσκολίας των στόχων για να οδηγήσει μια επαγωγική αναζήτηση (forward search) στο πεδίο των δυνατών καταστάσεων (situation space). (McDermott, 2002)

²⁴ Τα Open Nets πρόκειται για κοινά P/T Petri Nets που διαθέτουν όμως ένα διακριτό σύνολο από places, τα οποία αναπαριστούν τις διεπαφές των δικτύων με τα περιβάλλοντά τους και καλούνται open places. Αυτά μπορεί να είναι είτε «εισερχόμενα», είτε «εξερχόμενα», είτε και τα δύο. (Baldan et al. 2001; Heckel 2003)

λειτουργικών (non-functional) πληροφοριών. Κατά την εκτέλεση οι κανόνες αντιστοιχούνται με τις ενέργειες που προσαρμόζουν και ενεργοποιούνται είτε από την ίδια την εφαρμογή είτε από τον διαχειριστή προσαρμογών. Ο διαχειριστής προσαρμογών αποφασίζει επίσης τη σειρά εκτέλεσης των κανόνων σύμφωνα με την επίδραση που έχουν στην εφαρμογή.

Οι Spanoudakis et al. (2005) παρουσιάζουν μια προσέγγιση για στατικό και δυναμικό εντοπισμό και σύνθεση υπηρεσιών σε ένα υπηρεσιο-κεντρικό σύστημα (Service Centric System ή SCS). Συνδυάζουν λειτουργικές μονάδες (components) για την παρακολούθηση της συμμόρφωσης του υπηρεσιο-κεντρικού συστήματος με τις απαιτήσεις που έχουν τεθεί, με μονάδες για τον εντοπισμό υπηρεσιών. Οι περιγραφές των απαιτήσεων που παραβιάζονται και ανιχνεύονται από τις μονάδες παρακολούθησης, χρησιμοποιούνται για τη δημιουργία ερωτημάτων (queries) για τον εντοπισμό υπηρεσιών που μπορούν να υποκαταστήσουν τις υφιστάμενες δυσλειτουργούσες υπηρεσίες. Το συγκεκριμένο πλαίσιο εργασίας (framework) προϋποθέτει ότι το υπηρεσιο-κεντρικό σύστημα αναπτύσσεται ως μια συλλογή από Web Services και συντονίζεται (ενορχηστρώνεται) από μια διαδικασία σύνθεσης (composition process) εκπεφρασμένη σε BPEL. Το κεντρικό σημείο της αρχιτεκτονικής είναι ο διαχειριστής SCS (SCS manager) που δημιουργεί τα ερωτήματα εντοπισμού υπηρεσιών. Αυτά προσδιορίζουν τόσο τη διεπαφή των ζητούμενων υπηρεσιών (σε WSDL), όσο και τη συμπεριφορά τους που καθορίζεται από την διαδικασία σύνθεσης σε BPEL με χρήση κανόνων μετασχηματισμού.

Σε μεταγενέστερη εργασία τους, οι Spanoudakis and Zisman (2010) παρουσιάζουν ένα πλαίσιο εργασίας (framework) για τον εντοπισμό υπηρεσιών, το οποίο υποστηρίζει το σχεδιασμό συστημάτων βασισμένων σε υπηρεσίες (service-based systems) αξιοποιώντας προϋπάρχουσες υπηρεσίες και προσαρμόζοντάς τες κατά το χρόνο εκτέλεσης. Η ανάγκη για προσαρμογή μπορεί να οφείλεται στη μη διαθεσιμότητα ή τη δυσλειτουργία ορισμένων από τις υπηρεσίες που χρησιμοποιεί το σύστημα, σε αλλαγές του πλαισίου των υπηρεσιών ή του περιβάλλοντος του συστήματος, ή στην εμφάνιση νέων καλύτερων υπηρεσιών. Τα κύρια στοιχεία του πλαισίου εργασίας (framework) είναι: η μονάδα αίτησης υπηρεσίας (service requestor), η μονάδα επεξεργασίας ερωτημάτων (query processor) και ο ενδιάμεσος προς το μητρώο υπηρεσιών (service registry intermediary). Η μονάδα αίτησης υπηρεσίας παραλαμβάνει αιτήματα από τρίτες εφαρμογές καθώς και πληροφορίες του πλαισίου για τις υπηρεσίες που συμμετέχουν στο σύστημα. Έπειτα ο επεξεργαστής ερωτημάτων αναλύει τα διαφορετικά μέρη κάθε ερωτήματος και τα αποτιμά σε σχέση με τις προδιαγραφές που περιέχονται στα διάφορα μητρώα υπηρεσιών. Ο ενδιάμεσος υποστηρίζει τη χρήση διαφορετικών μητρώων υπηρεσιών καθώς και τον εντοπισμό υπηρεσιών αποθηκευμένων σε διαφορετικούς τύπους μητρώων. Το εν λόγω πλαίσιο εργασίας υποστηρίζει τη σχεδίαση συστημάτων ως μια επαναληπτική διαδικασία. Χρησιμοποιεί σχεδιαστικά μοντέλα δομών και συμπεριφορών, τα οποία με χρήση ερωτημάτων αιτούνται τον εντοπισμό υπηρεσιών που τα υλοποιούν. Κατόπιν οι υπηρεσίες αυτές χρησιμοποιούνται για να αναμορφώσουν το σχεδιασμό των (αρχικών) μοντέλων και επαναληπτικά προκαλούν νέες αναζητήσεις υπηρεσιών.

Οι Hu et al. (2008) παρουσιάζουν μια κατανομημένη προσέγγιση για την αυτόματη δημιουργία μιας σύνθεσης υπηρεσιών. Η σύνθεση βασίζεται στις επιθυμητές εισόδους

(inputs) και εξόδους (outputs) της ζητούμενης διαδικασίας. Οι υποψήφιες υπηρεσίες εντοπίζονται με αναζήτηση σε ένα αποθετήριο υπηρεσιών (service repository) με χρήση κριτηρίων δοσμένων από τον χρήστη. Προτείνεται δε η χρήση ενός μοντέλου αποστολής/εγγραφής (publish/subscribe) για την αναζήτηση.

Οι Lemahieu et al. (2003) πρότειναν έναν τρόπο βελτίωσης της τεχνολογίας των Web services για τον εντοπισμό, κλήση/εκτέλεση και σύνθεση υπηρεσιών, με την εισαγωγή της έννοιας των επιχειρησιακών γεγονότων (business events). Συγκεκριμένα, προτείνουν την αποστολή ειδοποιήσεων με τη μορφή γεγονότων προς όλες (ταυτόχρονα) τις υπηρεσίες που έχουν εκδηλώσει ενδιαφέρον για ένα συγκεκριμένο τύπο γεγονότων. Για την αναπαράσταση των συστημάτων με όρους επιχειρησιακών γεγονότων, την επίδρασή τους σε άλλα επιχειρησιακά αντικείμενα (enterprise objects) και τους σχετικούς επιχειρησιακούς κανόνες (business rules), χρησιμοποιήθηκε η MERODE, μια αντικειμενοστραφή μεθοδολογία ανάλυσης και σχεδιασμού. Παρότι ακολουθεί μια αντικειμενοστραφή προσέγγιση, τα επιχειρησιακά γεγονότα θεωρούνται ως ανεξάρτητες έννοιες και η σχέση τους με τους διάφορους τύπους αντικειμένων ορίζεται σε ένα πίνακα αντικειμένων – γεγονότων (Object – Event Table ή OET). Τέλος, η έννοια των γεγονότων ενσωματώνεται σε μια γλώσσα περιγραφής Web services, συμβατή με τα πρότυπα SOAP, WSDL και UDDI.

Οι Aydin et al. (2008) χρησιμοποιούν μια λογική γλώσσα ορισμού ενέργειας – αποτελέσματος (action – effect) που καλείται Λογισμός Γεγονότων (Event Calculus), προκειμένου να αυτοματοποιήσουν τη δημιουργία και εκτέλεση συνθέσεων Web services. Επιπλέον, αν υπάρχουν πολλαπλά εναλλακτικά σχέδια (συνθέσεων), αυτά μπορούν να ενωθούν σε έναν ενιαίο γράφο και το καλύτερο από αυτά να επιλεγεί από τον μηχανισμό εκτέλεσης. Επίσης παρουσίασαν έναν τρόπο μετασχηματισμού από OWL-S σε Event Calculus, έτσι ώστε να είναι δυνατός ο σχεδιασμός συνθέσεων Web Services με χρήση επιπλέον τυπολογιών (κι όχι μόνο σε Event Calculus).

Ο Juric (2010) προτείνει μια λύση που υποστηρίζει τη χρήση επιχειρησιακών γεγονότων και εννοιών της οδηγούμενης από γεγονότα αρχιτεκτονικής (Event-Driven Architecture) στην υπηρεσιοστρεφή αρχιτεκτονική (SOA). Αυτό επιτυγχάνεται προτείνοντας συγκεκριμένες επεκτάσεις στις γλώσσες WSDL και BPEL. Οι επεκτάσεις της WSDL είναι δομές XML που επιτρέπουν σε μια υπηρεσία να παράγει και λαμβάνει γεγονότα, διασφαλίζοντας όμως παράλληλα τη συμβατότητα με προηγούμενες εκδόσεις της γλώσσας (backward compatibility). Οι επεκτάσεις της BPEL προσθέτουν τη δυνατότητα ενορχήστρωσης υπηρεσιών με βάση τα γεγονότα. Επίσης προσθέτουν νέες ενέργειες (activities) για την ενεργοποίηση και τη λήψη γεγονότων, επεκτάσεις για χειριστές γεγονότων και σφαλμάτων (event and fault handlers), για μεταβλητές (variables) και για δείκτες.

Οι Lee et al. (2006) προτείνουν την ανάπτυξη της WS-ECA, μιας γλώσσας περιγραφής κανόνων ECA (Event – Condition – Action) βασισμένης στην XML, προκειμένου να υποστηρίξει αυτόνομες αλληλεπιδράσεις μεταξύ προσανατολισμένων σε υπηρεσίες συσκευών (service-oriented devices), σε περιβάλλοντα «πανταχού παρόντων» (ubiquitous) και «διάχυτων» (pervasive) υπολογιστικών δικτύων. Οι κανόνες είναι ενσωματωμένοι στις κατανεμημένες συσκευές, οι οποίοι καλούν τις απαραίτητες υπηρεσίες του δικτύου όταν

ενεργοποιούνται από κάποια εσωτερικά ή εξωτερικά γεγονότα. Τέτοιοι κανόνες προτείνονται κι από άλλους (για παράδειγμα Jung et al. (2007) και Park et al. (2008)) και μπορούν να εκφράσουν την επιθυμητή προσαρμοστικότητα σε ενορχηστρώσεις υπηρεσιών με «χαλαρή» δομή (loosely coupled service orchestrations).

3.6 Προτάσεις Προσαρμογών για Διαδικασίες και Εφαρμογές

Η αξιοποίηση των προτιμήσεων των χρηστών (user preferences) και των χαρακτηριστικών των υπηρεσιών (service properties) μπορεί να συμβάλει στην επιλογή των υπηρεσιών. Ειδικότερα, τεχνικές από τα συστήματα προτάσεων (recommender systems) έχουν χρησιμοποιηθεί σε ερευνητικά έργα για τη βελτίωση της επιλογής Web services. Μία από τις πρώτες εργασίες σε αυτόν τον τομέα είναι αυτή των Sreenath and Singh (2004) που προσάρμοσαν τις δύο κλασικές κατηγορίες αλγορίθμων συστημάτων προτάσεων, δηλαδή τη βασισμένη στο περιεχόμενο διήθηση πληροφοριών (content-based filtering) και τη συνεργατική διήθηση πληροφοριών (collaborative filtering), για επιλογή υπηρεσιών. Επίσης πρότειναν μια προσέγγιση αξιολόγησης των παρόχων υπηρεσιών (service providers) βασισμένη στη χρήση «πρακτόρων» (agents) που συνεργάζονται για να αξιολογήσουν τους παρόχους. Σύμφωνα με την προσέγγιση αυτή, οι πράκτορες βαθμολογούν ο ένας τον άλλο και αυτόνομα αποφασίζουν τι βάρος θα δίνουν στις προτάσεις άλλων πρακτόρων. Ο αλγόριθμος με τον οποίο πραγματοποιούν τους συλλογισμούς τους λειτουργεί στο πλαίσιο ενός «εννοιολογικού δικτύου» (concept lattice), το οποίο επιτρέπει τον εντοπισμό σχετικών πρακτόρων και την παροχή προτάσεων υπηρεσιών. Οι Mehta et al. (2004) θεωρούν ότι η σημασιολογία και το συντακτικό δεν επαρκούν πάντα για τον εντοπισμό υπηρεσιών που ικανοποιούν τις ανάγκες των χρηστών. Προσθέτουν έτσι δύο επιπλέον διαστάσεις: την ποιότητα και τα μοτίβα χρήσεων. Οι Blake and Nowlan (2007) προτείνουν έναν τρόπο βαθμολογίας των προτάσεων Web services με βάση το ταίριασμα αλφαριθμητικών (strings) που συλλέγονται από τις συνεδρίες (sessions) του χρήστη και τις περιγραφές των Web services. Κατόπιν με βάση τη βαθμολόγηση αυτή, η προσέγγισή τους κρίνει αν ένας χρήστης ενδιαφέρεται για κάποια υπηρεσία. Οι Maamar et al. (2005) προτείνουν ένα μοντέλο για το πλαίσιο (context) των αλληλεπιδράσεων μεταξύ Web services και τονίζουν ότι οι πόροι που χρησιμοποιούν (τα Web services) επιδρούν στο πως αυτά προσωποποιούνται.

Υπάρχουν ακόμη αρκετές προσεγγίσεις που χρησιμοποιούν τη συνεργατική διήθηση (collaborative filtering) για την πρόταση Web services. Σε αυτές τις προσεγγίσεις οι τιμές της Ποιότητας Υπηρεσίας (QoS values) για έναν χρήστη, εκτιμώνται με βάση τις καταγραφές τιμών των ίδιων παραμέτρων ποιότητας υπηρεσίας άλλων χρηστών, οι οποίοι όμως έχουν παρόμοιες προηγούμενες εμπειρίες ποιότητας υπηρεσίας, για συγκεκριμένα Web services.

Οι Shao et al. (2007) προτείνουν έναν αλγόριθμο συνεργατικής διήθησης που βασίζεται στα χαρακτηριστικά του χρήστη, για την πρόβλεψη των τιμών της ποιότητας υπηρεσίας. Οι Zheng et al. (2009) παρουσιάζουν έναν υβριδικό αλγόριθμο συνεργατικής διήθησης που βασίζεται τόσο στα χαρακτηριστικά του χρήστη όσο και των υπηρεσιών, προκειμένου να προτείνει ένα Web service. Οι Rong et al. (2009) προτείνουν ένα πλαίσιο εργασίας (framework) για την κατάταξη Web services, όπου πρώτα πρέπει να αναγνωριστεί μια

ομάδα χρηστών με παρόμοια ενδιαφέροντα. Έπειτα εφαρμόζονται τεχνικές «εξόρυξης» κανόνων συσχέτισης (association rule mining) σε όλες τις αλληλεπιδράσεις των συνθέσεων Web services (WS compositions) που σχετίζονται με την παραπάνω ομάδα χρηστών. Συνδυάζοντας τέλος την ομάδα χρηστών με τους κανόνες συσχέτισης που «εξορύχθηκαν» (mined), επιτυγχάνεται ένας προσωποποιημένος μηχανισμός βαθμολόγησης των Web services. Οι Toma et al. (2009) εξερευνούν την ιδέα της χρήσης κοινωνικών επισημάνσεων (social annotations) από την διαδικτυακή υπηρεσία del.icio.us για την κατάταξη Web services. Οι Chen et al. (2010) προτείνουν ένα υβριδικό αλγόριθμο συνεργατικής διήθησης που βασίζεται σε παρατηρήσεις της ποιότητας υπηρεσίας, η οποία θεωρείται ως το σύνολο των αντιληπτών στον χρήστη (user-perceived) χαρακτηριστικών, σε σχέση με τη φυσική τοποθεσία του χρήστη (physical location). Πρώτα ομαδοποιούνται οι χρήστες με βάση την τοποθεσία τους και τις ομοιότητές τους ως προς την ποιότητα υπηρεσίας, με χρήση ιστορικών δεδομένων. Έπειτα εντοπίζονται υπηρεσίες που εξαρτώνται από την περιοχή (region-sensitive services). Τέλος χρησιμοποιείται μια τροποποιημένη εκδοχή της μεθόδου του «κοντινότερου γείτονα» (nearest neighbor) για την πρόβλεψη της ποιότητας υπηρεσίας των υποψήφιων για πρόταση Web services προς ένα χρήστη. Η μέθοδος αξιοποιεί ιστορικές πληροφορίες για την ποιότητα υπηρεσίας, οι οποίες συλλέχθηκαν από άλλους χρήστες που ανήκουν σε ισχυρά συσχετισμένες περιοχές (highly correlated regions).

Πέραν από τα συστήματα υπηρεσιών (service-based systems), τα συστήματα προτάσεων χρησιμοποιούνται για την επίτευξη ευελιξίας στη διαχείριση διαδικασιών (process management) αλλά και συστημάτων ροής εργασιών (workflow systems). Οι Adams et al. (2003) ορίζουν κάθε βήμα μιας διαδικασίας ως μια «μεταβλητή». Αυτή αντικαθιστάται δυναμικά (κατά το χρόνο εκτέλεσης) από μια εργασία που λαμβάνεται από έναν κατάλογο με ορισμούς ροών εργασιών, με βάση το πλαίσιο. Επιπρόσθετα, προτάσεις μπορούν να ληφθούν και με βάση μια αδρή δομή εργασίας (rough task structure) (Eichholz et al. 2004). Σε αντίθεση με προσεγγίσεις που στηρίζονται σε μοντέλα προδιαγεγραμμένων διαδικασιών που μπορεί να μην ενεργοποιηθούν ποτέ στην πράξη, η δημιουργία προτάσεων μπορεί να βασιστεί σε βέλτιστες πρακτικές που μοιράζονται οι χρήστες μέσα σε μια επιχείρηση (Stoitsev et al. 2007). Οι Pesic and van der Aalst (2006) και Pesic et al. (2007) παρέχουν προτάσεις βασιζόμενοι σε παρελθούσες εμπειρίες και σε ένα συγκεκριμένο στόχο διαδικασίας. Αυτό το επιτυγχάνουν συγκρίνοντας ένα τρέχον στιγμιότυπο διαδικασίας με καταγραφές παρελθουσών εκτελέσεων (logs), επιλέγοντας τις εκτελέσεις εκείνες που επιτυγχάνουν τον ίδιο στόχο. Μια παρόμοια προσέγγιση προτείνεται και από τους Schonenberg, Weber, et al. (2008) όπου οι προτάσεις παράγονται με βάση την ομοιότητα με παρελθούσες εκτελέσεις της διαδικασίας, λαμβάνοντας υπόψη συγκεκριμένους στόχους βελτιστοποίησης. Μια άλλη προσέγγιση ακολουθείται από τους Almeida et al. (2004) όπου η δημιουργία προτάσεων βασίζεται σε μια οντολογία και σε σημασιολογικούς κανόνες που παράγουν εναλλακτικές διαδικασίες ή βήματα διαδικασιών, αν η εκτέλεση ενός στιγμιότυπου διαδικασίας αποτυγχάνει να προχωρήσει. Οι Vanderfeesten et al. (2008) ακολουθούν μια προσέγγιση όπου το επόμενο βήμα μιας διαδικασίας επιλέγεται με βάση μια στρατηγική, για παράδειγμα επιλογή με βάση το χαμηλότερο κόστος ή τον μικρότερο χρόνο επεξεργασίας, βασιζόμενοι στις διαθέσιμες πληροφορίες για την εκάστοτε

περίπτωση (case). Οι Dorn et al. (2010) παρουσιάζουν μία αυτό-ρυθμιζόμενη (self-adjusting) προσέγγιση για την παροχή προτάσεων προσαρμογής διαδικασιών, εξαρτώμενων από το πλαίσιο (context-sensitive process recommendations). Η παραγωγή των προτάσεων γίνεται με βάση την ανάλυση των συμπεριφορών των χρηστών, συλλογικών διαδικασιών (crowd processes) και συνεχούς εφαρμογής τεχνικών εντοπισμού διαδικασιών. Ειδικότερα, η προσέγγιση αυτή παρέχει προτάσεις που έχει «μάθει» από προηγούμενες εκτελέσεις διαδικασιών του ίδιου χρήστη και τις συσχετίζει με αποφάσεις για τη διαδικασία που λαμβάνονται από όλους τους εμπλεκόμενους χρήστες στο συγκεκριμένο τύπο διαδικασιών.

Η σημασία των πληροφοριών του πλαισίου (contextual information) για την πρόταση υπηρεσιών και διαδικασιών έχει υπογραμμιστεί από τους ερευνητές των σχετικών επιστημονικών πεδίων. Φερ' ειπείν στην προσέγγισή τους οι Chen et al. (2010), λαμβάνουν υπόψη την τοποθεσία των χρηστών προκειμένου να παρέχουν προτάσεις υπηρεσιών με βάση την ποιότητα υπηρεσίας (QoS). Επιπλέον, σε ορισμένες προσεγγίσεις προτάσεων (προσαρμογής) διαδικασιών, λαμβάνονται υπόψη ο στόχος της τρέχουσας διαδικασίας καθώς και πληροφορίες του πλαισίου για την παροχή των προτάσεων. Οι διάφορες προσεγγίσεις χρήσης των πληροφοριών του πλαισίου για την παροχή προτάσεων μπορούν χονδρικά να κατηγοριοποιηθούν σε δύο ομάδες. (1) Τις προσεγγίσεις που βασίζονται στη χρήση ερωτημάτων και αναζητήσεων οδηγούμενων από το πλαίσιο, και (2) τις προσεγγίσεις που βασίζονται στην «εκμείωση» και εκτίμηση των προτιμήσεων του χρήστη σε σχέση με το πλαίσιο (Adomavicius and Tuzhilin, 2010).

3.7 Προσαρμογή διαδικασιών οδηγούμενη από γεγονότα & επίγνωση πλαισίου

Οι εργασίες που εξετάστηκαν παραπάνω εστιάζουν σε συγκεκριμένα θέματα ή προβλήματα ωστόσο σπάνια αντιμετωπίζουν το θέμα της οδηγούμενης από γεγονότα προσαρμογής διαδικασιών συνολικά. Μια εργασία όμως που ασχολείται με πολλές από τις παραμέτρους του θέματος της παρούσας διατριβής, όπως αυτό παρουσιάστηκε στην εισαγωγή είναι η διδακτορική διατριβή του Gabriel Hermsillo. Για το λόγο αυτό θα παρουσιαστεί ιδιαίτερα.

Ο Hermsillo (2012) προτείνει το CEVICHE (Complex Event processing for Context-adaptive processes in pervasive and Heterogeneous Environments), ένα πλαίσιο εργασίας για προσαρμογή επιχειρησιακών διαδικασιών, οδηγούμενη από γεγονότα και με επίγνωση του πλαισίου, για διάχυτα και ετερογενή περιβάλλοντα. Το εν λόγω πλαίσιο εργασίας χρησιμοποιεί επεξεργασία σύνθετων γεγονότων (CEP) για να λαμβάνει τις απαραίτητες πληροφορίες πλαισίου και μάλιστα από οποιαδήποτε πηγή του περιβάλλοντος (όχι μόνο για συγκεκριμένα σημεία των επιχειρησιακών διαδικασιών). Εν συνεχεία χρησιμοποιεί τις πληροφορίες αυτές προκειμένου να προσαρμόζει τις επιχειρησιακές διαδικασίες κατά την εκτέλεσή τους, με βάση ένα σύνολο κανόνων προσαρμογής από ένα αποθετήριο κανόνων (rules repository). Μια σημαντική συνεισφορά του CEVICHE είναι ο ορισμός και η υλοποίηση ενός μηχανισμού άρσης των προσαρμογών. Όπως αναφέρει ο συγγραφέας, η άρση των προσαρμογών δεν είναι μια τετριμμένη εργασία και εύκολα μπορεί να οδηγήσει σε λάθη, ανεπιθύμητες καταστάσεις και ασταθείς διαδικασίες.

Το πεδίο της επεξεργασίας σύνθετων γεγονότων ακόμη εξελίσσεται ενώ δεν υπάρχει κάποιο πρότυπο για τον τρόπο ορισμού των γεγονότων και των κανόνων επεξεργασίας τους. Για το λόγο αυτό, στο CEVICHE προτείνεται και χρησιμοποιείται η Adaptive Business Process Language (ABPL), μια γλώσσα που επιτρέπει να αποτυπώνονται οι κανόνες επεξεργασίας των σύνθετων γεγονότων και οι κανόνες προσαρμογής των διαδικασιών. Οι κανόνες επεξεργασίας σύνθετων γεγονότων, με χρήση κατάλληλου plug-in, μετατρέπονται στη γλώσσα που κατανοεί ο μηχανισμός CEP που χρησιμοποιείται στην εκάστοτε υλοποίηση του CEVICHE. Από την άλλη, οι κανόνες προσαρμογής αποθηκεύονται σε ένα (εσωτερικό) αποθετήριο κανόνων απ' όπου ο διαχειριστής προσαρμογών (είναι μέρος του CEVICHE) επιλέγει εκείνους που θα πρέπει να εφαρμοστούν, ανάλογα με τις πληροφορίες του πλαισίου.

Η εφαρμογή των κανόνων προσαρμογής γίνεται ως εξής. Οι ορισμοί των επιχειρησιακών διαδικασιών λαμβάνονται ως μοντέλα σε BPEL και μεταφράζονται σε γράφους εκτέλεσης αποτελούμενους από αντικείμενα, σύμφωνα με την αρχιτεκτονική σύνθεσης υπηρεσιών (Service Composition Architecture ή SCA components). Η αρχιτεκτονική αυτή επιτρέπει την αντιστοίχιση (bind) των αντικειμένων σε συγκεκριμένες υλοποιήσεις (ως υπηρεσίες ιστού), κατά την εκτέλεση. Ο διαχειριστής προσαρμογών μπορεί να παρεμβαίνει στην αντιστοίχιση μεταξύ των αντικειμένων του γράφου εκτέλεσης και των υλοποιήσεών τους (υπηρεσίες ιστού), και να προσθέτει, αφαιρεί, αντικαθιστά ή τροποποιεί αυτά δυναμικά. Στην υλοποίηση του CEVICHE χρησιμοποιήθηκαν από κοινού τα EasyBPEL²⁵ και EasyViper²⁶ για τον προσαρμοστικό μηχανισμό εκτέλεσης επιχειρησιακών διαδικασιών, ενώ για τον μηχανισμό CEP μπορεί να χρησιμοποιηθεί οποιοδήποτε από τα Esper, Etalis ή Drools μαζί με το αντίστοιχο plug-in για τη μετάφραση των κανόνων επεξεργασίας σύνθετων γεγονότων από την ABPL στη γλώσσα του εκάστοτε μηχανισμού.

Το CEVICHE δεν παρέχει επίγνωση κατάστασης κατά την έννοια που παρουσιάστηκε στο κεφάλαιο 2, δηλαδή ως συνδυασμό του πλαισίου με μια υπόθεση για τον τρόπο με τον οποίο το πλαίσιο σχετίζεται με τους παράγοντες εκείνους που ενδιαφέρουν σχεδιαστές και εφαρμογές. Δεν κάνει επίσης κάποια αναφορά στον κύκλο διαχείρισης καταστάσεων των Jakobson et al. (2007). Αντίθετα φαίνεται να χρησιμοποιεί τον όρο πιο ελεύθερα ως έναν άλλο τρόπο για να αναφέρεται στην επίγνωση του πλαισίου. Ακόμη, το CEVICHE δε φαίνεται να μπορεί να παρέχει αφ' εαυτού προτάσεις προσαρμογής των επιχειρησιακών διαδικασιών. Έτσι δεν δίνει την ευκαιρία στους υπευθύνους των διαδικασιών να επιλέγουν το αν και το ποια προσαρμογή θα γίνει. Επίσης, δεν ενημερώνει άλλα συστήματα για την ενέργεια αυτή.

Στην ενότητα 4.3.8.1, αφού πρώτα παρουσιαστεί η προσέγγιση που προτείνει η παρούσα διατριβή, γίνεται μια σύγκριση αυτής με το CEVICHE.

²⁵ <https://research.petalslink.org/display/easybpel/EasyBPEL+Overview>

²⁶ <https://research.petalslink.org/display/easyviper/EasyViper+Overview>

4 Πρόταση της Διατριβής

Στα προηγούμενα κεφάλαια παρουσιάστηκαν τα ευρήματα της βιβλιογραφικής έρευνας που πραγματοποιήθηκε σε επιστημονικά πεδία που σχετίζονται έμμεσα ή άμεσα με την προσαρμοστικότητα ως χαρακτηριστικό συστημάτων, διαδικασιών και εφαρμογών υπηρεσιών (SBAs). Στο πλαίσιο αυτό εξετάστηκαν αρκετές προσεγγίσεις και εργασίες που άπτονται διαφόρων ειδικών θεμάτων και προτείνουν απαντήσεις στα αντίστοιχα ζητήματα.

Στο παρόν κεφάλαιο επιχειρείται μια δημιουργική σύνθεση στοιχείων από την παραπάνω έρευνα που θα οδηγήσει στην ακριβή διαμόρφωση του προβλήματος της διατριβής και θα θέσει τα ερευνητικά ερωτήματα στα οποία απαντά. Κατόπιν παρουσιάζεται το περίγραμμα της προτεινόμενης λύσης και επισημαίνονται οι καινοτομίες της. Το κεφάλαιο κλείνει με μια αντιστοίχιση των δημοσιεύσεών μας με τις καινοτομίες και τα μέρη της προτεινόμενης λύσης.

4.1 Διαμόρφωση του Προβλήματος

Όπως έχει ήδη αναφερθεί από την εισαγωγή, οι επιχειρησιακές διαδικασίες αποτελούν ένα σημαντικό κεφάλαιο των επιχειρήσεων και οργανισμών. Ειδικότερα, οι αυτοματοποιημένες (ή ημι-αυτοματοποιημένες) επιχειρησιακές διαδικασίες καθίστανται ένα ολόένα και πιο σημαντικό περιουσιακό στοιχείο αυτών. Σύμφωνα με την εταιρεία Kissflow²⁷, οι 5 κυριότεροι λόγοι είναι η εξορθολογισμένη επικοινωνία μέσα στη διαδικασία, η ξεκάθαρη λογοδοσία (απόδοση ευθυνών), η ελαχιστοποίηση του κόστους λόγω ανθρωπίνων λαθών και ανεπάρκειας, η σταδιακή απόκτηση επίγνωσης σχετικά με την εξέλιξη και τις συνέπειες των διαδικασιών και η καθιέρωση μιας ξεκάθαρης ιεραρχίας στη λήψη αποφάσεων (KISSFLOW, 2015). Πέραν όμως τούτων βασικοί λόγοι μπορεί ακόμη να είναι η βελτιωμένη δυνατότητα διοίκησης, ποιοτικού ελέγχου και ελεγχιμότητας (accountability) των διαδικασιών, η αύξηση της ακρίβειας και της ταχύτητας διεκπεραίωσης των εργασιών, η

²⁷ Εταιρεία παραγωγής λογισμικού αυτοματοποίησης επιχειρησιακών διαδικασιών [<https://kissflow.com/>]

σαφήνεια και η σχετικά ευκολότερη τεκμηρίωση, η ελάφρυνση του προσωπικού, η ελάττωση των αναγκών εκπαίδευσης και η μείωση της καμπύλης εκμάθησης, η διασφάλιση της ικανοποίησης των πελατών, η βελτίωση της συμμόρφωσης σε κανονισμούς και τις εταιρικές νόρμες, και επίτευξη επιχειρησιακής ευελιξίας (Leeming, 2004). Είναι επόμενο οι διοικήσεις οργανισμών και επιχειρήσεων να δίνουν έμφαση στην ανάπτυξη και συντήρηση μηχανογραφημένων και αυτοματοποιημένων επιχειρησιακών διαδικασιών για τις μονάδες που διοικούν.

Φυσικά τα παραπάνω έχουν νόημα όταν οι επιχειρησιακές διαδικασίες λειτουργούν σωστά, επιτυγχάνουν τους στόχους για τους οποίους σχεδιάστηκαν και παράγουν τα επιθυμητά αποτελέσματα. Στην πράξη όμως ανακύπτουν πολλοί λόγοι για τους οποίους οι διαδικασίες αποκλίνουν και δεν επιτυγχάνουν τους στόχους τους, είτε λόγω δυσλειτουργιών και αστοχιών, είτε επειδή άλλαξαν οι συνθήκες του περιβάλλοντος ή οι απαιτήσεις των μετεχόντων σε αυτές (stakeholders). Τότε απαιτούνται κατάλληλες παρεμβάσεις και τροποποιήσεις τόσο στη δομή τους, όσο και στη λειτουργικότητά τους ή ακόμη και στους στόχους τους, προκειμένου αυτές να συνεχίσουν να λειτουργούν κανονικά, να παράγουν τα αναμενόμενα αποτελέσματα και να καλύπτουν τις εκάστοτε ανάγκες των μετεχόντων. Σε μια οριακή περίπτωση οι προσαρμογές θα μπορούσαν να αφορούν ακόμη και κατάργηση υφισταμένων ή εισαγωγή νέων διαδικασιών λόγω αλλαγών του περιβάλλοντος ή των απαιτήσεων.

Ειδικότερο ενδιαφέρον παρουσιάζουν οι οδηγούμενες από γεγονότα επιχειρησιακές διαδικασίες (event-driven business processes) επειδή αξιοποιούν τεχνολογίες αιχμής που επιτρέπουν την εκμετάλλευση των δυνατοτήτων που παρέχει η Οδηγούμενη από Γεγονότα Αρχιτεκτονική (EDA). Ειδικότερα, η χρήση της EDA παρέχει αρχιτεκτονική ευελιξία, απλούστευση της δομής και αποσύζευξη (decoupling) των εμπλεκόμενων συστημάτων. Ακόμη παρέχει επεξεργασία σε πραγματικό χρόνο, υψηλές επιδόσεις και αυξημένη δυνατότητα κλιμάκωσης (Norberg, 2010).

Στο πλαίσιο αυτής της διατριβής θεωρούμε ότι τόσο οι επιχειρησιακές διαδικασίες όσο και το περιβάλλον τους, παράγουν και ανταλλάσσουν γεγονότα σχετικά με την κατάστασή τους και τις συνθήκες που επικρατούν. Η υπόθεση αυτή δεν είναι αίολη και αυθαίρετη. Αντιθέτως, τα τελευταία χρόνια ολοένα και περισσότερα προϊόντα και υπηρεσίες, ακόμη και ευρείας κατανάλωσης, είναι σε θέση να παράγουν και δημοσιεύουν τα δικά τους γεγονότα ενώ κάποια μπορούν να αξιοποιούν και γεγονότα που προέρχονται από τρίτους. Αυτή η τάση αποτελεί μέρος μιας ιδέας που είναι γνωστή ως *Διαδίκτυο των Πραγμάτων* ή *Internet of Things (IoT)*, το οποίο ευαγγελίζεται έναν κόσμο όπου υπολογιστές, συσκευές, τοποθεσίες (κτίρια ή πόλεις), μέσα μεταφοράς, wearables και άλλα, θα διασυνδέονται μεταξύ τους και με τους χρήστες μέσω του Διαδικτύου, μεταμορφώνοντας τον τρόπο που οι άνθρωποι αλληλεπιδρούν και βιώνουν το πραγματικό τους περιβάλλον. Αυτό καθίσταται εφικτό μέσω ταχέως αναπτυσσόμενων τεχνολογιών που παρέχουν φθηνή και ισχυρή επεξεργαστική ικανότητα, δυνατότητα διασύνδεσης οπουδήποτε, κινητή υπολογιστική (mobile computing) ή αλληλεπιδραστικές εφαρμογές.

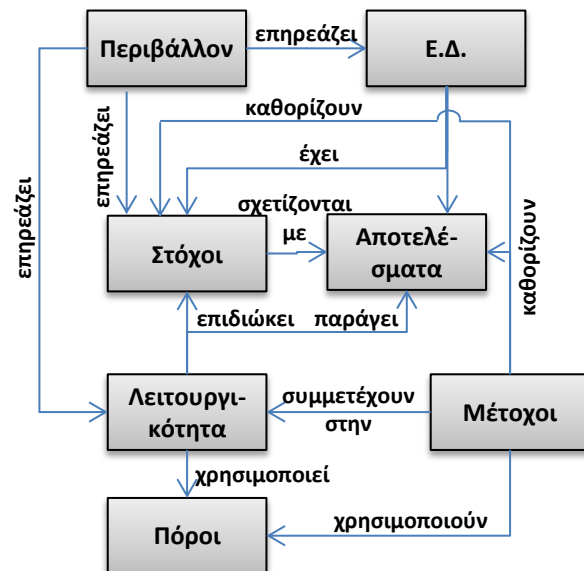
Εταιρείες παραγωγής υλικού (hardware manufacturers) όπως οι ARM, Atmel και Texas Instruments²⁸ προσφέρουν προϊόντα (όπως κάρτες WiFi, μικροελεγκτές για home automation, αυτοκίνηση, υγεία και άλλα) που παράγουν και δημοσιεύουν γεγονότα. Επίσης υπάρχουν αρκετές πλατφόρμες και λογισμικό που παρέχουν υπηρεσίες διακίνησης των γεγονότων, όπως για παράδειγμα τα Xively, Etherios Device Cloud, Open.Sen.se, Beebotte, Fluxtream, dtweet.io και ThingSpeak²⁹. Προϊόντα σχετικά με την κινητή τηλεφωνία και την τεχνολογία RFID μπορούν επίσης να παράγουν αλλά και να αξιοποιούν γεγονότα.

4.1.1 Η δική μας οπτική για το πρόβλημα

Προκειμένου να συστηματοποιήσουμε τη διαμόρφωση του προβλήματος της διατριβής, παρουσιάζουμε ακολούθως τη δική μας οπτική για τις επιχειρησιακές διαδικασίες, τον τρόπο λειτουργίας τους και τη σχέση τους με μια σειρά από παράγοντες που συμμετέχουν σε αυτές και τις επηρεάζουν. Αυτοί δίνονται διαγραμματικά στα ακόλουθα σχήματα και εν συνεχεία παρέχονται ορισμένες σύντομες διευκρινίσεις.



Σχήμα 4-1. Παράγοντες που επηρεάζουν τις επιχειρησιακές διαδικασίες



Σχήμα 4-2. Σχέσεις των παραγόντων που επηρεάζουν τις επιχειρησιακές διαδικασίες

²⁸ ARM [<http://arm.com/>], Atmel [<http://www.atmel.com/>], Texas Instruments [<http://www.ti.com/>]

²⁹ **Xively:** (πρώην Pachube και Cosm) πλατφόρμα νέφους για το Internet of Things (IoT) που παρέχεται ως υπηρεσία (Platform as a Service) [www.Xively.com]

Etherios Device Cloud: μια εναλλακτική του Xively [<http://www.etherios.com/products/devicecloud/>]

Open.Sen.se: ανοικτή και χωρίς χρέωση πλατφόρμα διασύνδεσης συσκευών, εφαρμογών, ανθρώπων [<http://open.sen.se/>]

Beebotte: πλατφόρμα νέφους για το IoT [<https://beebotte.com/>]

Fluxtream: ανοικτού κώδικα, μη κερδοσκοπικού χαρακτήρα πλαίσιο εργασίας για την οπτικοποίηση προσωπικών δεδομένων [<https://fluxtream.org/>]

dtweet.io: πολύ απλή ανταλλαγή δεδομένων για το IoT [<https://dweet.io/>]

ThingSpeak: ανοικτού κώδικα πλατφόρμα για το IoT [<https://thingspeak.com/>]

Οι **στόχοι** των επιχειρησιακών διαδικασιών καθορίζουν ποια είναι η αποστολή τους και τι εν τέλει θα πρέπει να επιτύχουν. Δεν νοείται επιχειρησιακή διαδικασία χωρίς κάποιον στόχο (έστω και απλό). Θεωρούμε ότι οι στόχοι αποτελούν την «καρδιά» ή την «κεντρική ιδέα» των διαδικασιών ενώ οι όποιες προσαρμογές και παρεμβάσεις σε αυτές θα πρέπει να διαφυλάσσουν τους στόχους αναλλοίωτους. Εξάιρεση αποτελεί η περίπτωση αλλαγής των απαιτήσεων των μετόχων (stakeholders) όπου τότε μπορεί να προσαρμόζονται και οι στόχοι. Ο βαθμός επίτευξης των στόχων μπορεί να μετρηθεί και να αποτελέσει ένα δείκτη του αν η διαδικασία χρίζει προσαρμογής ή όχι.

Οι επιχειρησιακές διαδικασίες παράγουν κάποιας μορφής **αποτελέσματα**, τα οποία φυσικά σχετίζονται ή είναι τα ζητούμενα των στόχων. Τα αποτελέσματα μπορεί να είναι η παραγωγή προϊόντων ή η παροχή υπηρεσιών. Ακόμη μπορεί να είναι υλικά ή άυλα (π.χ. ηλεκτρονικά αρχεία).

Για την επιδίωξη των στόχων τους και την παραγωγή των επιθυμητών αποτελεσμάτων, οι επιχειρησιακές διαδικασίες εμπεριέχουν μια συγκεκριμένη **λειτουργικότητα**, ως σύνθεση (ή ροή) των επιμέρους εργασιών (tasks) που τις απαρτίζουν. Οι εργασίες αυτές αξιοποιούν τα διαθέσιμα μέσα και πόρους και αλληλεπιδρούν με το περιβάλλον (εσωτερικό και εξωτερικό) και τους μετόχους προκειμένου να υλοποιήσουν τους στόχους. Καθορίζουν έτσι τη «συμπεριφορά» των διαδικασιών σε σχέση με το περιβάλλον τους.

Θεωρούμε ότι για διαφορετικές συνθήκες και διαμορφώσεις του περιβάλλοντος καθώς και των διαθέσιμων μέσων-πόρων, ενδέχεται να απαιτούνται διαφορετικές λειτουργίες για την επίτευξη των **ιδίων** πάντα στόχων. Αυτό συμβαίνει διότι οι εργασίες και άρα η συνολική λειτουργικότητα των διαδικασιών ενδέχεται να έχουν όρια και περιορισμούς εντός των οποίων μπορούν να λειτουργούν σωστά, με επάρκεια και με έναν αποδεκτό βαθμό απόδοσης, κατάλληλο για τους σκοπούς τους. Εκτός αυτών των ορίων όμως, προκαλούνται δυσλειτουργίες στις διαδικασίες που μπορεί να οδηγήσουν σε σημαντική απόκλιση από τους στόχους τους.

Τα **μέσα και οι πόροι** είναι όλες εκείνες οι δυνατότητες που παρέχονται στις διαδικασίες προκειμένου να επιδιώξουν και επιτύχουν τους στόχους τους. Περιλαμβάνουν τους ανθρώπινους πόρους (για διαδικασίες που απαιτούν τη συμμετοχή ανθρώπων), υποδομές πληροφορικής, συλλογές ή βάσεις δεδομένων και γνώσεων. Επίσης, αν το αποτέλεσμα είναι υλικό προϊόν, μπορεί να περιλαμβάνουν μηχανές παραγωγής, πρώτες ύλες και άλλα. Η έλλειψη των κατάλληλων πόρων και μέσων τη στιγμή που αυτά είναι αναγκαία μπορεί να οδηγήσει σε αστοχία της διαδικασίας οπότε απαιτείται προσαρμογή της λειτουργικότητάς της στις συνθήκες έλλειψης των πόρων.

Το **περιβάλλον** των διαδικασιών είναι το σύνολο των τρίτων οντοτήτων καθώς και των μεταξύ τους σχέσεων, οι οποίες αλληλεπιδρούν (άμεσα ή έμμεσα) ή άλλως σχετίζονται ή αφορούν ή επηρεάζουν τη λειτουργία των διαδικασιών. Με την έννοια αυτή μπορούμε να πούμε ότι τα μέσα και οι πόροι είναι μέρος του περιβάλλοντος, άποψη που θα ακολουθήσουμε εφεξής.

Τέλος οι **μέτοχοι** (stakeholders) είναι όλα εκείνα τα πρόσωπα και ρόλοι που με κάποιο τρόπο εμπλέκονται σε μια διαδικασία. Αυτοί μπορεί να είναι (α) οι διεκπεραιωτές και οι συμμετέχοντες στην εκτέλεση συγκεκριμένων εργασιών (tasks) ή στο συντονισμό της ροής (workflow) των εργασιών της διαδικασίας (είτε με διοικητικό, αποφασιστικό ρόλο, είτε ως απλοί χρήστες με διεκπεραιωτικό ρόλο), (β) οι αποδέκτες του «προϊόντος» των διαδικασιών (consumers), (γ) ο «ιδιοκτήτης» της διαδικασίας (οργανισμός, διοίκηση), (δ) ο διοικητικός υπεύθυνος της διαδικασίας, (ε) ο επιχειρησιακός υπεύθυνος της διαδικασίας. (στ) Επίσης, αν πρόκειται για αυτοματοποιημένες διαδικασίες, μέτοχοι είναι και οι σχεδιαστές, κατασκευαστές και διαχειριστές της πληροφοριακής υποδομής που υποστηρίζει το σχεδιασμό, την εκτέλεση και την παρακολούθηση των διαδικασιών. Ιδιαίτερη βαρύτητα στον καθορισμό των στόχων των διαδικασιών έχουν οι ανάγκες του «ιδιοκτήτη» και των «αποδεκτών» των αποτελεσμάτων των διαδικασιών. Ωστόσο όλοι οι μέτοχοι συμβάλλουν στο βαθμό που τους αφορά στην διαμόρφωση των τελικών στόχων και λειτουργιών.

Καθένας απ' τους παραπάνω παράγοντες είναι δυνητικά πηγή / αιτία «αποκλίσεων» και «αποτυχίας» των διαδικασιών, υπό την έννοια ότι αν αλλάξουν οι συνθήκες για κάποιον από αυτούς οι διαδικασίες ενδέχεται να μην ανταποκρίνονται πλέον στους στόχους τους. Εν συνεχεία συνοψίζουμε τον τρόπο λειτουργίας των επιχειρησιακών διαδικασιών και σκιαγραφούμε το πότε και πως δημιουργείται η ανάγκη για προσαρμογή.

1. Οι διαδικασίες σχεδιάζονται και εκτελούνται προκειμένου να επιδιώκουν και επιτυγχάνουν έναν προκαθορισμένο στόχο ή σύνολο στόχων και να παράγουν κάποιο αντίστοιχο αποτέλεσμα (προϊόν ή υπηρεσία).
2. Για την επίτευξη των στόχων τους χρησιμοποιούν συγκεκριμένες λειτουργικότητες ενώ είναι πιθανό ένας στόχος να μπορεί να επιτευχθεί με περισσότερες από μία διαφορετικές λειτουργικότητες. Τότε οι καταλληλότερες από αυτές επιλέγονται και χρησιμοποιούνται, ανάλογα με τις συνθήκες που επικρατούν.
3. Όταν αλλάζουν ορισμένες (κρίσιμες) συνθήκες του περιβάλλοντος ή και εσωτερικοί παράγοντες, οι επιχειρησιακές διαδικασίες δεν μπορούν πλέον να επιτυγχάνουν επαρκώς τους στόχους τους και να παράγουν τα αναμενόμενα αποτελέσματα. Κατά συνέπεια αποκλίνουν από αυτούς.
4. Τότε απαιτείται κατάλληλη προσαρμογή αυτών. Οι προσαρμογές «μεταβάλλουν» τη συμπεριφορά – λειτουργικότητα των επιχειρησιακών διαδικασιών αλλά όχι τους στόχους τους και τα αποτελέσματά τους. Ειδική περίπτωση είναι η αλλαγή των αναγκών των μετόχων των διαδικασιών, όπου τότε πιθανώς να πρέπει να τροποποιηθούν αντίστοιχα οι στόχοι ή και τα αποτελέσματα.

Οι προσαρμογές συνήθως αποφασίζονται, σχεδιάζονται και εκτελούνται από ανθρώπους με κατάλληλες αλλαγές, τόσο διοικητικής φύσεως όσο και τεχνικής (αν πρόκειται για αυτοματοποιημένες διαδικασίες). Τέτοιες όμως παρεμβάσεις ενέχουν σημαντικό κόστος, ιδίως όταν το περιβάλλον είναι ευμετάβλητο, ενώ ελλοχεύει ο κίνδυνος της μη συστηματικής (ad-hoc) αντιμετώπισης των αναγκών αλλαγής που εν τέλει οδηγεί σε αυξημένη πολυπλοκότητα, δυσκολία συντήρησης και φυσικά αυξημένο λειτουργικό και διοικητικό κόστος. Η χρήση τυπικών (formal) μεθόδων για τον σχεδιασμό, αποτύπωση και

εφαρμογή των προσαρμογών θα μπορούσε να βελτιώσει και τυποποιήσει τον σχεδιασμό και εφαρμογή των προσαρμογών στις διαδικασίες – στόχους. Παράλληλα θα μπορούσε να μετριάσει ή και αποσοβήσει τα προαναφερθέντα προβλήματα.

Για τον έγκαιρο εντοπισμό τυχόν αποκλίσεων καθώς και της ανάγκης για προσαρμογή, απαιτείται διαρκής πληροφόρηση κατά την διάρκεια εκτέλεσης των διαδικασιών. Όταν μάλιστα υπάρχουν ενδείξεις απόκλισης ή όταν οι συνθήκες του περιβάλλοντος δημιουργούν τέτοιες υπόνοιες, τότε είναι πιθανό να χρειαστεί η αλλαγή ή η προσθήκη επιπλέον πηγών πληροφόρησης, ακόμη και κατά τη διάρκεια εκτέλεσης των επιχειρησιακών διαδικασιών. Επιπλέον, ο σχεδιασμός και η υλοποίηση των προσαρμογών απαιτεί επαρκή, έγκυρη και επίκαιρη ενημέρωση, τόσο για την κατάσταση των διαδικασιών όσο και του περιβάλλοντός τους. Η αξιοποίηση των γεγονότων και των σχετικών τεχνολογιών μπορεί να συμβάλει προς αυτήν την κατεύθυνση.

Ήδη από την παραπάνω συνοπτική παρουσίαση της οπτικής μας ανακύπτουν αρκετά ενδιαφέροντα ερωτήματα, των οποίων η απάντηση κάθε άλλο παρά τετριμμένη μπορεί να θεωρηθεί. Φέρ' ειπείν το πως αντιλαμβανόμαστε ότι μια διαδικασία χρίζει προσαρμογής, δηλαδή πως ανιχνεύουμε ότι έχει δημιουργηθεί μια τέτοια ανάγκη, τι είδους προσαρμογή απαιτείται, πως επιλέγουμε μία, πως εφαρμόζεται μια προσαρμογή σε μια διαδικασία.

Στη συνέχεια παρουσιάζουμε τα ερωτήματα εκείνα στα οποία θα εστιάσουμε, προκειμένου να ορίσουμε το πλαίσιο και να διαμορφώσουμε ακριβώς το πρόβλημα που πραγματεύεται και απαντά η παρούσα διατριβή.

4.2 Ερευνητικά Ερωτήματα

Τα ερωτήματα στα οποία επιλέγουμε να εμβαθύνουμε καλύπτουν το μεγαλύτερο μέρος του «κύκλου ζωής» των προσαρμογών, συγκεκριμένα από την ανίχνευση της ανάγκης για προσαρμογή έως και την εφαρμογή τους. Δεν εξετάζονται όμως οι επιπτώσεις των προσαρμογών στις διαδικασίες καθώς και το αν πέτυχαν να διορθώσουν την απόκλιση ή το πρόβλημα. Θα μπορούσαμε όμως να θεωρήσουμε ότι τυχόν αποτυχία μιας προσαρμογής είναι οιονεί μια νέα ανάγκη προσαρμογής. Στη συνέχεια περιγράφεται σύντομα ο κύκλος ζωής των προσαρμογών και ακολούθως διατυπώνονται και αναλύονται τα ερευνητικά ερωτήματα που θέσαμε.

4.2.1 Κύκλος Ζωής Προσαρμογών (Adaptation Life-Cycle)

Ο Κύκλος Ζωής των Προσαρμογών είναι ουσιαστικά μια εξειδίκευση (specialization) και συμπλήρωση του γνωστού κύκλου Παρατήρησης – Κατεύθυνσης – Απόφασης – Δράσης (Observe – Orient – Decide – Act) ή OODA loop (Greene, 2007; Osinga, 2007; Richards, 2004), στο πεδίο των προσαρμοστικών εφαρμογών. Συγκεκριμένα, ο κύκλος αυτός περιγράφει τις φάσεις από τη δημιουργία του αιτίου απόκλισης μιας διαδικασίας από τους στόχους της ή τα όρια λειτουργίας της, την καταγραφή της από κατάλληλα μέσα παρακολούθησης – ανιχνευτές (ως γεγονότα), την ανίχνευση – διαπίστωση της ανάγκης προσαρμογής της διαδικασίας, την επιλογή μιας κατάλληλης προσαρμογής, ενδεχομένως την πρότασή της στον υπεύθυνο της διαδικασίας για έλεγχο και έγκριση, και τέλος την

εφαρμογή της προσαρμογής, δηλαδή την τροποποίηση της διαδικασίας. Επίσης συμπεριλαμβάνει και τη «διόρθωση» του προβλήματος ως αποτέλεσμα της αλλαγής συμπεριφοράς της διαδικασίας αλλά και την επανάληψη του κύκλου αν η προσαρμογή δεν πέτυχε να θεραπεύσει το πρόβλημα. Σημειώνουμε ότι ο κύκλος ισχύει και για την περίπτωση των βελτιωτικών προσαρμογών αλλά σε αυτήν την περίπτωση ζητούμενο είναι η ανίχνευση τυχόν ευκαιριών βελτίωσης ενώ το αποτέλεσμα της προσαρμογής είναι η αξιοποίηση της ευκαιρίας.

Η παρούσα διατριβή εστιάζει στις εξής φάσεις του κύκλου ζωής των προσαρμογών:

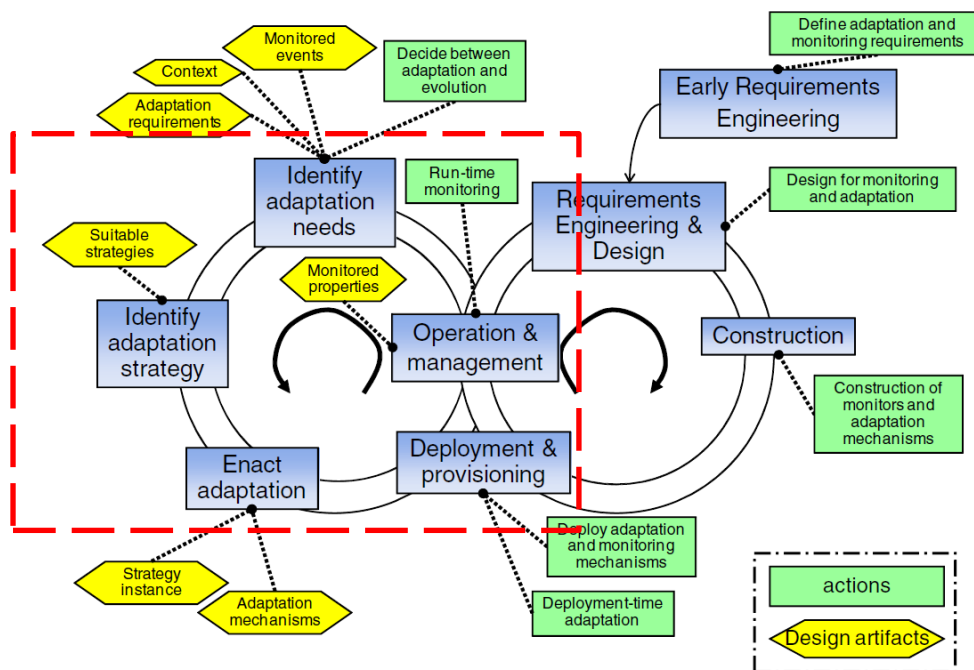
- ανίχνευσης της ανάγκης προσαρμογής,
- επιλογής της προσαρμογής,
- πρότασης της προσαρμογής, και
- εφαρμογής της προσαρμογής.

Στις υπόλοιπες φάσεις του κύκλου ζωής των προσαρμογών, οι οποίες αφορούν κυρίως τα τεκτενόμενα στο περιβάλλον, δεν υπεισέρχεται. Στο διάγραμμα που ακολουθεί οι φάσεις που αφορούν την διατριβή σημειώνονται με μπλε χρώμα.

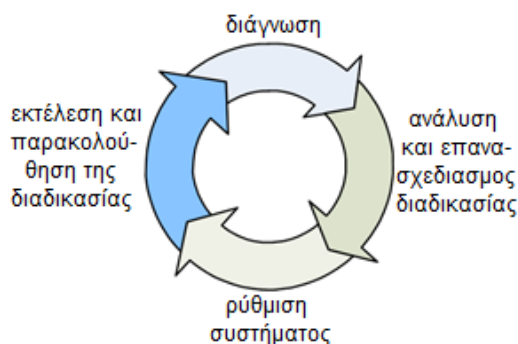


Σχήμα 4-3. Κύκλος ζωής των προσαρμογών

Ο Κύκλος Ζωής Προσαρμογών προσομοιάζει αρκετά με τον «Κύκλο Ζωής Προσαρμοστικών Εφαρμογών Βασισμένων σε Υπηρεσίες» (Life Cycle of Adaptable SBAs) που προτείνουν οι (Bucchiarone et al., 2010) (αριστερός κύκλος στο Σχήμα 4-4), κατάλληλα προσαρμοσμένου στο πεδίο των επιχειρησιακών διαδικασιών. Μια άλλη εργασία που κινείται σε παρόμοια κατεύθυνση είναι εκείνη των (ter Hofstede et al., 2010) όπου παρουσιάζεται ο «Κύκλος Ζωής της Διαχείρισης Επιχειρησιακών Διαδικασιών» (Life-Cycle of BPM) (Σχήμα 4-5).



Σχήμα 4-4. Κύκλος ζωής των προσαρμοστικών εφαρμογών (Bucchiarone et al., 2010)



Σχήμα 4-5. Κύκλος ζωής Διαχείρισης Επιχειρησιακών Διαδικασιών (ter Hofstede et al., 2010)

Όλες οι παραπάνω προσεγγίσεις περιλαμβάνουν αρχικά ένα στάδιο παρακολούθησης της διαδικασίας ή εφαρμογής, ένα στάδιο ανάλυσης και διαπίστωσης της ανάγκης για αλλαγή / προσαρμογή, την επιλογή της κατάλληλης προσέγγισης για αλλαγή και την εφαρμογή της αλλαγής ή προσαρμογής που επιλέχθηκε.

4.2.2 Ανίχνευση της ανάγκης προσαρμογής με χρήση γεγονότων

Ίσως το πιο δύσκολο σημείο σε ολόκληρο τον κύκλο ζωής μιας προσαρμογής (στο τμήμα που μας αφορά) είναι η αρχή του. Με άλλα λόγια είναι ο έγκαιρος εντοπισμός και η αναγνώριση της ανάγκης για προσαρμογή, όταν χρειάζεται. Προκειμένου να διαπιστωθεί αυτή η ανάγκη θα πρέπει να συνεκτιμηθούν οι μεταβολές των διαφόρων παραγόντων που επηρεάζουν την εκάστοτε διαδικασία. Στο πλαίσιο αυτής της διατριβής μια επιπλέον πρόκληση είναι το ότι οι μεταβολές των παραγόντων καταγράφονται υπό τη μορφή γεγονότων (σύμφωνα με την υπόθεσή μας). Τα γεγονότα συνήθως μεταφέρουν πρωτογενή και σχετικά απλή πληροφορία (π.χ. μετρήσεις) που δεν παραπέμπει ευθέως στην ανάγκη για προσαρμογή. Θα πρέπει λοιπόν να συνδυαστούν και συσχετιστούν κατάλληλα

προκειμένου να διαφανεί η εν λόγω ανάγκη. Επιπρόσθετα θα πρέπει να εξετάζεται το κατά πόσο οι μεταβολές των παραγόντων δημιουργούν πραγματική ανάγκη για προσαρμογή καθώς κάποιες από αυτές θα αφορούν αδιάφορες διακυμάνσεις των συνθηκών. Άλλες πάλι μπορούν να «απορροφηθούν» από την υφιστάμενη διαδικασία ως έχει.

Από τα παραπάνω υπογραμμίζουμε τα εξής δύο σημεία. Πρώτον απαιτείται συνεχής έλεγχος των διαθέσιμων πληροφοριών (μέσω ανάλυσης των αντίστοιχων γεγονότων) για τον εντοπισμό τυχόν σημαντικών μεταβολών στους παράγοντες που επηρεάζουν τις επιχειρησιακές διαδικασίες (είτε μεμονωμένα είτε συνδυαστικά). Αυτό είναι σημαντικό διότι μόνο με αυτόν τον τρόπο μπορούν να εντοπίζονται έγκαιρα οι ανάγκες προσαρμογής και να προτείνονται κατόπιν κατάλληλες προσαρμογές τη στιγμή που αυτές χρειάζονται. Δεύτερον, προκειμένου να διαπιστωθεί αν οι όποιες εξωτερικές ή εσωτερικές μεταβολές πρέπει να οδηγήσουν σε προσαρμογή απαιτείται γνώση της δομής, του τρόπου λειτουργίας και των ορίων – δυνατοτήτων των ίδιων των διαδικασιών.

Όσον αφορά το πρώτο σημείο, μια απάντηση παρέχεται από το ερευνητικό πεδίο της «Επίγνωσης και Διαχείρισης Καταστάσεων» (Situation Awareness and Management). Στόχος της επίγνωσης καταστάσεων είναι ο εντοπισμός του σε ποια κατάσταση βρίσκεται μια διαδικασία ή σύστημα και τότε αυτή αλλάζει και μεταβαίνει σε μια νέα (situation change). Η κατάσταση μιας διαδικασίας (ή του περιβάλλοντός της) είναι το σύνολο ή ο συνδυασμός των συνθηκών και παραγόντων που την επηρεάζουν (μεμονωμένα ή συνδυαστικά) και ισχύουν σε ορισμένο χρονικό πλαίσιο³⁰. Οι παράγοντες και οι συνθήκες αυτές πρέπει να έχουν ένα κοινό νόημα, πιθανώς ίδια γενεσιουργό αιτία, και από κοινού να επηρεάζουν την εξέλιξη της εκτέλεσης της διαδικασίας.

Επίσης στον κύκλο Διαχείρισης Κατάστασης των (Jakobson et al., 2007) φαίνεται ξεκάθαρα ότι η επίγνωση κατάστασης προέρχεται από την ανίχνευση (sensing) των κατάλληλων πληροφοριών (εδώ γεγονότων). Συνεπώς ένα ακόμη ζήτημα είναι αυτό της επιλογής των εκάστοτε κατάλληλων πηγών γεγονότων και άρα πληροφόρησης. Η επίγνωση κατάστασης πρέπει να οδηγεί στην κατανόηση των αιτιακών συνθηκών της αλλαγής κατάστασης, των παραμέτρων τους καθώς και των συνεπειών τους. Ο λόγος είναι ότι η βαθύτερη κατανόηση των αιτίων μπορεί να συμβάλει στη δημιουργία ή στην επιλογή της πιο κατάλληλης προσαρμογής που θα θεραπεύσει τις όποιες αποκλίσεις.

Όσον τώρα αφορά το δεύτερο σημείο, δηλαδή τη γνώση της ίδιας της διαδικασίας (δομή, λειτουργία, όρια – δυνατότητες) για την διαπίστωση της ανάγκης προσαρμογής. Είναι σημαντικό να εκτιμηθεί το κατά πόσο συνεχίζουν να επιτυγχάνονται οι στόχοι των διαδικασιών, να μετρηθεί ο βαθμός επίτευξής τους καθώς και η μεταβολή του σε σχέση με την αλλαγή των παραμέτρων των παραγόντων που επηρεάζουν τις διαδικασίες. Εδώ φυσικά υπάρχει το ζήτημα του ποιος βαθμός επίτευξης θεωρείται επαρκής (ή μη) προκειμένου να οδηγήσει στη διαπίστωση μιας ανάγκης προσαρμογής. Επιπλέον, σε συνδυασμό με τους στόχους, θα πρέπει να εξεταστούν και οι επιμέρους λειτουργίες που

³⁰ “the set of conditions that exist at a particular time in a particular place” MacMillan British Dictionary [<http://www.macmillandictionary.com/dictionary/british/situation>]

επιτελούνται σε μια διαδικασία, προκειμένου να μετρηθεί το πόσο επηρεάζονται από τις μεταβολές των παραγόντων της κατάστασης και αν συνεχίζουν να είναι εντός των αποδεκτών ορίων οι παράμετροι που σχετίζονται με αυτές. Σημειώνουμε σε αυτό το σημείο ότι θεωρούμε ότι η χρήση εμπειρογνομόνων (experts) είναι η μόνη πλήρης, ορθή και ασφαλής λύση. Ομοίως και η επιλογή των πηγών πληροφόρησης (εδώ γεγονότων), ο τρόπος αλλαγής αυτών ή η προσθήκη νέων, αναλόγως με την εκάστοτε κατάσταση που επικρατεί, πρέπει να επαφίενται επίσης στην κρίση εμπειρογνομόνων.

Τέλος συμπληρώνουμε ότι εκτός από τις προβληματικές καταστάσεις που χρίζουν προσαρμογής, μπορεί να υπάρξουν και καταστάσεις όπου όλα λειτουργούν κανονικά, ωστόσο εμφανίζονται «ευκαιρίες» για βελτίωση της λειτουργίας των διαδικασιών. Αν κάτι τέτοιο είναι επιθυμητό, τότε ζητούμενο θα ήταν η επιδίωξη καταστάσεων όπου η επίτευξη των στόχων των διαδικασιών θα γίνεται με πιο οικονομικό τρόπο ή με καλύτερη χρήση των πόρων και μέσων, στις οποίες (καταστάσεις) θα οδηγείτο η διαδικασία μέσα από βελτιωτικές προσαρμογές. Επίσης αν μπορεί να μετρηθεί ο βαθμός επίτευξης των στόχων ή η ποιότητα των αποτελεσμάτων, θα μπορούσαν τότε να αναζητηθούν προσαρμογές των λειτουργιών των διαδικασιών που οδηγούν σε καλύτερους δείκτες εκτέλεσης.

4.2.3 Προσαρμογή διαδικασιών οδηγούμενη από την επίγνωση κατάστασης

Μετά την αντίληψη μιας «προβληματικής» κατάστασης ή μιας «ευκαιρίας» βελτίωσης, το ερευνητικό ερώτημα που λογικά έπεται είναι το πως μπορούμε να οδηγηθούμε στην προσαρμογή των διαδικασιών, έτσι ώστε να θεραπευτεί το πρόβλημα ή να αξιοποιηθεί η ευκαιρία. Το ερώτημα αφορά τόσο τη δημιουργία ή επιλογή μιας προσαρμογής όσο και την εφαρμογή της στην προβληματική διαδικασία.

Κατ' αρχήν ένα ζήτημα είναι η επιλογή της στρατηγικής με την οποία παράγονται οι προσαρμογές. Τις στρατηγικές μπορούμε να τις διακρίνουμε σε δύο βασικές κατηγορίες: (α) αυτές που συνθέτουν (δημιουργούν) προσαρμογές απ' την αρχή και (β) αυτές που επιλέγουν προσαρμογές μέσα από ένα σύνολο εναλλακτικών, τις οποίες ενδεχομένως να εξειδικεύουν. Στην παρούσα διατριβή εστιάζουμε στην δεύτερη περίπτωση, δηλαδή στην επιλογή από ένα σύνολο εκ των προτέρων (a priori) σχεδιασμένων και διαθέσιμων εναλλακτικών προσαρμογών. Αυτό μπορεί να γίνει με διάφορες μεθόδους (όπως για παράδειγμα πολυκριτηριακές μέθοδοι) που λαμβάνουν υπόψη πληροφορίες για την κατάσταση καθώς και τα χαρακτηριστικά της κάθε εναλλακτικής προσαρμογής. Επίσης οι εναλλακτικές δύνανται να είναι πρότυπα (templates) προσαρμογών ή γενικές (generic) προσαρμογές που «εξειδικεύονται» (instantiated ή specialized) στην εκάστοτε περίπτωση προτού εφαρμοστούν. Επιπλέον μπορεί οι καλύτερες από αυτές και να προτείνονται στον υπεύθυνο της διαδικασίας προκειμένου αυτός να επιλέξει εκείνη που τελικά θα χρησιμοποιηθεί (recommendation approach). Αυτήν ακριβώς την προσέγγιση υιοθετούμε στην παρούσα διατριβή.

Ένα άλλο ζήτημα είναι το πως εκφράζονται και καταγράφονται οι προσαρμογές. Γενικά η μοντελοποίηση των προσαρμογών είναι βασική προϋπόθεση για την αποτελεσματική επικοινωνία μεταξύ του αποφασίζοντος και επιλέγοντας τις προσαρμογές με εκείνον που τις εφαρμόζει. Αυτό είναι ιδιαιτέρως κρίσιμο όταν ο κύκλος ζωής της προσαρμογής

απαιτείται να αυτοματοποιηθεί. Κατά τη μοντελοποίηση πρέπει να καθορίζεται και το εύρος εφαρμογής των προσαρμογών, ήτοι σε ποιες εκτελέσεις / στιγμιότυπα της διαδικασίας πρέπει να εφαρμόζονται καθώς και για πόσο διάστημα.

Ένα ακόμη σημείο που αξίζει περαιτέρω προσοχής είναι το αν μια μέθοδος προσαρμογής απαιτεί την προηγούμενη επεξεργασία και προετοιμασία των διαδικασιών προκειμένου αυτές να μπορούν να δεχτούν προσαρμογές. Τέτοιες προσεγγίσεις είναι πρακτικά δυσκολότερο να εφαρμοστούν σε ήδη υφιστάμενες διαδικασίες ενώ θέτουν περιορισμούς στα σημεία (των διαδικασιών) που μπορούν να δεχτούν προσαρμογή. Από την άλλη πλευρά όμως επιτρέπουν πιο στοχευμένες και εξειδικευμένες παρεμβάσεις.

Η εφαρμογή προσαρμογών χωρίς προηγούμενη γνώση ή προετοιμασία των διαδικασιών αντιμετωπίζει αρκετές προκλήσεις, όπως η έλλειψη γνώσης εξειδικευμένης της εκάστοτε διαδικασίας. Ωστόσο οι σχετικές μέθοδοι δεν απαιτούν αλλαγή στις διαδικασίες παρά μόνο κατάλληλο λογισμικό εκτέλεσης. Στη βιβλιογραφία έχουν προταθεί διάφορες τέτοιες μέθοδοι όπως είναι η θεματοστρεφής προσαρμογή (π.χ. AO4BPEL).

Συνοψίζοντας, δίνεται ένας πίνακας με τα βασικά ερευνητικά ερωτήματα της διατριβής καθώς και των επιμέρους παραμέτρων τους. Στη συνέχεια της διατριβής θα αναφέρουμε πως εμείς προσεγγίζουμε τα θέματα που συζητήθηκαν παραπάνω.

Πίνακας 4-1. Ερευνητικά ερωτήματα και οι παράμετροί τους

Ερευνητικό Ερώτημα	Επιμέρους Παράμετροι
Ανίχνευση της ανάγκης προσαρμογής με χρήση γεγονότων (Ανίχνευση & Απόφαση)	<ul style="list-style-type: none"> • Πώς αξιοποιούνται τα γεγονότα; • Πώς επιτυγχάνεται η επίγνωση κατάστασης; • Πώς εκφράζεται / μοντελοποιείται η επίγνωση της κατάστασης και της ανάγκης προσαρμογής; • Πότε υπάρχει ανάγκη προσαρμογής;
Προσαρμογή επιχειρησιακών διαδικασιών οδηγούμενη από την επίγνωση κατάστασης (Επιλογή & Πρόταση)	<ul style="list-style-type: none"> • Πώς γίνεται η επιλογή προσαρμογών με βάση μια δεδομένη ανάγκη ή κατάσταση; • Πώς μοντελοποιείται η διαδικασία αυτή;
Εφαρμογή προσαρμογών	<ul style="list-style-type: none"> • Πώς εφαρμόζονται οι προσαρμογές στις διαδικασίες; • Απαιτείται a priori προετοιμασία των διαδικασιών; • Πώς καταγράφονται οι προσαρμογές;

4.2.4 Συμβάσεις και Επιλογές της διατριβής

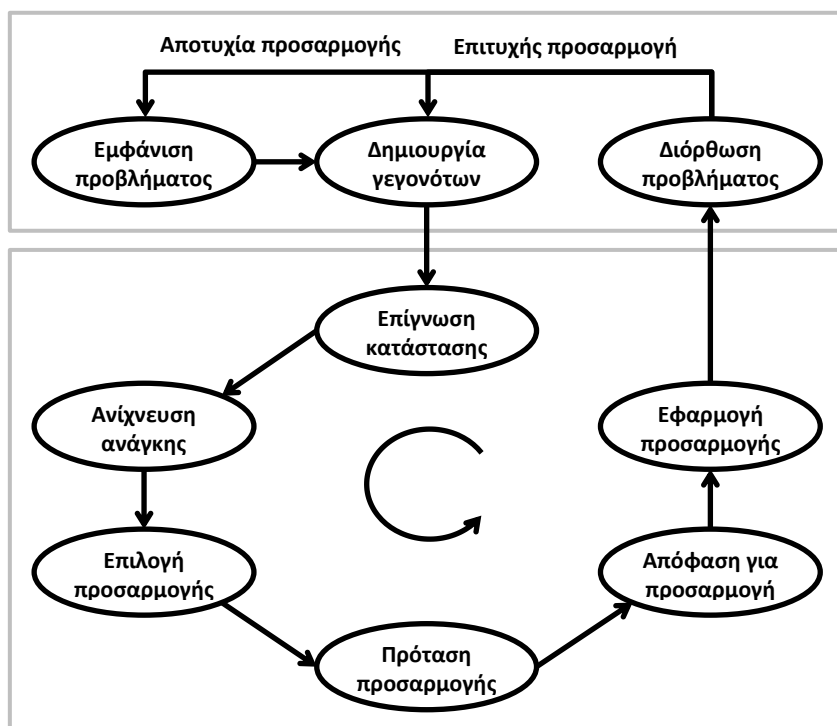
Στην παρούσα διατριβή επιλέγουμε να προσεγγίσουμε τη λύση του προβλήματος χρησιμοποιώντας τις ακόλουθες παραδοχές και επιλογές, όσον αφορά τα ερευνητικά ερωτήματα που τέθηκαν παραπάνω. Ειδικότερα, θα εξετάσουμε:

- Διαδικασίες, εφαρμογές και συστήματα για των οποίων την κατάσταση υπάρχουν γεγονότα (τουλάχιστο για τις ανάγκες προσαρμογής τους).

- Περιπτώσεις διορθωτικών προσαρμογών κι όχι βελτιωτικών.
- Εστιάζουμε στην επιλογή προσαρμογών από ένα σύνολο (προδιαγεγραμμένων) εναλλακτικών κι όχι στη σύνθεση νέων.
- Εστιάζουμε σε προσαρμογές που δεν απαιτούν την εκ των προτέρων (a priori) προετοιμασία των διαδικασιών προκειμένου να δεχτούν προσαρμογές.
- Διαδικασίες και προσαρμογές αποτυπωμένες με τη γλώσσα BPMN 2.0.

Η γλώσσα BPMN 2.0 (Business Process Modeling Notation version 2.0) είναι μια γενικής χρήσης, γραφική γλώσσα μοντελοποίησης επιχειρησιακών διαδικασιών (graphical modeling language). Οι περιγραφές διαδικασιών στη γλώσσα αυτή είναι απευθείας εκτελέσιμες από τους αντίστοιχους μηχανισμούς εκτέλεσης. Η BPMN 2.0 είναι ένα πρότυπο του διεθνούς κοινότυπου Object Management Group³¹ (OMG) και φιλοδοξεί να αποτελέσει τη lingua franca για τη μοντελοποίηση επιχειρησιακών διαδικασιών (Bonnet et al., 2014). Για τους λόγους αυτούς στην παρούσα διατριβή επιλέξαμε να εστιάσουμε μόνο σε διαδικασίες μοντελοποιημένες με την BPMN 2.0.

Με βάση την προηγούμενη ανάλυση των ερευνητικών ερωτημάτων και των επιλογών που έχουμε κάνει για την παρούσα διατριβή, ο κύκλος ζωής προσαρμογών που απεικονίζεται στο Σχήμα 4-3 μετασχηματίζεται και αναλύεται ως ακολούθως, προκειμένου να λάβει υπόψη του έννοιες όπως τα γεγονότα και η επίγνωση κατάστασης.



Σχήμα 4-6. Κύκλος ζωής των προσαρμογών στο πλαίσιο της διατριβής

³¹ <http://www.omg.org/>

4.3 Πρόταση της Διατριβής

Στη συνέχεια του κεφαλαίου παρουσιάζεται συνοπτικά η προσέγγιση που προτείνεται στην παρούσα διατριβή ως μια απάντηση στο πρόβλημα της οδηγούμενης από γεγονότα προσαρμογής επιχειρησιακών διαδικασιών που συζητήθηκε προηγουμένα. Η προσέγγιση αυτή απαντά στα ερευνητικά ερωτήματα όπως τέθηκαν παραπάνω.

4.3.1 Περίγραμμα της προτεινόμενης λύσης

Η προτεινόμενη λύση αποτελεί ένα πλαίσιο εργασίας (framework) κατάλληλο αφενός για την αποτύπωση και μοντελοποίηση του τρόπου επιλογής προσαρμογών και αφετέρου εφαρμογής τους στις επιχειρησιακές διαδικασίες. Το προτεινόμενο πλαίσιο δεν προτείνει ή εστιάζει σε συγκεκριμένες τεχνικές ανίχνευσης της ανάγκης για προσαρμογή ή επιλογής προσαρμογών, αλλά παραμένει αρκετά γενικό έτσι ώστε να είναι δυνατή η αποτύπωση και υλοποίηση με αυτό σχεδόν οποιασδήποτε μεθόδου οδηγούμενης από γεγονότα.

Η κεντρική ιδέα της προσέγγισης είναι ότι συνδυάζοντας και αξιοποιώντας γεγονότα οδηγούμαστε στην επίγνωση της κατάστασης μιας διαδικασίας – στόχου καθώς και του περιβάλλοντός της. Στη συνέχεια, με βάση την επίγνωση της κατάστασης, εκτιμάται αν υπάρχει ανάγκη προσαρμογής της διαδικασίας κι αν ισχύει κάτι τέτοιο τότε επιλέγεται μία κατάλληλη προσαρμογή από ένα σύνολο εναλλακτικών και προτείνεται στον υπεύθυνο της διαδικασίας. Κατόπιν, αν αυτή γίνει δεκτή, εφαρμόζεται στη διαδικασία και την τροποποιεί. Το περίγραμμα της προσέγγισης δίνεται στο διάγραμμα που ακολουθεί όπου φαίνονται και οι κύριες φάσεις της.



Σχήμα 4-7. Φάσεις της προτεινόμενης προσέγγισης

Από τα γεγονότα που θα λαμβάνονται μετά την εφαρμογή μιας προσαρμογής είναι δυνατό να εκτιμηθεί το αν και κατά πόσο αυτή συνέβαλλε στην επίλυση του προβλήματος. Αν δεν ήταν επιτυχής τότε με επανάληψη της προτεινόμενης προσέγγισης μπορεί να προταθεί μια νέα προσαρμογή. Αν πάλι ήταν επιτυχής και η διαδικασία επανήλθε στη φυσιολογική της κατάσταση, τότε μπορεί να εξεταστεί το ενδεχόμενο άρσης της προσαρμογής, αν φυσικά εκτιμηθεί ότι κάτι τέτοιο δεν εγκυμονεί τον κίνδυνο επανεμφάνισης του προβλήματος.

Στις επόμενες ενότητες θα περιγράψουμε για κάθε φάση της προτεινόμενης προσέγγισης το μέρος του προβλήματος στο οποίο απαντά καθώς και τη μέθοδο που χρησιμοποιεί. Επίσης θα επισημάνουμε την καινοτομία που εισάγει ενώ θα παρουσιάσουμε συνοπτικά και τα συστήματα (λογισμικό) που την υλοποιούν.

4.3.2 Λήψη και αξιοποίηση γεγονότων

Η πρώτη φάση της προσέγγισης επιγράφεται ως «Λήψη σχετικών γεγονότων» και προβλέπει τη λήψη (από διάφορες πηγές) γεγονότων που αφορούν τη διαδικασία – στόχο,

το περιβάλλον της αλλά και τις οντότητες που αλληλεπιδρούν και συνυπάρχουν με αυτή. Η φάση αυτή δεν απαντά απευθείας σε κάποιο ερευνητικό ερώτημα ωστόσο συλλέγει και παρέχει την απαραίτητη πληροφορία στις επόμενες φάσεις του προτεινόμενου πλαισίου εργασίας.

Στη φάση αυτή επιτελούνται δύο βασικές λειτουργίες. Η πρώτη αφορά την εγγραφή σε πηγές γεγονότων, σχετικές με τους σκοπούς και τις λειτουργίες της διαδικασίας – στόχου, όταν αυτό απαιτείται. Η εγγραφή στις πηγές γεγονότων οδηγείται από την επίγνωση καταστάσεων που επιτυγχάνεται στην επόμενη φάση, με βάση γεγονότα που λαμβάνονται σε τούτη. Η ανατροφοδότηση αυτή απεικονίζεται στο Σχήμα 4-7 με το βέλος από τη φάση 2 στη φάση 1 (Φ2 ► Φ1). Το υπόδειγμα Αποστολής – Εγγραφής (Pub/Sub model) για τη διανομή και προώθηση των γεγονότων θεωρείτε ως η πλέον ενδεδειγμένη λύση γι' αυτή τη φάση.

Η δεύτερη λειτουργία αφορά την επεξεργασία των πρωτογενών γεγονότων που λαμβάνονται από τις πηγές, σύμφωνα με προκαθορισμένους κανόνες και πρότυπα, και την παραγωγή «σύνθετων γεγονότων». Αυτά φέρουν υψηλότερης προστιθέμενης αξίας ή ανώτερο σημασιολογικά περιεχόμενο. Για το σκοπό αυτό χρησιμοποιούνται τεχνικές σύνθετης επεξεργασίας γεγονότων (CEP).

Τόσο οι πηγές γεγονότων όσο και οι τεχνικές, οι κανόνες ή τα πρότυπα παραγωγής των σύνθετων γεγονότων, καθορίζονται και καταγράφονται από κάποιον εμπειρογνώμονα με τον τρόπο που θα περιγραφεί στην αμέσως επόμενη φάση. Σημειώνουμε ακόμη ότι τα παραπάνω μπορούν να μεταβάλλονται κατά τη διάρκεια εκτέλεσης μιας διαδικασίας, ανάλογα με τις συνθήκες που επικρατούν και τους εκάστοτε επιδιωκόμενους στόχους, έτσι ώστε να καταστεί εφικτή στην επόμενη φάση η απόκτηση της επιθυμητής επίγνωσης κατάστασης για τη διαδικασία – στόχο.

4.3.2.1 Υποσύστημα Διαύλου Γεγονότων & Επεξεργασίας Σύνθετων Γεγονότων

Το υποσύστημα αυτό περιλαμβάνει ένα δίαυλο γεγονότων (event bus) για προώθηση και διανομή των γεγονότων, το οποίο ακολουθεί το υπόδειγμα Αποστολής–Εγγραφής (Pub/Sub event bus) καθώς και ένα μηχανισμό επεξεργασίας σύνθετων γεγονότων (CEP engine). Ο μεν δίαυλος απαιτείται προκειμένου να μεταφέρει τα γεγονότα από τις πηγές τους στα άλλα συστήματα του πλαισίου εργασίας, τον μηχανισμό CEP ή και σε τρίτες εφαρμογές. Ο δε μηχανισμός επεξεργασίας σύνθετων γεγονότων παράγει σύνθετα γεγονότα από τα πρωτογενή.

Ο δίαυλος γεγονότων θα πρέπει να υποστηρίζει θέματα (event topics) ή διακριτές ροές γεγονότων (event streams) έτσι ώστε γεγονότα διαφορετικού είδους ή διαφορετικού αντικειμένου να μην αναμιγνύονται μεταξύ τους. Ακόμη, η δυνατότητα αποθήκευσης των γεγονότων (για μελλοντική αναφορά και χρήση) καθώς και η δυνατότητα αναζήτησης αυτών μπορεί να αυξήσει σημαντικά την αποτελεσματικότητα και εκφραστικότητα του μηχανισμού CEP αφού θα μπορεί να δημιουργεί σύνθετα γεγονότα που συνδυάζουν τόσο τρέχοντα όσο και παλαιότερα (ιστορικά) γεγονότα.

Στο πλαίσιο της παρούσας διατριβής δεν εντρυφούμε περαιτέρω στο θέμα του διαύλου γεγονότων και του μηχανισμού σύνθετης επεξεργασίας γεγονότων αφού δεν εμπίπτουν στα ενδιαφέροντά της. Άλλωστε τόσο στη βιβλιογραφία όσο και εμπορικά προτείνονται αρκετές μέθοδοι και λύσεις για διαύλους γεγονότων, μεθόδους pub/sub και μηχανισμούς CEP³². Για την υλοποίηση του προτεινόμενου πλαισίου εργασίας χρησιμοποιήθηκε λογισμικό που αναπτύχθηκε στο πλαίσιο του ερευνητικού έργου PLAY³³. Συγκεκριμένα για τον δίαυλο γεγονότων χρησιμοποιήθηκε το Distributed Service Bus (DSB) που βασίζεται στο ανοικτού κώδικα σύστημα Petals ESB³⁴. Για την επεξεργασία σύνθετων γεγονότων χρησιμοποιήθηκε ο μηχανισμός Distributed Complex Event Processing (DCEP), ο οποίος είναι βασισμένος στο σύστημα ETALIS³⁵ (Anicic et al., 2012b).

4.3.3 Ανίχνευση αναγκών προσαρμογής

Η δεύτερη φάση «**Ανίχνευση των αναγκών προσαρμογής**», αφορά την αξιοποίηση γεγονότων (πρωτογενών και σύνθετων) που ληφθήκαν ή παραχθήκαν στην πρώτη φάση και που οδηγούν στην επίγνωση της κατάστασης της διαδικασίας – στόχου και των συνθηκών του περιβάλλοντός της. Απαντά στο πρώτο ερευνητικό ερώτημα, δηλαδή στο πως μπορεί να ανιχνεύεται η ανάγκη προσαρμογής μιας διαδικασίας από γεγονότα που έχουν ληφθεί ή παραχθεί στην προηγούμενη φάση.

Οι πληροφορίες που προέρχονται από τα γεγονότα συνδυάζονται με άλλες που αντλούνται από εξωτερικές πηγές ή πληροφορίες που είναι ήδη γνωστές, προκειμένου να εκτιμηθεί η τρέχουσα κατάσταση και να διαπιστωθεί η ανάγκη (ή μη) για προσαρμογή της διαδικασίας – στόχου. Επίσης εδώ υπολογίζονται και καθορίζονται οι τιμές των παραμέτρων που ποσοτικοποιούν την ανάγκη προσαρμογής και οι οποίες θα επιτρέψουν μετέπειτα την επιλογή μιας προσαρμογής. Είναι προφανές ότι οι παράμετροι αυτές διαφέρουν από διαδικασία σε διαδικασία και από εφαρμογή σε εφαρμογή.

Η χρήση ενός **εμπειρογνώμονα** στο εκάστοτε πεδίο ή εφαρμογή (domain ή application expert), είναι απαραίτητη προκειμένου να επιλέξει τον κατάλληλο τρόπο για την επίγνωση κατάστασης αλλά και την επιλογή προσαρμογών (στην τρίτη φάση).

4.3.3.1 Καινοτομία 1: Δίκτυα Κατάστασης – Ενέργειας (SANs)

Για τη μοντελοποίηση του τρόπου που οδηγούμαστε στην επίγνωση της κατάστασης μιας διαδικασίας (και ειδικότερα στην ανίχνευση της ανάγκης προσαρμογής αυτής), εισάγονται και προτείνονται τα **Δίκτυα Κατάστασης – Ενέργειας** ή Situation – Action Networks (**SANs**). Τα SANs υιοθετούν μια προσέγγιση μοντελοποίησης προσανατολισμένη σε στόχους (goal-oriented modeling), δηλαδή η αποτύπωση της επίγνωσης κατάστασης ξεκινά με τον

³² Παραδείγματα διαύλων και CEP engines: SQLstream, Microsoft StreamInsight, openPDC, Apama, SAP ESP, TIBCO BusinessEvents & Streambase, IBM WebSphere Operational Decision Management, Informatica RulePoint, Drools Fusion, GigaSpaces XAP, Oracle Event Processing και Esper.

³³ “Pushing dynamic and ubiquitous interaction between services Leveraged in the Future Internet by ApplYing complex event processing” [<http://www.play-project.eu>]

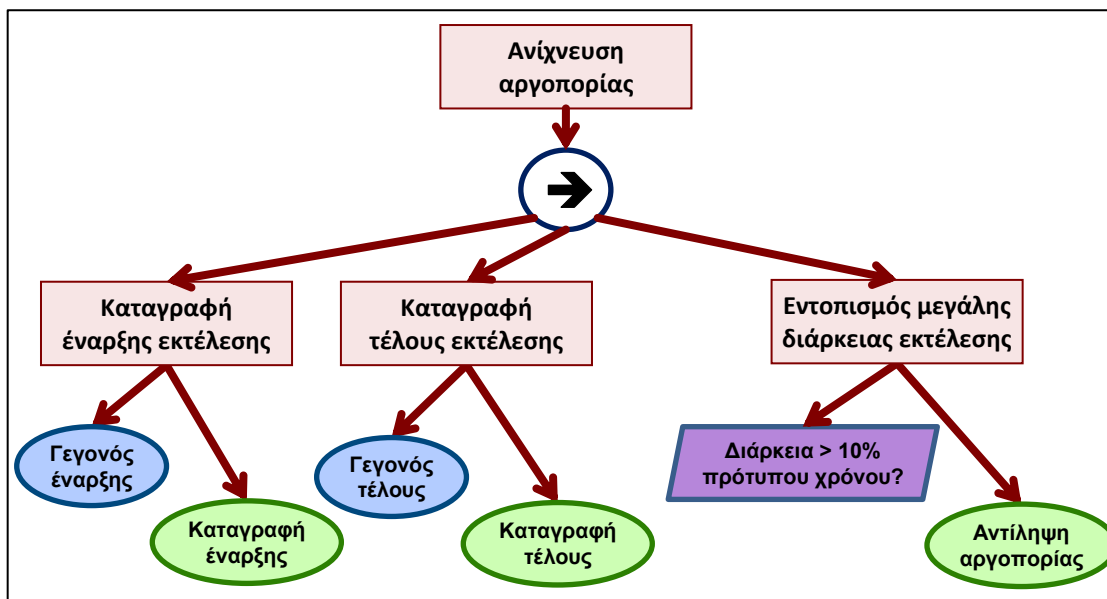
³⁴ <http://petals.ow2.org/>

³⁵ <https://code.google.com/p/etalis/>

προσδιορισμό του προς επίτευξη στόχου της διαδικασίας καθώς και των καταστάσεων που συνδέονται με αυτόν. Ακολουθώντας αυτός αναλύεται σε υποστόχους που πρέπει να επιτευχθούν προκειμένου να επιτευχθεί και ο αρχικός στόχος και ούτω καθεξής. Για κάθε στόχο (ή υποστόχο) χρησιμοποιείται ένα γεγονός (σύνθετο ή πρωτογενές) που δίνει το έναυσμα για την επιδίωξή του. Το γεγονός αυτό σηματοδοτεί την ανίχνευση μιας αλλαγής κατάστασης και παρέχει πληροφορίες σχετικές με αυτήν. Οι πληροφορίες αυτές μπορούν να συνδυαστούν με άλλες που είναι ήδη διαθέσιμες από προηγούμενα γεγονότα κι αν χρειαστεί εκτελούνται συγκεκριμένες ενέργειες ή ενεργοποιούνται και επιδιώκονται οι υποστόχοι του στόχου (δυναμική ανάλυση του στόχου). Επιπλέον μπορεί να εξετάζεται αν συντρέχουν ορισμένες συνθήκες προτού εκτελεστούν οι προβλεπόμενες ενέργειες ή δυναμική ανάλυση του στόχου.

Κάθε φορά που ξεκινά η επιδίωξη ενός στόχου ή υποστόχου, γίνεται αυτόματα εγγραφή στην πηγή που αποστέλλει τα ζητούμενα γεγονότα. Μετά την εγγραφή αναμένεται η λήψη του σχετικού γεγονότος προτού προχωρήσει στην εκτέλεση των αντίστοιχων ενεργειών. Επίσης μετά τη λήψη του γεγονότος γίνεται αυτόματα διαγραφή από την αντίστοιχη πηγή. Με αυτόν τον τρόπο υλοποιείται η ανατροφοδότηση από τη φάση 2 στη φάση 1 που φαίνεται στο Σχήμα 4-7. Ακόμη, αν ο σχετικός μηχανισμός CEP υποστηρίζει τη δυναμική δημιουργία και διαχείριση νέων τύπων γεγονότων (οιονεί δημιουργία νέων πηγών γεγονότων) με χρήση προτύπων σύνθετων γεγονότων (CEPat), τότε πριν την επιδίωξη ενός στόχου δημιουργείται και καταχωρείται ένα αντίστοιχο πρότυπο και ακολούθως γίνεται εγγραφή για τα σχετικά γεγονότα. Επίσης μετά τη λήψη του αναμενόμενου γεγονότος μπορεί να γίνει διαγραφή και να καταργηθεί το αντίστοιχο πρότυπο (CEPat).

Για την καλύτερη κατανόηση των παραπάνω δίνεται ένα απλό μεν αλλά ενδεικτικό παράδειγμα χρήσης των SANs για αποτύπωση της επίγνωσης κατάστασης. Έστω ότι μας ενδιαφέρει να γνωρίζουμε αν η επιχειρησιακή διαδικασία – στόχος εκτελείται με αργό ρυθμό και έστω ότι κατά την έναρξη και τον τερματισμό της στέλνει αντίστοιχα γεγονότα (με χρονοσήμανση). Τότε ο στόχος (goal) του SAN θα μπορούσε να είναι η «ανίχνευση αργοπορίας» και να αναλύεται στους υπο-στόχους «καταγραφή έναρξης διαδικασίας», «καταγραφή τέλους διαδικασίας» και «εντοπισμός μεγάλης διάρκειας εκτέλεσης». Ο μεν πρώτος υπο-στόχος επιτυγχάνεται με την λήψη ενός γεγονότος έναρξης από τη διαδικασία και αντιδρά ενεργοποιώντας μια ενέργεια καταγραφής της ώρας έναρξης. Ο δε δεύτερος επιτυγχάνεται με τη λήψη του γεγονότος τέλους της διαδικασίας και αντίστοιχα καταγράφει την ώρα τέλους. Επίσης ενεργοποιεί δευτερογενώς και τον τελευταίο υπο-στόχο «εντοπισμός μεγάλης διάρκειας εκτέλεσης» όπου εκεί γίνεται ο υπολογισμός της διάρκειας εκτέλεσης της διαδικασίας και σύγκριση με τον πρότυπο χρόνο ολοκλήρωσης. Αν η διάρκεια εκτέλεσης είναι π.χ. 10% μεγαλύτερη από τον πρότυπο χρόνο τότε ο υπο-στόχος επιτυγχάνεται (ανιχνεύθηκε μεγάλη διάρκεια) και μαζί με αυτόν και ο αρχικός στόχος αφού ανιχνεύθηκε η αργοπορία της διαδικασίας. Αν πάλι η διάρκεια ήταν κανονική τότε ο τελευταίος υπο-στόχος αποτυγχάνει και μαζί με αυτόν και ο αρχικός στόχος. Δηλαδή δεν ανιχνεύτηκε κατάσταση αργοπορίας. Στη συνέχεια της διατριβής θα δοθούν περισσότερα στοιχεία για τα SANs.



Σχήμα 4-8. Παράδειγμα αποτύπωσης του τρόπου αντίληψης της αργοπορίας μιας διαδικασίας

4.3.3.2 Υποσύστημα Επίγνωσης Κατάστασης

Η δεύτερη φάση του προτεινόμενου πλαισίου υλοποιείται με το Υποσύστημα Επίγνωσης Κατάστασης ή Situation Awareness Subsystem (SAS). Το υποσύστημα αυτό βασίζεται στη χρήση των SANs (α) για την αποτύπωση του τρόπου επίγνωσης της κατάστασης των διαδικασιών – στόχων και (β) για την ανίχνευση (επίγνωση) των αλλαγών καταστάσεων και της εμφάνισης των αναγκών προσαρμογής. Δηλαδή το υποσύστημα αυτό έχει διπλό ρόλο. Πρώτον να υποστηρίξει τη διαδικασία αποτύπωσης της επίγνωσης κατάστασης σε SANs και δεύτερον να παρακολουθήσει τις διαδικασίες – στόχους και με τη βοήθεια των SANs να διαπιστώσει την αλλαγή καταστάσεων σε αυτές και να ανιχνεύσει την εμφάνιση αναγκών προσαρμογής.

Η υλοποίηση του συστήματος επίγνωσης κατάστασης που έχουμε αναπτύξει στο πλαίσιο της διατριβής απαρτίζεται από δύο εργαλεία λογισμικού: τον Επεξεργαστή SANs (SAN Editor) και τον Μηχανισμό (εκτέλεσης) SANs (SAN engine). Ο επεξεργαστής SANs είναι ένα εξειδικευμένο γραφικό περιβάλλον σχεδιασμού που παρέχει στους εμπειρογνώμονες όλα τα αναγκαία εργαλεία για τη δημιουργία και επεξεργασία SANs και τον καθορισμό των απαραίτητων πληροφοριών και ρυθμίσεων. Οπτικοποιεί δε τα SANs με δύο τρόπους: (α) ως μια δενδρική ιεραρχία από κόμβους ή (β) ως δένδρα με ιδιαίτερα σχήματα και χρώματα για τα διάφορα είδη κόμβων (δείτε το παραπάνω παράδειγμα). Τέλος, τα SANs εξάγονται και αποθηκεύονται ως αρχεία, τα οποία στη συνέχεια μπορούν να χρησιμοποιηθούν από τον μηχανισμό εκτέλεσης SANs. Το σύνολο των SANs που αφορούν μια διαδικασία – στόχο αποτελούν το μοντέλο επίγνωσης κατάστασης αυτής, όσον αφορά βέβαια την ανίχνευση των αναγκών προσαρμογής αυτής.

Ο Μηχανισμός εκτέλεσης SANs φορτώνει κι «εκτελεί» τα SANs που αφορούν μια διαδικασία – στόχο, με απώτερο στόχο την παρακολούθηση της εξέλιξης της εκτέλεσής τους κι επίσης

την ανίχνευση των αλλαγών των καταστάσεών τους. Για το σκοπό αυτό επιτελεί μια σειρά από λειτουργίες όπως η διάσχιση των γράφων των SANs, η διαχείριση στιγμιότυπων των SANs, η διαχείριση των πλαισίων εκτέλεσης, λήψη και contextualization των γεγονότων, η δημιουργία (όταν χρειάζεται) προτύπων σύνθετων γεγονότων (CEPats) και η καταχώρησή τους στον μηχανισμό CEP (δείτε υποσύστημα Διαύλου Γεγονότων & Επεξεργασίας Σύνθετων Γεγονότων), η εγγραφή σε και η διαγραφή από ροές γεγονότων και η εκτέλεση των απλών ενεργειών. Είναι φανερό ότι ο μηχανισμός εκτέλεσης των SANs πρόκειται για ένα σύνθετο λογισμικό αποτελούμενο από πολλές συνεργαζόμενες μονάδες. Αυτές επικοινωνούν μέσω μιας προγραμματιστικής διεπαφής (API), η οποία επιτρέπει τη μεταξύ τους συνεργασία και επιπλέον καθιστά εύκολη την αντικατάστασή τους με εναλλακτικές υλοποιήσεις τους. Έτσι παρέχεται ένα ευέλικτο και προσαρμόσιμο εργαλείο. Περισσότερα για τον μηχανισμό εκτέλεσης των SANs δίνονται στις ενότητες «5.3 Λειτουργία και Εκτέλεση των SANs» και «6.1 Αρχιτεκτονική του Μηχανισμού Εκτέλεσης των SANs (SAN engine)».

4.3.4 Επιλογή και Πρόταση προσαρμογών

Η τρίτη φάση «**Επιλογή και Πρόταση προσαρμογών**» ενεργοποιείται όταν διαπιστωθεί κάποια ανάγκη προσαρμογής και απαντά στο δεύτερο ερευνητικό ερώτημα. Συγκεκριμένα απαντά στο πως γίνεται η επιλογή μιας προσαρμογής και πως αποτυπώνεται ο τρόπος επιλογής.

Στη φάση αυτή επιλέγεται μια (πρότυπη) προσαρμογή από ένα σύνολο εναλλακτικών και εξειδικεύεται στην εκάστοτε περίπτωση, με βάση τις τιμές των παραμέτρων της τρέχουσας κατάστασης και της ανάγκης προσαρμογής (ήτοι απόκλισης από τους στόχους και τα όρια των λειτουργιών της διαδικασίας). Ειδικότερα, αξιοποιούνται πληροφορίες από τα γεγονότα που έχουν ληφθεί και υπολογίζονται οι παράμετροι που ποσοτικοποιούν την εκάστοτε ανάγκη προσαρμογής. Εν συνεχεία επιλέγεται η πλέον κατάλληλη εναλλακτική προσαρμογή και εξειδικεύεται με βάση τις τιμές των παραμέτρων που υπολογίστηκαν καθώς και τις λοιπές διαθέσιμες πληροφορίες. Ανεξάρτητα από τον ακριβή τρόπο επιλογής της, η προσαρμογή προτείνεται στον ιδιοκτήτη ή στον υπεύθυνο ή στον διαχειριστή ή άλλο «μέτοχο» (stakeholder) της διαδικασίας για έλεγχο και έγκριση. Η δυνατότητα επιλογής για την εφαρμογή ή μη των προτεινόμενων προσαρμογών αποτελεί ένα πλεονέκτημα της προτεινόμενης λύσης. Αν η προσαρμογή γίνει δεκτή τότε ενεργοποιείται η επόμενη φάση.

4.3.4.1 Καινοτομία 2: Δυναμική επιλογή προσαρμογής από ένα σύνολο εναλλακτικών

Για τη δυναμική και ευέλικτη επιλογή και πρόταση προσαρμογών από ένα σύνολο διαθέσιμων εναλλακτικών προσαρμογών, εισάγεται η έννοια των Δεξαμενών Ενεργειών (Action Pools). Πρόκειται για μια επέκταση των Δικτύων Κατάστασης – Ενέργειας (SANs) που επιτρέπει τη μοντελοποίηση της επίγνωσης κατάστασης και της επιλογής των προσαρμογών με δυναμικά και εξελίξιμα χαρακτηριστικά. Πιο συγκεκριμένα, τα SANs ως έχουν, απαιτούν την εκ των προτέρων γνώση των καταστάσεων / αναγκών που μπορεί να εμφανιστούν αλλά και τις λεπτομέρειες των τρόπων αντιμετώπισής τους. Στην πράξη, ωστόσο, δεν είναι πάντα δυνατό να προβλεφθεί κάθε πιθανή περίπτωση που μπορεί να ανακύψει, ούτε και να προδιαγραφούν οι αντίστοιχες απαιτούμενες ενέργειες. Με την εισαγωγή των δεξαμενών ενεργειών καθίσταται δυνατή η σε πιο αφηρημένο επίπεδο

περιγραφή των στόχων και των απαιτούμενων ενεργειών, χωρίς όμως να παρέχονται οι λεπτομέρειες υλοποίησής τους. Οι λεπτομέρειες αυτές επιλέγονται και εξειδικεύονται κατά την εκτέλεση των SANs, σύμφωνα με κανόνες και παραμέτρους που έχουν τεθεί κατά τον σχεδιασμό. Πρακτικά οι δεξαμενές ενεργειών εισάγουν ένα πλαίσιο εργασίας που περιλαμβάνει τα εξής στοιχεία:

1. Την έννοια της αφηρημένης ενέργειας (abstract action), η οποία χρησιμοποιείται στα SANs κατά τη μοντελοποίηση. Μια αφηρημένη ενέργεια δεν αναλύεται περαιτέρω (σε υπο-στόχους) ή απλούστερες ενέργειες, αλλά προσδιορίζονται μόνο οι παράμετροι που θα επιτρέψουν την επιλογή μιας εναλλακτικής υλοποίησής της κατά το χρόνο εκτέλεσης.
2. Δεξαμενή ενεργειών. Είναι ένα σύνολο από εναλλακτικές υλοποιήσεις μιας αφηρημένης ενέργειας. Δύναται να υλοποιηθεί είτε ως μια στατική συλλογή εναλλακτικών ενεργειών που παρέχεται κατά τον σχεδιασμό, είτε ένα δυναμικά προσδιοριζόμενο σύνολο ή λαμβανόμενο από κάποια εξωτερική υπηρεσία (όπως μια βάση δεδομένων ή μια υπηρεσία ιστού). Στη δεύτερη περίπτωση οι εναλλακτικές μπορεί να γίνονται διαθέσιμες ή να αποσύρονται αναλόγως με τις συνθήκες που επικρατούν.
3. Εναλλακτικές ενέργειες. Είναι πιθανές υλοποιήσεις μιας αφηρημένης ενέργειας. Παρότι επιδιώκουν τον ίδιο στόχο, εντούτοις δεν είναι όλες τους κατάλληλες για όλες τις πιθανές καταστάσεις / ανάγκες που μπορεί να ανακύψουν. Συνεπώς σε κάθε περίπτωση θα πρέπει να επιλέγεται εκείνη που ταιριάζει περισσότερο. Η φύση των εναλλακτικών δεν καθορίζεται ή περιορίζεται από το πλαίσιο εργασίας. Μπορεί να είναι απλές ενέργειες (όπως η επιλογή μιας προσαρμογής) ή πιο σύνθετες δομές (όπως ένα ολόκληρο SAN).
4. Ελάχιστο κοινό σύνολο μετα-δεδομένων. Είναι εκείνα που θα πρέπει να διαθέτουν όλες οι εναλλακτικές προκειμένου να είναι δυνατή η εφαρμογή ενός αλγορίθμου επιλογής. Κατ' αυτόν τον τρόπο τα μετα-δεδομένα χρησιμοποιούνται ως κριτήρια επιλογής.
5. Μέθοδος αναζήτησης και επιλογής. Είναι μια τεχνική ή αλγόριθμος που χρησιμοποιώντας τα μετα-δεδομένα των εναλλακτικών, επιλέγουν εκείνες που ταιριάζουν περισσότερο στις παραμέτρους των αφηρημένων ενεργειών αλλά και τις πληροφορίες που έχουν ληφθεί από τα γεγονότα (context). Είναι δυνατό να επιλεγεί καμία, μία ή περισσότερες εναλλακτικές, ως πιθανές υλοποιήσεις μιας αφηρημένης ενέργειας. Στο πλαίσιο της παρούσας διατριβής έχουμε υλοποιήσει δύο απλές πολυκριτηριακές μεθόδους (MCDM), την Απλή Τεχνική Βαθμολόγησης Πολλαπλών Κριτηρίων (Simple Multi-Attribute Rating Technique, SMART) για χρήση με αριθμητικά μεταδεδομένα και την Γλωσσική Διατεταγμένη Σταθμισμένη Άθροιση (Linguistic Ordered Weighted Average, LOWA) για χρήση με γλωσσικά μεταδεδομένα.
6. Πολιτική επίλυσης. Όταν μια μέθοδος αναζήτησης και επιλογής επιστρέψει περισσότερες από μία εναλλακτικές τότε εφαρμόζεται η πολιτική επίλυσης

προκειμένου να επιλεγεί μία από αυτές ως η τελική υλοποίηση μιας αφηρημένης ενέργειας, ή να συνδυάσει όλες τις εναλλακτικές σε μία συνθετότερη.

Η δεξαμενή ενεργειών, η μέθοδος αναζήτησης και επιλογής και η πολιτική επίλυσης είναι παράμετροι των αφηρημένων ενεργειών που πρέπει να προσδιορίζονται κατά τον σχεδιασμό ενός SAN. Τα μετα-δεδομένα των εναλλακτικών απ' την άλλη, προσδιορίζονται κατά τον σχεδιασμό και ανάπτυξη των δεξαμενών ενεργειών. Εν κατακλείδι σημειώνουμε ότι οι δεξαμενές ενεργειών (ως πλαίσιο εργασίας) προσθέτει στα SANs ευελιξία τόσο σε επίπεδο μοντελοποίησης (αφηρημένες ενέργειες) όσο και σε επίπεδο εκτέλεσης (δυναμικά προσδιοριζόμενα σύνολα εναλλακτικών). Περισσότερες πληροφορίες για τις δεξαμενές ενεργειών δίνονται στο κεφάλαιο «9 Δυναμικά Εξελισσόμενα SANs».

4.3.4.2 Υποσύστημα πρότασης προσαρμογών διαδικασιών

Αφού επιτευχθεί η επίγνωση της κατάστασης μιας διαδικασίας – στόχου και διαπιστωθεί ότι υπάρχει ανάγκη για διόρθωσή της, θα πρέπει ακολούθως να επιλεγεί μια κατάλληλη προσαρμογή. Η δραστηριότητα αυτή αποτελεί το αντικείμενο του υπο-συστήματος πρότασης προσαρμογής διαδικασιών ή Process Adaptation Recommender (**PAR**). Υπενθυμίζουμε ότι η προσέγγιση που ακολουθεί το πλαίσιο εργασίας είναι να προτείνει στον υπεύθυνο ή ιδιοκτήτη της διαδικασίας μία ή περισσότερες προσαρμογές κι έπειτα αυτός να επιλέγει αν θα αποδεχτεί κάποια από αυτές (και ποια) ή αν θα τις απορρίψει όλες. Επίσης ο PAR μπορεί να προτείνει και την κατάργηση προηγούμενων προσαρμογών που έχουν ήδη γίνει δεκτές και εφαρμοστεί. Αν ο υπεύθυνος της διαδικασίας το επιθυμεί, ο PAR μπορεί να αποστέλλει τις προσαρμογές άμεσα στο επόμενο υπο-σύστημα για εφαρμογή.

Στο προτεινόμενο πλαίσιο εργασίας ο PAR βασίζεται στη χρήση SANs για την αποτύπωση του τρόπου επιλογής (και πρότασης) των προσαρμογών. Κατ' αυτόν τον τρόπο αξιοποιείται το ίδιο περιβάλλον σχεδιασμού και εκτέλεσης τόσο για την επίγνωση κατάστασης (φάση 2) όσο και για την πρόταση προσαρμογών, γεγονός που διευκολύνει την εργασία των εμπειρογνώμων. Σε αυτή τη φάση οι εμπειρογνώμονες καλούνται να προσδιορίσουν τις παραμέτρους που ποσοτικοποιούν την επίτευξη των στόχων ή την υπέρβαση των ορίων λειτουργίας της διαδικασίας. Ακόμη καλούνται να καθορίσουν τις μεθόδους επιλογής εναλλακτικών προσαρμογών που χρησιμοποιούνται στην εκάστοτε περίπτωση (κατάσταση, συνδυασμούς τιμών των παραμέτρων).

Ο PAR υποστηρίζει και τη δυναμική επιλογή προσαρμογών με τη χρήση δεξαμενών ενεργειών. Αυτές μπορεί να είναι συγκεκριμένες προσαρμογές ή πρότυπα προσαρμογών (template adaptations). Στη δεύτερη περίπτωση, αφού επιλεγεί μία πρότυπη προσαρμογή και προτού αυτή χρησιμοποιηθεί, απαιτείται η εξειδίκευσή της για την εκάστοτε διαδικασία – στόχο. Δηλαδή από το πρότυπο προκύπτει ένα στιγμιότυπο της προσαρμογής, εξειδικευμένο (instantiated) για ένα συγκεκριμένο στιγμιότυπο της διαδικασίας. Ο PAR παρέχει ορισμένες ευκολίες για την εργασία αυτή.

Οι προσαρμογές που παράγει ο PAR θα πρέπει να έχουν τη μορφή και το περιεχόμενο που απαιτείται από το εκάστοτε σύστημα που θα κληθεί να τις εφαρμόσει. Το πλαίσιο εργασίας περιλαμβάνει ένα τέτοιο υπο-σύστημα για το οποίο η μορφή και το περιεχόμενο των προσαρμογών περιγράφεται στην επόμενη ενότητα.

4.3.5 Εφαρμογή προσαρμογών

Στην τέταρτη και τελευταία φάση «**Εφαρμογή προσαρμογών**», οι προσαρμογές που έγιναν δεκτές (κατά την προηγούμενη φάση) εφαρμόζονται στη διαδικασία – στόχο και την τροποποιούν κατάλληλα. Οι αλλαγές αυτές θα επηρεάσουν τον τρόπο λειτουργίας της διαδικασίας με σκοπό να θεραπευτεί το όποιο πρόβλημα και έτσι αυτή να συνεχίσει να επιτυγχάνει τους στόχους της. Η τέταρτη φάση απαντά στο τρίτο ερευνητικό ερώτημα.

Όπως έχουμε ήδη προαναφέρει, στην παρούσα διατριβή έχουμε επιλέξει να εστιάσουμε σε διαδικασίες αποτυπωμένες με την BPMN 2.0 κι επίσης να εστιάσουμε σε προσαρμογές που δεν έχουν απαιτήσεις για προηγούμενη (a priori) επεξεργασία ή προετοιμασία της διαδικασίας – στόχου προκειμένου αυτή να μπορέσει να τις δεχτεί. Ένας τέτοιος τρόπος που εντοπίστηκε στη βιβλιογραφία στηρίζεται στη χρήση τεχνικών θεματοστρεφούς προγραμματισμού (Aspect-Oriented Programming) για την εφαρμογή των προσαρμογών (AO4BPEL (Charfi and Mezini, 2007), AO4BPMN (Charfi et al., 2010) και AO4BPMN 2.0 (Witteborg et al., 2014)).

4.3.5.1 Καινοτομία 3: Μέθοδος θεματοστρεφούς προσαρμογής για διαδικασίες BPMN2 (FlexiBPMN2.0)

Για την εφαρμογή προσαρμογών σε επιχειρησιακές διαδικασίες μοντελοποιημένων με τη BPMN 2.0, προτείνεται μια νέα μέθοδος θεματοστρεφούς προσαρμογής (aspect-oriented adaptation) που καλείται Flexible BPMN 2.0 (συντομογραφικά **FlexiBPMN2.0**). Η μέθοδος αυτή βασίζεται στην μέθοδο AO4BPMN που έχει προταθεί από τους (Charfi and Mezini, 2007) και την επεκτείνει. Συγκεκριμένα η FlexiBPMN2.0 χρησιμοποιεί την BPMN 2.0 που είναι εκτελέσιμη γλώσσα, σε αντίθεση με την BPMN 1 που χρησιμοποιεί η AO4BPMN, η οποία αποσκοπεί μόνο στη μοντελοποίηση διαδικασιών. Έχει δε σχεδιαστεί για δυναμική εφαρμογή των προσαρμογών ενώ υποστηρίζει ακόμη την προσαρμογή μεμονωμένων στιγμιότυπων μιας διαδικασίας ή του συνόλου αυτών. Επίσης η FlexiBPMN2.0 έχει αρκετά κοινά σημεία με τη νεότερη έκδοση της AO4BPMN (Witteborg et al., 2014), αν και ακολουθούν άλλες προσεγγίσεις όσον αφορά στην εφαρμογή των προσαρμογών. Η μεν πρώτη, πριν την εκτέλεση μετασχηματίζει τα μοντέλα των διαδικασιών σε άλλα που περιλαμβάνουν και τις σχετικές προσαρμογές. Αντίθετα η προτεινόμενη εισάγει τις προσαρμογές στη ροή εργασιών της κανονικής διαδικασίας κατά το χρόνο εκτέλεσης, χωρίς να απαιτείται μετασχηματισμός των αρχικών μοντέλων.

Κεντρικές έννοιες στην FlexiBPMN2.0 είναι αυτές του θεματοστρεφούς προγραμματισμού, δηλαδή τα *Θέματα (Aspects)*, τα *Σημεία Τομών (Pointcuts)* και οι *Συμβουλές (Advices)*. Τα θέματα είναι σύνθετες δομές αποτελούμενες από ένα ή περισσότερα σημεία τομών και μία συμβουλή. Έχουν δε ένα συγκεκριμένο σκοπό και παρέχουν μια λειτουργία ανεξάρτητη μεν αλλά «τεμνόμενη» (cross-cutting) με τις κύριες λειτουργίες της κύριας διαδικασίας. Στο πλαίσιο αυτής της διατριβής τα θέματα έχουν ως σκοπό την εφαρμογή μιας συγκεκριμένης προσαρμογής στην αντίστοιχη διαδικασία – στόχο. Τα σημεία τομών είναι εκείνες οι δομές που επιτρέπουν την επιλογή των επιθυμητών σημείων και μερών στα μοντέλα διαδικασιών (καλούνται joint points), στα οποία μπορεί να συμβεί κάποια προσαρμογή. Η επιλογή αυτή μπορεί να γίνεται είτε με απλή αναφορά των εν λόγω σημείων (joint points) στα μοντέλα

των διαδικασιών ή ακόμη και με χρήση κατάλληλων ερωτημάτων (queries), εκφράσεων (expressions) και συνθηκών (conditions). Οι συμβουλές απ' την άλλη είναι διαδικασίες ή τμήματα διαδικασιών (process fragments) που υλοποιούν τις προσαρμογές, σύμφωνα πάντα με τους σκοπούς των θεμάτων που τις περιέχουν. Ο σκοπός μιας συμβουλής είναι να αντικαταστήσει τα επιλεγμένα σημεία τομών μιας διαδικασίας με μια προσαρμοσμένη διαδικασία (ή τμήμα διαδικασίας).

Οι προσαρμογές εφαρμόζονται όταν πρόκειται να εκτελεστούν προκαθορισμένες εργασίες ή τμήματα των διαδικασιών, τα οποία είναι σημεία τομής και εφόσον συντρέχουν οι απαιτούμενες συνθήκες εφαρμογής τους. Πρακτικά, κάθε φορά που η εκτέλεση μιας διαδικασίας φτάνει σε μια εργασία ή σε μια πύλη απόφασης (και πριν αυτή εκτελεστεί), εξετάζεται αν αποτελεί σημείο τομής για κάποιο θέμα. Αν ισχύει κάτι τέτοιο τότε η συμβουλή του αντίστοιχου θέματος θα εκτελεστεί αντί της εργασίας ή πύλης απόφασης που ακολουθεί.

Η μέθοδος FlexiBPMN2.0 μπορεί να συνδυαστεί με τα SANs ή κάποια άλλη μέθοδο επιλογής ή δημιουργίας θεματοστρεφών προσαρμογών. Κατ' αυτόν τον τρόπο καθίσταται εφικτός ο οδηγούμενος από γεγονότα, δυναμικός προσδιορισμός και εφαρμογή προσαρμογών σε διαδικασίες αποτυπωμένες σε BPMN 2.0. Περισσότερα στοιχεία για τη μέθοδο FlexiBPMN2.0 δίνονται στο αντίστοιχο κεφάλαιο της διατριβής («8 Θεματοστρεφής προσαρμογή – Η μέθοδος FlexiBPMN2.0»).

Όσον αφορά την αποτύπωση των προσαρμογών ένας τρόπος είναι να μοντελοποιούνται ως «διαδικασίες» ή «τμήματα διαδικασιών», τα οποία «εισάγονται» σε προκαθορισμένα σημεία μέσα στην αρχική διαδικασία αλλάζοντας πιθανώς και τη ροή των εργασιών της (workflow). Οι αλλαγές αυτές γίνονται είτε στο σύνολο είτε σε μεμονωμένα στιγμιότυπα των διαδικασιών. Μπορεί ακόμη να επηρεάζουν είτε τις νέες μόνο εκτελέσεις (στιγμιότυπα) είτε και αυτά που ήδη εκτελούνται. Οι υλοποιήσεις αυτής της προσέγγισης παρεμβάλλονται στον τρόπο λειτουργίας του μηχανισμού εκτέλεσης των διαδικασιών και «παγιδεύουν» τρόπον τινά την εκτέλεση των εργασιών. Εκεί μπορεί να εκτελείται η λογική της εκάστοτε εφαρμοζόμενης προσαρμογής και να τροποποιούνται ή αντικαθιστώνται οι ενέργειες που προβλέπει η αρχική διαδικασία. Αυτό όμως γίνεται με τρόπο «διάφανο», δηλαδή η διαδικασία δεν μπορεί να αντιληφθεί το γεγονός της προσαρμογής³⁶.

4.3.5.2 Υποσύστημα προσαρμοστικής εκτέλεσης διαδικασιών

Η παρουσίαση του προτεινόμενου πλαισίου εργασίας ολοκληρώνεται με το υπο-σύστημα προσαρμοστικής εκτέλεσης των διαδικασιών ή Adaptive Process Execution Subsystem (APES). Ουσιαστικά πρόκειται για ένα μηχανισμό εκτέλεσης επιχειρησιακών διεργασιών, ο οποίος επιπλέον έχει την ικανότητα να δέχεται περιγραφές προσαρμογών από τον PAR (ή και χειρονακτικά) και να τις εφαρμόζει στις διαδικασίες που εκτελεί. Υποστηρίζει δε επιχειρησιακές διαδικασίες εκπεφρασμένες με τη γλώσσα περιγραφής BPMN 2.0.

³⁶ Αυτή η προσέγγιση έχει χρησιμοποιηθεί για διαδικασίες που υλοποιούνται ως συνθέσεις από υπηρεσίες ιστού (web service compositions). Εκεί οι προσαρμογές εφαρμόζονται τροποποιώντας τις κλήσεις (invocations) προς τα web services.

Όσον αφορά την εφαρμογή των προσαρμογών των επιχειρησιακών διαδικασιών, το APES βασίζεται στη μέθοδο Aspect-Orientation for BPMN 2.0 (**FlexiBPMN2.0**). Οι προσαρμογές πρέπει να είναι αποτυπωμένες ως «θέματα» (aspects) έτσι ώστε να είναι αξιοποιήσιμες από το APES και τη μέθοδο. Κάθε προσαρμογή – θέμα πρέπει να περιέχει τα εξής στοιχεία:

1. Μοναδικό αναγνωριστικό, διαδικασία – στόχο (μπορεί να είναι πολλές), αριθμός και χρονολογία έκδοσης, στοιχεία του εξουσιοδοτούντος, λοιπά μετα-δεδομένα.
2. Είδος και Εύρος προσαρμογής. Για παράδειγμα εφαρμογή σε όλα τα στιγμιότυπα, μόνο σε νέα στιγμιότυπα, άρση υφιστάμενης προσαρμογής (τότε παραλείπονται τα σημεία 3, 4).
3. Τα σημεία ή τμήματα της διαδικασίας στα οποία εφαρμόζεται (σημεία τομής, poinctcuts). Επίσης οι συνθήκες που πρέπει να συντρέχουν για κάθε ένα από αυτά προκειμένου να θεωρηθεί ενεργό σημείο τομής.
4. Οι συμβουλές (advices), δηλαδή οι προσαρμοσμένες εργασίες που εισάγονται στη θέση των σημείων τομής, ως τμήματα διαδικασιών αποτυπωμένων σε BPMN 2.0. Αυτές μπορούν ακόμη να εκτελούν (ως μέρος τους) και τις εργασίες του σημείου τομής που αντικαθιστούν (εντολή Proceed).

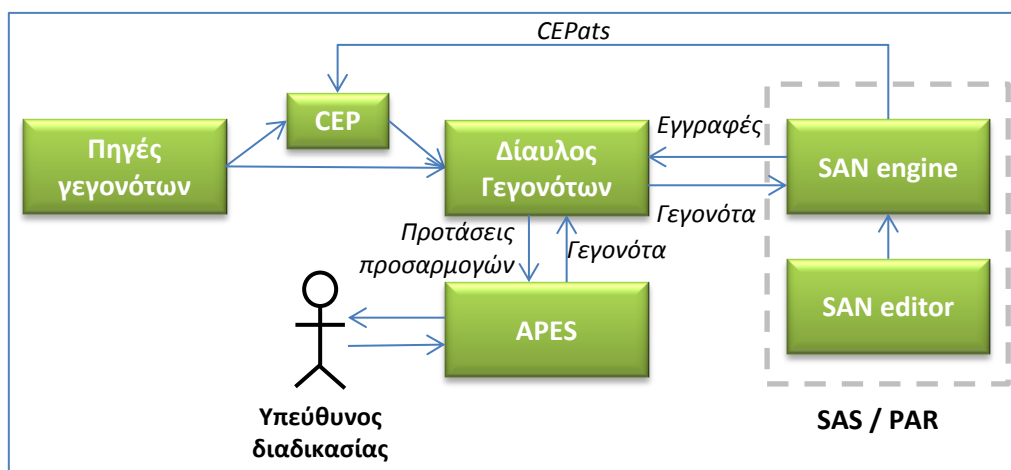
Η χρήση του υπο-συστήματος προσαρμοστικής εκτέλεσης διαδικασιών (APES) είναι προαιρετική, αφού το πλαίσιο εργασίας μπορεί να φθάσει μέχρι την επιλογή και πρόταση των προσαρμογών χωρίς όμως να προχωρήσει και στην εφαρμογή τους. Μετά είναι ευθύνη του ιδιοκτήτη ή υπευθύνου της διαδικασίας να επιλέξει την εφαρμογή ή μη κάποιας πρότασης προσαρμογής. Περισσότερα στοιχεία για το σύστημα APES δίνονται στο κεφάλαιο «9 Δυναμικά Εξελισσόμενα SANs».



Σχήμα 4-9. Συσχέτιση φάσεων, μεθόδων και συστημάτων του προτεινόμενου πλαισίου εργασίας

4.3.6 Υλοποίηση της προτεινόμενης προσέγγισης

Στο πλαίσιο της διατριβής έχει σχεδιαστεί και αναπτυχθεί μια πλατφόρμα λογισμικού που υλοποιεί την προτεινόμενη προσέγγιση, την οποία καλούμε «Πλατφόρμα Πρότασης Προσαρμογών» ή Adaptation Recommendation Platform (ARP). Έτσι τονίζεται η σημασία που δίνουμε στην πρόταση των προσαρμογών πριν την εφαρμογή τους. Η πλατφόρμα συμπεριλαμβάνει τα υποσυστήματα λογισμικού, τις μεθόδους και τις τεχνικές που παρουσιάστηκαν παραπάνω ενώ η λογική αρχιτεκτονική της δίνεται στο διάγραμμα που ακολουθεί.



Σχήμα 4-10. Λογική αρχιτεκτονική της υλοποίησης του πλαισίου εργασίας

Σημειώνουμε ότι θεωρούμε καινοτόμα τη συνολική προτεινόμενη προσέγγιση καθώς και το πλαίσιο εργασίας, ως σύνθεση συγκεκριμένων καινοτόμων τεχνικών και μεθόδων για την απάντηση των ερευνητικών ερωτημάτων που έχουν τεθεί.

Εν κατακλείδι επαναλαμβάνουμε ότι η προτεινόμενη προσέγγιση πρέπει να εξειδικεύεται και να συγκεκριμενοποιείται στην εκάστοτε διαδικασία – στόχο από κάποιον εμπειρογνώμονα. Αυτό που προτείνεται στην παρούσα διατριβή είναι το πλαίσιο εργασίας, τα μεθοδολογικά εργαλεία που χρησιμοποιεί και οι υλοποιήσεις αυτών που θα παρουσιαστούν σε επόμενα κεφάλαια.

4.3.7 Αντιστοίχιση του πλαισίου εργασίας με τα ερευνητικά ερωτήματα

Στην ενότητα αυτή αντιστοιχίζονται οι τέσσερις φάσεις του προτεινόμενου πλαισίου εργασίας με τα ερευνητικά ερωτήματα που τέθηκαν παραπάνω στην ενότητα «4.2 Ερευνητικά Ερωτήματα» καθώς και τις επιμέρους παραμέτρων αυτών. Επίσης αναφέρονται και οι σχετικές μας δημοσιεύσεις.

Πίνακας 4-2. Αντιστοίχιση του προτεινόμενου πλαισίου εργασίας με τα ερευνητικά ερωτήματα

Ερευνητικό ερώτημα	Προτεινόμενο πλαίσιο εργασίας	Καινοτομία	Δημοσιεύσεις	Κεφ.
1. Αντίληψη της ανάγκης προσαρμογής με χρήση γεγονότων	<ul style="list-style-type: none"> Χρήση γεγονότων (events) και τεχνικών επεξεργασίας σύνθετων γεγονότων (CEP) για την επίτευξη οδηγούμενης από γεγονότα επίγνωσης κατάστασης (event-driven situation awareness). Χρήση των SANs για μοντελοποίηση της επίγνωσης κατάστασης κάθε διαδικασίας – στόχου. Αξιοποιούνται τα γεγονότα. (Υποσύστημα SAS). 	<ul style="list-style-type: none"> Τα SANs ως μέθοδος μοντελοποίησης της επίγνωσης κατάστασης διαδικασιών, οδηγούμενης από γεγονότα. 	<p>[J1] “Dynamic Event Subscriptions in Distributed Event Based Architectures.”</p> <p>[C9] “A Goal Driven Dynamic Event Subscription Approach.”</p> <p>[C1] “Context Management in Event Marketplaces.”</p>	5, 6, 7
2. Προσαρμογή επιχειρησιακών διαδικασιών οδηγούμενη από την επίγνωση κατάστασης	<ul style="list-style-type: none"> Χρήση των SANs για μοντελοποίηση του τρόπου επιλογής προσαρμογών. (Υποσύστημα PAR). 	<ul style="list-style-type: none"> Τα SANs ως μέθοδος μοντελοποίησης στοχοστρεφών στρατηγικών επιλογής. 	[C10] “Service Adaptation Recommender in the Event Marketplace: Conceptual View.”	5, 9
	<ul style="list-style-type: none"> Χρήση των Δεξαμενών Ενεργειών για δυναμική επιλογή προσαρμογών. (Υποσύστημα PAR). 	<ul style="list-style-type: none"> Δεξαμενές Ενεργειών (επέκτασης των SANs) για τη δυναμική επιλογή πρότυπων προσαρμογών διαδικασίες – στόχους. 	<p>[C6] “Collaborative Process Flexibility Using Multi-Criteria Decision Making.”</p> <p>[C5] “A Framework for Situation-Aware Adaptation of Service-Based Applications.”</p>	9, 10
3. Εφαρμογή προσαρμογών	<ul style="list-style-type: none"> Χρήση της μεθόδου FlexiBPMN2.0 για εφαρμογή των προσαρμογών (Υποσύστημα APES). 	<ul style="list-style-type: none"> Μέθοδος FlexiBPMN2.0 για την δυναμική εφαρμογή προσαρμογών σε διαδικασίες μοντελοποιημένες με την BPMN2.0. 	[C7] “An Aspect Oriented Approach for Implementing Situational Driven Adaptation of BPMN2.0 Workflows.”	8

4.3.8 Σύγκριση με τη βιβλιογραφία

Στην ενότητα αυτή επιχειρούμε μια συνοπτική σύγκριση του προτεινόμενου πλαισίου εργασίας με ενδεικτικές εργασίες που έχουν προταθεί στη βιβλιογραφία. Η σύγκριση αυτή γίνεται με βάση τα ερευνητικά ερωτήματα που θέσαμε καθώς και τις επιμέρους παραμέτρους τους (Πίνακας 4-3).

Πίνακας 4-3. Σύγκριση με τη βιβλιογραφία

Ερευν. Ερωτ.	Παράμετροι	Προτάσεις στη Βιβλιογραφία	Πρόταση Διατριβής
Ανίχνευση της ανάγκης προσαρμογής με χρήση γεγονότων	Επίγνωση Κατάστασης με χρήση γεγονότων	Χρήση CEP για επίγνωση κατάστασης ή πλαισίου	Με συνδυασμό SANs και CEP
	Αποτύπωση του τρόπου ανίχνευσης (επίγνωση κατάστασης)	<ul style="list-style-type: none"> • Κανόνες ECA • Πρότυπα σύνθετων γεγονότων (CEPats) ή αντίστοιχοι κανόνες • Θέματα παρακολούθησης (AO4BPMN) 	Με SANs, CEPats και εγγραφή σε πηγές γεγονότων
	Ανίχνευση ανάγκης προσαρμογής	<ul style="list-style-type: none"> • Αποτυχία κατά τη εκτέλεση υπηρεσίας ή εφαρμογής • Παραβίαση κάποιας SLA ή επιδείνωση κάποιου δείκτη ποιότητας (QoS) • Αλλαγή συνθηκών του περιβάλλοντος • Θέματα παρακολούθησης (AO4BPMN) 	Γεγονός που σηματοδοτεί την αλλαγή μιας κατάστασης
Επιλογή προσαρμογής	Επιλογή προσαρμογών οδηγούμενη από την κατάσταση (ανάγκη)	<ul style="list-style-type: none"> • Βασισμένη σε κανόνες (rule-based) • Αλγόριθμοι σχεδιασμού (planning) • Θέματα προσαρμογής (AO4BPMN) 	Με Δεξαμενές ενεργειών
	Αποτύπωση του τρόπου επιλογής	<ul style="list-style-type: none"> • Κανόνες επιλογής • Pre- / post- conditions • Θέματα προσαρμογής (AO4BPMN) 	Με SANs
Εφαρμογή προσαρμογής	Εφαρμογή προσαρμογής	<ul style="list-style-type: none"> • Στατική εφαρμογή προσαρμογών (απαιτείται compile) • Δυναμική εφαρμογή (π.χ. AO4BPEL, AO4BPMN 2.0) • Σύνθεση υπηρεσιών ή μερών διαδικασιών • Δυναμική αντιστοίχιση εργασιών με υπηρεσίες • Χρήση προσαρμοστικού μηχανισμού εκτέλεσης διαδικασιών BPEL 	<ul style="list-style-type: none"> • Δυναμική εφαρμογή προσαρμογών • FlexiBPMN2.0 (με χρήση προσαρμοστικού μηχανισμού εκτέλεσης διαδικασιών BPMN2)

Προετοιμασία διαδικασιών	<ul style="list-style-type: none"> • Δεν απαιτείται για AOP-based τεχνικές • Άλλες προσεγγίσεις απαιτούν προσθήκη μετα-δεδομένων ή επισημάνσεων (annotations) 	• Δεν απαιτείται κάποια προετοιμασία
Αποτύπωση προσαρμογών	<ul style="list-style-type: none"> • Ως σύνολα υπηρεσιών • Ως μέρη διαδικασιών • Ως θέματα (για BPMN 1.0 και BPEL διαδικασίες) 	Ως θέματα για BPMN 2.0 διαδικασίες

4.3.8.1 Σύγκριση με το CEVICHE

Ειδικότερα για το CEVICHE (Hermosillo, 2012) οφείλεται ένας ιδιαίτερος σχολιασμός και σύγκριση με την πρόταση της διατριβής. Ο λόγος είναι ότι και τα δύο έχουν αρκετή επικάλυψη ενδιαφερόντων και ιδεών. Μολαταύτα οι στόχοι και οι προτεραιότητες των δύο προσεγγίσεων δεν ταυτίζονται απόλυτα με αποτέλεσμα να καταλήγουν διαφορετικές προτάσεις. Το σύστημα CEVICHE παρουσιάστηκε στην ενότητα 3.7.

Πίνακας 4-4. Σύγκριση της πρότασης της διατριβής με το CEVICHE

Κριτήριο	CEVICHE	Πρόταση διατριβής
Μέθοδος	Εξειδικεύεται σε διαδικασίες BPEL, με επίγνωση πλαισίου αποκλειστικά μέσω CEP και με δική του γλώσσα για τους κανόνες προσαρμογής (ABPL)	Προτείνει τα SANs, ένα μοντέλο αντίδρασης σε αλλαγές καταστάσεων και πλαισίου, πλήρως διαχωρισμένο από τις γλώσσες που αφορούν την επεξεργασία σύνθετων γεγονότων, τη διαχείριση του πλαισίου και την αντίδραση στις αλλαγές αυτές. Μεθοδολογικά τα SANs είναι ανεξάρτητα και από τη γλώσσα αποτύπωσης των διαδικασιών αλλά στην δική μας υλοποίηση έχουμε χρησιμοποιήσει τη γλώσσα BPMN 2
Οδηγούμενη από γεγονός	Ναι. Με δική του γλώσσα για τους κανόνες επεξεργασίας και plug-in για τον εκάστοτε CEP engine (ABPL)	Ναι. Δέχεται οποιαδήποτε γλώσσα κατανοητή από τον εκάστοτε CEP engine. Απαιτείται plug-in.
Επίγνωση κατάστασης	Εμμέσως. Με χρήση CEP	Ναι. Με βάση τα μοντέλα των SANs σε συνδυασμό με CEP
Πλαισιοποίηση	Έμμεσα. Με λήψη σύνθετων γεγονότων	Ναι. Με ερωτήματα πλαισιοποίησης στους κόμβους κατάστασης των SANs (εκτελούνται με τη λήψη γεγονότων). Στην υλοποίησή μας είναι σε EP-SPARQL. (Δείτε ενότητες 5.3.3 και 5.4)
Διαχείριση πλαισίου	Δεν παρέχεται	Ναι, με κανόνες διαχείρισης πλαισίου που ορίζονται στους κόμβους συνθήκης πλαισίου των SANs. Στην υλοποίησή μας αυτοί αποτυπώνονται σε JavaScript ή Java με κατάλληλο API για πρόσβαση στις πληροφορίες πλαισίου. (Δείτε ενότητες 5.3.3)

Κριτήριο	CEVICHE	Πρόταση διατριβής
Αντίδραση σε αλλαγές καταστάσεων και πλαισίου	Ναι. Γίνεται έλεγχος των κανόνων προσαρμογής που βρίσκονται στο αποθετήριο κανόνων.	Ναι. Με χρήση γλώσσας εκτέλεσης εργασιών. Στην υλοποίησή μας αυτή είναι η JavaScript ή η Java, με API για επικοινωνία με υπηρεσίες ιστού, αποστολή γεγονότων και δημιουργία προσαρμογών από πρότυπα
Αρχικοποίηση εκτέλεσης διαδικασίας	Μετατροπή σε γράφους εκτέλεσης με στοιχεία SCA και αντιστοίχιση αυτών σε υπηρεσίες ιστού	Απευθείας εκτέλεση διαδικασιών BPMN 2 και εφαρμογή θεματοστρεφών προσαρμογών σε αυτές από τον μηχανισμό προσαρμοστικής εκτέλεσης διαδικασιών APES
Εύρος προσαρμογών	Προσθήκη, αφαίρεση, αλλαγή μεμονωμένων εργασιών των διαδικασιών με δυναμική αντιστοίχιση σε υπηρεσίες ιστού	Η μέθοδος FlexiBPMN2.0 επιτρέπει την προσθήκη, αφαίρεση, αλλαγή ολόκληρων μερών μιας διαδικασίας, την αναδρομολόγηση της ροής εργασιών (workflow rerouting) και τη διακοπή εκτελουμένων εργασιών (running task interruption)
Εφαρμογή προσαρμογών	Με ταίριασμα κανόνων	Με δύο τρόπους: <ul style="list-style-type: none"> • Σύγχρονα: σε κάθε βήμα της διαδικασίας ερωτάτε ο διαχειριστής προσαρμογών (χρησιμοποιεί SANS) • Ασύγχρονα: αποστολή γεγονότων πρότασης προσαρμογής από τον PAR στον μηχανισμό APES (εκτελούνται παράλληλα)
Προτάσεις προσαρμογών	Δεν παρέχονται	Ναι, με αποστολή γεγονότων πρότασης προσαρμογών και ρύθμιση του APES σε interactive mode

Θεωρούμε ότι η πρόταση της διατριβής, όσον αφορά την αποτύπωση του τρόπου προσαρμογής, είναι ευρύτερη και γενικότερη απ' ότι η προσέγγιση του CEVICHE, καθότι τα SANS λειτουργούν ως ένα μετα-μοντέλο που μπορεί να εξειδικευτεί και προσαρμοστεί στις εκάστοτε απαιτήσεις. Η υλοποίηση των SANS που έχουμε αναπτύξει παρέχει ένα σύστημα ανάλογο με του CEVICHE αλλά εστιασμένο σε διαδικασίες BPMN 2. Η γενικότητα όμως της πρότασης επιτρέπει την εύκολη εξειδίκευσή της και σε διαφορετικές απαιτήσεις μέσω διαφορετικών υλοποιήσεων. Φερ' ειπείν είναι εφικτή η αντικατάσταση του APES με έναν μηχανισμό για BPEL και δυναμικές θεματοστρεφείς προσαρμογές.

Ακόμη, η προτεινόμενη προσέγγιση προάγει έναν στοχοστρεφή και ιεραρχικά δομημένο τρόπο αποτύπωσης των κανόνων προσαρμογής (ή οποιουδήποτε τρόπου αντίδρασης στην αλλαγή καταστάσεων και πλαισίου), ενώ διαχωρίζει από το βασικό μοντέλο τους ορισμούς της επεξεργασίας σύνθετων γεγονότων, της πλαισιοποίησης, των εκφράσεων ελέγχου και διαχείρισης των συνθηκών πλαισίου και των εντολών εκτέλεσης ενεργειών. Θεωρούμε ότι η προτεινόμενη προσέγγιση υπερέρχει του CEVICHE και στο σημείο αυτό, αφού παρέχει ένα

«πλαίσιο σκέψης» στους σχεδιαστές για τον τρόπο αποτύπωσης των κανόνων αντίδρασης / προσαρμογής ενώ αυξάνει τη σαφήνεια και τη συντηρησιμότητα των μοντέλων.

Όσον αφορά τις διαδικασίες αυτές καθαυτές, η δική μας υλοποίηση (APES) υποστηρίζει διάφορους τύπους εργασιών (tasks) πέραν των υπηρεσιών ιστού (που αποκλειστικά υποστηρίζει η BPEL και κατ' επέκταση το CEVICHE). Ακόμη, το εύρος των προσαρμογών στην προτεινόμενη προσέγγιση είναι μεγαλύτερο σε σχέση με το CEVICHE αφού μπορεί να πραγματοποιήσει προσθήκη, αφαίρεση ή τροποποίηση ολόκληρων μερών των διαδικασιών κι όχι μόνο μεμονωμένων εργασιών, να αλλάζει δυναμικά τη ροή των εργασιών και να διακόπτει εκτελούμενες εργασίες. Επιπρόσθετο πλεονέκτημα της πρότασης της διατριβής θεωρούμε την αποστολή γεγονότων προτάσεως προσαρμογών.

Στα κεφάλαια που ακολουθούν παρουσιάζονται αναλυτικά οι καινοτομίες της διατριβής καθώς και τα αντίστοιχα συστήματα που τις περιλαμβάνουν. Πρώτα περιγράφονται τα Δίκτυα Κατάστασης – Ενέργειας (SANS), τα οποία χρησιμοποιούνται για τη μοντελοποίηση της ανίχνευσης των αναγκών προσαρμογής αλλά και των μεθόδων επιλογής προσαρμογών. Κατόπιν αναλύεται η μέθοδος FlexiBPMN2.0 που χρησιμοποιείται για την προσαρμοστική εκτέλεση διαδικασιών από το σύστημα APES. Τέλος παρουσιάζονται οι Δεξαμενές Ενεργειών (Action Pools), καθώς και το υποσύστημα PAR που κάνοντας χρήση των SANS και των δεξαμενών ενεργειών προτείνει, όποτε χρειάζεται, προσαρμογές για τις διαδικασίες – στόχους.

5 Δίκτυα Κατάστασης – Ενέργειας

Στην εισαγωγή της διατριβής προσδιορίσαμε ότι το ενδιαφέρον μας εστιάζεται σε διαδικασίες οδηγούμενες από γεγονότα (event-driven processes) με δυνατότητα προσαρμογής στις εκάστοτε συνθήκες του περιβάλλοντος. Αυτό αφενός απαιτεί τρόπους με τους οποίους οι διαδικασίες θα αποκτούν επίγνωση των συνθηκών αυτών καθώς και των αλλαγών που συμβαίνουν μέσα τους. Αφετέρου θα πρέπει να μπορούν να προσαρμόζονται στις νέες συνθήκες. Προς την κατεύθυνση αυτή είναι σημαντικό να έχουμε στη διάθεσή μας έναν τυπικό τρόπο καταγραφής που θα μας επιτρέπει να συσχετίζουμε τις συνθήκες του περιβάλλοντος με την αντίστοιχη λειτουργικότητα.

Στο παρόν κεφάλαιο παρουσιάζονται και περιγράφονται τα *Δίκτυα Κατάστασης – Ενέργειας* ή *Situation – Action Networks* (συντομογραφικά SANs, όρο που θα χρησιμοποιούμε εφεξής για λόγους συντομίας). Τα SANs είναι ένας τρόπος μοντελοποίησης της λειτουργικότητας διαδικασιών που βασίζεται στην ανάλυση των κύριων στόχων των διαδικασιών σε δίκτυα υποστόχων και ενεργειών. Θεωρούμε ότι οι στόχοι σχετίζονται με τις καταστάσεις του περιβάλλοντος και η επιδίωξή τους εξαρτάται από τις αλλαγές του. Τα SANs μπορούν να αποτελέσουν τη βάση για την μοντελοποίηση και ανάπτυξη συστημάτων ικανών να επιτελούν το έργο τους έχοντας επίγνωση της κατάστασης του περιβάλλοντος.

Κατ' αρχήν γίνεται μια σύντομη επισκόπηση ορισμένων σχετικών προσεγγίσεων που έχουν προταθεί στη βιβλιογραφία. Στη συνέχεια παρουσιάζεται το θεωρητικό υπόβαθρο των SANs που είναι απαραίτητο για την κατανόησή τους. Σε επόμενο κεφάλαιο παρουσιάζεται μια υλοποίησή τους που αναπτύχθηκε στο πλαίσιο της διατριβής.

5.1 Σχετικές προσεγγίσεις στη βιβλιογραφία

Στην ενότητα αυτή παρουσιάζουμε ορισμένες προσεγγίσεις και τεχνολογίες που διαθέτουν ενδιαφέροντα χαρακτηριστικά και οι οποίες έδωσαν χρήσιμες ιδέες για την υλοποίηση της προτεινόμενης λύσης. Το ενδιαφέρον προς αυτές εδράζεται στο γεγονός ότι επιτρέπουν (σε διαφορετικό βαθμό η κάθε μία) την δυναμική επιλογή και εκτέλεση ενεργειών για την επίτευξη των στόχων των συστημάτων. Κατ' αυτόν τον τρόπο ελαττώνουν την απαιτούμενη

προσπάθεια για τη μοντελοποίηση και ανάπτυξη των συστημάτων ή διαδικασιών, αφού δεν χρειάζεται πλέον να αναλύονται και κωδικοποιούνται εξαντλητικά όλα τα σενάρια χρήσεων ή πιθανών σφαλμάτων. Επιπλέον, είναι αρκετά γενικές για να μοντελοποιήσουν σχεδόν οποιοδήποτε στόχο και παράμετρο λειτουργίας κι όχι μόνο παραμέτρους ποιότητας υπηρεσίας (QoS) ή άλλες μη λειτουργικές παραμέτρους, όπως συμβαίνει σε πολλές άλλες προσεγγίσεις.

5.1.1 Ιεραρχικά Δίκτυα Εργασιών

Μια από τις ευρέως διαδεδομένες ιδέες για τη διαχείριση της πολυπλοκότητας των συστημάτων και διαδικασιών είναι η «ιεραρχική ανάλυση» (hierarchical decomposition). Το κύριο όφελός της είναι ότι κάθε εργασία σε ένα επίπεδο της ιεραρχίας, αναλύεται σε ένα μικρό αριθμό από απλούστερες εργασίες στο επόμενο επίπεδο. Κατ' αυτόν τον τρόπο το υπολογιστικό κόστος της εύρεσης του ορθού τρόπου διάταξης των εργασιών παραμένει μικρό. Απεναντίας, οι μη ιεραρχικές μέθοδοι αναλύουν μια εργασία απευθείας σε έναν μεγάλο αριθμό στοιχειωδών ενεργειών. Ωστόσο για προβλήματα μεγάλης κλίμακας αυτή η προσέγγιση καθίσταται μη πρακτική.

Μια ευρέως γνωστή μέθοδος σχεδιασμού (planning), ήτοι διάταξης συγκεκριμένων στοιχειωδών ενεργειών για την ολοκλήρωση μιας σύνθετης εργασίας, η οποία στηρίζεται στην ιεραρχική ανάλυση, είναι ο σχεδιασμός με **Ιεραρχικά Δίκτυα Εργασιών (Hierarchical Task Network planning ή HTN planning)**. Σύμφωνα με αυτή το αρχικό σχέδιο που περιγράφει την επίλυση ενός προβλήματος, απεικονίζεται ως μια υψηλού επιπέδου περιγραφή (high-level description) του τι πρέπει να γίνει, για παράδειγμα το «χτίσιμο ενός σπιτιού». Ακολούθως το σχέδιο αναλύεται και γίνεται πιο λεπτομερές εφαρμόζοντας ιεραρχική ανάλυση των σύνθετων εργασιών. Κάθε ενέργεια ανάλυσης μετατρέπει μια σύνθετη (υψηλότερου επιπέδου) εργασία σε ένα μερικώς διατεταγμένο σύνολο από εργασίες χαμηλότερου επιπέδου. Δηλαδή οι αναλύσεις των εργασιών εμπεριέχουν τη γνώση του πώς υλοποιείται η εκάστοτε σύνθετη εργασία. Για παράδειγμα, το χτίσιμο ενός σπιτιού μπορεί να αναλυθεί: στην απόκτηση άδειας οικοδομής, την επιλογή ενός κατασκευαστή, την εκτέλεση της κατασκευής και την εξόφληση του κατασκευαστή. Η διαδικασία αυτή μπορεί να συνεχιστεί μέχρις ότου παραμείνουν στο σχέδιο μόνο στοιχειώδεις ενέργειες που δεν αναλύονται περαιτέρω. Τυπικά, οι στοιχειώδεις ενέργειες είναι απλές εργασίες οι οποίες μπορούν να εκτελεστούν αυτόματα από κατάλληλο λογισμικό. Η ιεραρχική ανάλυση των εργασιών είναι μια «φυσική» απεικόνιση σε πολλά πρακτικά προβλήματα και πεδία, συμπεριλαμβανομένων του στρατιωτικού σχεδιασμού (Mitchell, 1997), στρατηγικών παιχνιδιών για Η/Υ (Hoang et al., 2005) αλλά και διαδικασιών παραγωγής (Nau et al., 2005).

Στην βασική μορφή του σχεδιασμού με HTNs, τα σχέδια παράγονται με απλές διαδοχικές αναλύσεις. Δηλαδή η μόνη μεταβολή που υφίσταται το αρχικό σχέδιο είναι η (ιεραρχική) ανάλυσή του σε στοιχειώδεις ενέργειες. Ένα μειονέκτημα του βασικού σχεδιασμού με HTNs είναι ότι οι αναλύσεις μπορεί να εφαρμόζονται αναδρομικά (recursively), για παράδειγμα η εργασία «περπάτημα» μπορεί να αναλυθεί σε «ένα βήμα» και μετά ξανά «περπάτημα». Για το λόγο αυτό τα παραγόμενα σχέδια μπορεί να έχουν απροσδιόριστο μήκος. Για τον ίδιο

λόγο σε ορισμένα προβλήματα, ειδικά εκείνα με συνδυαστικούς (conjunctive) στόχους, η χρήση της βασικής μορφής σχεδιασμού με HTNs είναι μάλλον αφύσικη και ακατάλληλη. Σε αυτές τις περιπτώσεις συχνά προτιμάται κάποια υβριδική προσέγγιση όπου οι αναλύσεις αναλύουν το αρχικό σχέδιο σε ένα μερικώς διατεταγμένο σχέδιο στοιχειωδών ενεργειών (partial-order planning). Αλγόριθμοι που μπορεί να τοποθετήσουν δύο ενέργειες σε ένα σχέδιο χωρίς όμως να προσδιορίσουν τη σειρά τους καλούνται «σχεδιαστές μερικής διάταξης» (partial-order planners).

Οι αλγόριθμοι για σχεδιασμό με HTNs (όπως και όλοι οι τυπικοί αλγόριθμοι σχεδιασμού) μοντελοποιούν μεταβολές του περιβάλλοντος καθώς αναζητούν μια αλληλουχία ενεργειών που θα επιτυγχάνει ένα στόχο. Τυπικά δεν λαμβάνεται κάποια είσοδος από το περιβάλλον μετά τη δημιουργία του σχεδίου, κατά τη διάρκεια της εκτέλεσής του. Συνεπώς οι τυπικοί αλγόριθμοι σχεδίασης θεωρούν ότι το περιβάλλον είναι σχετικά σταθερό και δεν υπόκειται σε πολλές σημαντικές αλλαγές.

5.1.2 Δυναμικός Σχεδιασμός

Ο νόμος του Merphy μας διδάσκει ότι και τα καλύτερα σχέδια συχνά αποτυγχάνουν. Το πρόβλημα είναι ότι σχεδόν σε όλα τα πραγματικά συστήματα μπορεί να εμφανιστούν απροσδόκητες περιστάσεις για τις οποίες οι όποιες προδιαγραφές και αντιδράσεις δεν ισχύουν ή είναι λάθος. Συνεπώς σε πραγματικές συνθήκες απαιτείται παρακολούθηση της εκτέλεσης των σχεδίων. Ο **δυναμικός σχεδιασμός (Dynamic Planning)** ή **επανασχεδιασμός (Re-planning)** αναφέρεται σε προσεγγίσεις που αν συμβεί κάτι απροσδόκητο μπορούν να δημιουργήσουν εναλλακτικά σχέδια που θα επιτύχουν τον στόχο. Συνήθως αυτό γίνεται «διορθώνοντας» το αρχικό σχέδιο, δηλαδή βρίσκοντας έναν τρόπο μετάβασης από την απροσδόκητη κατάσταση σε μία γνωστή που προβλέπεται από το αρχικό σχέδιο. Έτσι εξοικονομείται χρόνος αποφεύγοντας τη δημιουργία ενός νέου σχεδίου απ' την αρχή. Το σύστημα CYPRESS (Myers et al., 1994) για παράδειγμα ενσωματώνει ένα σύστημα δυναμικού σχεδιασμού με ένα σύστημα εκτέλεσης. Έχει δε την ικανότητα να αντιδρά σε απρόβλεπτες αλλαγές του περιβάλλοντος με επανασχεδιασμό (re-planning). Το σύστημα RETSINA (Paolucci et al., 1999) συνδυάζει ενέργειες σχεδιασμού και εκτέλεσης ενώ υποστηρίζει σχεδιασμό για δυναμικά και ρεαλιστικά περιβάλλοντα. Τέλος, το σύστημα Repair-SHOP (Nau et al., 2005; Warfield et al., 2007) παρέχει τη δυνατότητα προσαρμογής και διόρθωσης των σχεδίων.

5.1.3 Δένδρα Συμπεριφορών στην Τεχνολογία Συμπεριφορών

Μια άλλη μέθοδος που μπορεί να συμβάλει στην αντιμετώπιση της πολυπλοκότητας σε συστήματα λογισμικού είναι τα **Δένδρα Συμπεριφορών (Behavior Trees)**. Ένα δένδρο συμπεριφοράς είναι μια τυπική (formal), γραφική, δενδροειδής δομή που αναπαριστά τη συμπεριφορά μεμονωμένων μονάδων (components) ή ολόκληρων δικτύων από μονάδες ενός συστήματος λογισμικού. Οι μονάδες επηρεάζουν την κατάσταση του συστήματος, λαμβάνουν αποφάσεις, δημιουργούν ή ανταποκρίνονται σε γεγονότα και αλληλεπιδρούν ανταλλάσσοντας πληροφορίες ή περνώντας τον έλεγχο του συστήματος από τη μία στην άλλη.

Τα δένδρα συμπεριφορών χρησιμοποιούνται και στη Μηχανική Συμπεριφορών για τη διαχείριση της πολυπλοκότητας και την αύξηση της κατανόησης πολύπλοκων συστημάτων. Η Μηχανική Συμπεριφορών είναι ένα γνωστικό υπο-πεδίο της Τεχνολογίας Λογισμικού (software engineering) που ασχολείται με την ανάπτυξη συστημάτων λογισμικού μεγάλης κλίμακας και με πολλές αλληλεξαρτήσεις.

Τα δένδρα συμπεριφορών και ο τρόπος εφαρμογής τους σε συστήματα λογισμικού (και στην Τεχνολογία Λογισμικού εν γένει), προτάθηκαν αρχικά από τον Dromey (2006) ενώ κάποιες σχετικές ιδέες και έννοιες δημοσιεύτηκαν ήδη από το 2001 (Dromey, 2001). Τα δένδρα συμπεριφορών χρησιμοποιούνται για την τυπική περιγραφή (formal representation) των επί μέρους συμπεριφορών κάθε απαίτησης από ένα σύστημα (system requirement). Γενικά, η συμπεριφορά σε ένα μεγάλης κλίμακας σύστημα (με παραλληλισμό εργασιών) αναπαρίσταται ως ένα σύνολο από σειριακές διεργασίες (threads) που επικοινωνούν μεταξύ τους. Η σημειογραφία που χρησιμοποιείται καταγράφει τις σύνθετες καταστάσεις των μερών του συστήματος σε μια απλή δενδροειδή μορφή (Behavior Tree Group, 2007).

Οι συμπεριφορές εκφράζονται ως μονάδες συστημάτων (components) που υλοποιούν καταστάσεις και μονάδες που δημιουργούν και καταργούν σχέσεις. Χρησιμοποιώντας τη λογική, τις σημειογραφικές και γραφικές «συμβάσεις» από διάφορες γλώσσες προγραμματισμού, οι μονάδες μπορεί να υποστηρίξουν την εκτέλεση ενεργειών, αποστολή/λήψη γεγονότων, έλεγχο των ροών εκτέλεσης και των ροών δεδομένων του συστήματος, καθώς και των διεργασιών αυτού (Dromey, 2003).

Με τα δένδρα συμπεριφορών μπορεί να καταγραφούν και εκφραστούν με ακρίβεια οι λειτουργικές απαιτήσεις (functional requirements), οι οποίες συνήθως εκφράζονται με χρήση της φυσικής γλώσσας. Ενώ χρησιμοποιείται το λεξιλόγιο και η ορολογία της απαίτησης (σε φυσική γλώσσα), τα δένδρων συμπεριφορών με χρήση κατάλληλων γραφικών δομών εξαλείφουν τυχόν ασάφειες. Κατ' αυτόν τον τρόπο παρέχουν μια ξεκάθαρη σχέση μεταξύ αυτών που εκφράζονται σε φυσική γλώσσα και των τυπικών προδιαγραφών στις οποίες αντιστοιχούν (Dromey, 2005). Η βάση της σχετικής σημειογραφίας είναι το ότι μια συμπεριφορά πάντοτε σχετίζεται (ή αφορά) μία μονάδα του συστήματος. Οι καταστάσεις των μονάδων αναπαριστώνται ως κόμβοι των δένδρων συμπεριφορών και συντίθενται, είτε σειριακά είτε παράλληλα, για να σχηματίζουν ένα δένδρο που αναπαριστά την απαιτούμενη συμπεριφορά, σύμφωνα με μία λειτουργική απαίτηση που άλλως εκφράζεται με χρήση φυσικής γλώσσας.

Ένα δένδρο συμπεριφοράς προδιαγράφει τις μεταβολές της κατάστασης μιας μονάδας, τον τρόπο μεταφοράς δεδομένων και ελέγχου μεταξύ των μονάδων καθώς και το πως αλληλεπιδρούν οι διεργασίες του συστήματος. Έχουν προβλεφθεί στη σημειογραφία δομές για τη δημιουργία ή κατάργηση σχέσεων μεταξύ των μονάδων, την αλλαγή ή έλεγχο της κατάστασής τους καθώς και μηχανισμοί για επικοινωνία μεταξύ διεργασιών, συμπεριλαμβανομένων της αποστολής μηνυμάτων ή γεγονότων (message or event passing), του συγχρονισμού (synchronization) των διεργασιών και της προστασίας των μοιραζόμενων μεταβλητών (shared variable blocking). Η τυπική σημασιολογία (formal semantics) των

δένδρων συμπεριφορών ορίζεται με χρήση μιας άλγεβρας διεργασιών (process algebra) καθώς και της αντίστοιχης λειτουργικής σημασιολογίας (operational semantics) (Colvin et al., 2008; Colvin and Hayes, 2006).

5.1.4 Δένδρα Συμπεριφορών στην Τεχνητή Νοημοσύνη παιχνιδιών Η/Υ

Η μοντελοποίηση με δένδρα συμπεριφορών έχει χρησιμοποιηθεί σε μια ευρεία γκάμα από εφαρμογές. Μερικές από αυτές είναι τα συστήματα μεγάλης κλίμακας (large-scale systems), ενσωματωμένα συστήματα (embedded systems), έλεγχος πρόσβασης βασισμένος σε ρόλους (role-based access control), μοντελοποίηση βιολογικών συστημάτων και μοντελοποίηση της τεχνητής νοημοσύνης παιχνιδιών για Η/Υ. Τα δένδρα συμπεριφορών χρησιμοποιούνται ευρύτατα από τους προγραμματιστές παιχνιδιών Η/Υ (για παράδειγμα Palma et al., 2011; Lee et al., 2011; Mehta and Ram, 2009) για την υλοποίηση αλληλεπιδραστικών μεθόδων λήψης αποφάσεων και ελέγχου «εικονικών πλασμάτων ή οντοτήτων» που υπάρχουν στα παιχνίδια Η/Υ, όπως για παράδειγμα στο DEFCON (Lim et al., 2010) ή στον διαγωνισμό Mario AI Competition (Perez et al., 2011). Αυτά τα δένδρα συμπεριφορών είναι πιο εξειδικευμένα και διαφέρουν από εκείνα άλλων εφαρμογών. Συγκεκριμένα προσομοιάζουν περισσότερο στις ιεραρχικές μηχανές πεπερασμένων καταστάσεων (Hierarchical Finite State Machines) ή στα ιεραρχικά δίκτυα εργασιών (hierarchical task networks) που έχουμε ήδη παρουσιάσει.

Ορθότερα, τα δένδρα συμπεριφορών στα παιχνίδια Η/Υ πρόκειται για κατευθυντικούς ακυκλικούς γράφους (Directed Acyclic Graphs ή DAGs). Ένας κόμβος μπορεί να έχει πολλούς γονείς γεγονός που επιτρέπει την επαναχρησιμοποίηση υπο-δένδρων συμπεριφορών. Τα δένδρα συμπεριφορών αντικαθιστούν την πολυπλοκότητα των μεταβάσεων κατάστασης των μηχανών πεπερασμένων καταστάσεων (finite state machines ή FSMs) με μια πιο αυστηρή αλλά και πιο δομημένη προσέγγιση. Δημιουργούνται δε με την ιεραρχική οργάνωση προϋπαρχόντων υπο-δένδρων συμπεριφορών, τα οποία γίνονται οι κόμβοι των νέων δένδρων. Η διάσχιση (ή επίσκεψη ή ενεργοποίηση) ενός κόμβου ενός δένδρου συμπεριφοράς ισοδυναμεί με εκτέλεση κάποιων λειτουργιών που προσδιορίζονται από τη σημασιολογία (semantics) του κόμβου. Αν ο προς εκτέλεση κόμβος είναι ο γονέας ενός υπο-δένδρου τότε εκτελούνται αντίστοιχα και οι κόμβοι του υπο-δένδρου. Η εκτέλεση ενός κόμβου ή ενός υπο-δένδρου επιστρέφει ένα αποτέλεσμα που εκφράζει την (συνολική) του κατάσταση. Για παράδειγμα, «ο κόμβος είναι έτοιμος για εκτέλεση», «εκτελέστηκε με επιτυχία», «η εκτέλεση δεν ολοκληρώθηκε και πρέπει δοθεί επιπλέον χρόνος, στον επόμενο κύκλο», «αποτυχία εκτέλεσης χωρίς παρενέργειες (side-effects)», «σφάλμα με παρενέργειες (side-effects) που χρίζουν αντιμετώπισης».

Τα δένδρα συμπεριφορών εκτελούνται χρησιμοποιώντας τη μέθοδο της διάσχισης κατά βάθος (depth-first search) ξεκινώντας από την ρίζα τους. Όμως κάθε κόμβος που εκτελείται μπορεί να επηρεάσει την περαιτέρω διάσχιση. Για παράδειγμα, οι κόμβοι-επιλογείς (selector nodes) επιλέγουν για διάσχιση έναν από τους κόμβους-παιδιά τους (αν υπάρχουν), κι ανάλογα με το αποτέλεσμα της εκτέλεσης αποφασίζουν για την εκτέλεση επιπλέον κόμβων-παιδιών ή όχι. Επιπλέον παραδείγματα κόμβων που μεταβάλλουν τον τρόπο διάσχισης είναι:

- Οι σειριακοί κόμβοι (Sequence nodes) εκτελούν διαδοχικά όλους τους κόμβους-παιδιά τους. Αν όμως έστω και ένα από αυτά αποτύχει τότε η εκτέλεση ολόκληρου του σειριακού κόμβου θεωρείται αποτυχημένη. Όταν ολοκληρωθεί με επιτυχία η εκτέλεση και του τελευταίου κόμβου-παιδιού τότε θεωρείται ότι ολοκληρώνεται με επιτυχία και η εκτέλεση του σειριακού κόμβου.
- Οι κόμβοι-επιλογείς (Selector nodes) εκτελούν τους κόμβους-παιδιά τους σύμφωνα με μια σειρά προτεραιότητας μέχρις ότου κάποιο από αυτά εκτελεστεί με επιτυχία. Υπάρχουν διάφορες παραλλαγές που αφορούν τις λεπτομέρειες επιλογής και εκτέλεσης των παιδιών σε κάθε επίσκεψη του κόμβου-επιλογέα.
- Οι κόμβοι-βρόχοι (Loop nodes) είναι όπως οι σειριακοί κόμβοι αλλά μετά την ολοκλήρωση και του τελευταίου κόμβου-παιδιού αρχίζουν έναν νέο κύκλο εκτέλεσης αντί να επιστρέψουν την διάσχιση στον κόμβο-γονέα τους.
- Οι κόμβοι-τυχαίας επιλογής (Random selector nodes) επιλέγουν τυχαία έναν από τους κόμβους-παιδιά τους για εκτέλεση.
- Οι κόμβοι παραλληλισμού (Concurrent nodes) εκτελούν ταυτόχρονα όλα τα παιδιά τους. Συνήθως αν ένας προκαθορισμένος αριθμός κόμβων-παιδιών αποτύχει τότε αποτυγχάνει και η εκτέλεση του κόμβου παραλληλισμού. Ο κόμβος αυτός μπορεί να συνδυαστεί και με έλεγχο συνθηκών εκτέλεσης των κόμβων-παιδιών του έτσι ώστε να επιλεγεί ποιοι από αυτούς θα εκτελεστούν και ποιοι όχι.
- Οι κόμβοι Decorators (Decorator nodes) έχουν τυπικά μόνο έναν κόμβο-παιδί και χρησιμοποιούνται είτε για να καθορίσουν το αποτέλεσμα της εκτέλεσης, είτε για τη δημιουργία χρονομέτρων (timers) που ελέγχουν την διάρκεια ή τη συχνότητα εκτέλεσης του κόμβου-παιδιού.

Οι τερματικοί κόμβοι (φύλλα του δένδρου) διακρίνονται (α) σε «ενέργειες» (actions) που υλοποιούν τις αλλαγές κατάστασης του παιχνιδιού ή ενός συγκεκριμένου «παίκτη» ή μίας «οντότητας» του παιχνιδιού, και (β) σε «συνθήκες» (conditions) που ελέγχουν αν η κατάσταση του παιχνιδιού ή ενός παίκτη ή οντότητας έχει κάποια συγκεκριμένη τιμή (ή τιμές). Αν ένας σειριακός κόμβος έχει ως παιδί του έναν κόμβο συνθήκης και αποτύχει η εκτέλεσή του (δηλαδή δεν ισχύει η συνθήκη τη στιγμή της εκτέλεσης του κόμβου), τότε δεν θα εκτελεστούν οι κόμβοι-παιδιά που έπονται του κόμβου συνθήκης. Αν μια συνθήκη χρησιμοποιηθεί ως παιδί σε έναν κόμβο παραλληλισμού τότε λειτουργεί ως έλεγχος εκτέλεσης των επόμενων κόμβων-παιδιών.

Το κύριο όφελος που κομίζουν τα δένδρα συμπεριφορών στην μοντελοποίηση της τεχνητής νοημοσύνης παιχνιδιών για Η/Υ είναι το ξεκάθαρο «λεξιλόγιο» (vocabulary) που χρησιμοποιούν (σε αντίθεση με τις FSMs) που λειτουργεί και ως οδηγός για τον τρόπο σκέψης του αναλυτή-προγραμματιστή. Επιπλέον, οι ενέργειες, οι συνθήκες ή και ολόκληρα υπο-δένδρα μπορεί να εμφανίζονται σε πολλαπλά σημεία ενός άλλου δένδρου συμπεριφοράς, επιτρέποντας έτσι την ανάπτυξη συνδεόμενων επεκτάσεων (plugins) και επαναχρησιμοποιήσιμων δομικών μονάδων (reusable building blocks). Οι Mateas and Stern (2002) και Skorupski and Mateas (2010) ανέπτυξαν μια αρχιτεκτονική που βασίζεται στα δένδρα συμπεριφορών, μια αντίστοιχη γλώσσα περιγραφής χαρακτήρων (character authoring language) καθώς και έναν διαχειριστή δράματος (drama manager) προκειμένου

να υλοποιήσουν ένα αλληλεπιδραστικό δράμα (interactive drama) για τις ανθρώπινες σχέσεις.

Οι παραπάνω τεχνολογίες και προσεγγίσεις αποτέλεσαν τη βάση για την ανάπτυξη της πρώτης καινοτομίας της διατριβής (δηλαδή των SANs), η οποία αξιοποίησε ιδέες από αυτές έτσι ώστε να απαντήσει στο πρώτο ερευνητικό ερώτημα που τέθηκε στο κεφάλαιο 4. Ειδικότερα, επιλέχθηκαν τα δένδρα συμπεριφορών στην τεχνητή νοημοσύνη παιχνιδιών Η/Υ και τα ιεραρχικά δίκτυα εργασιών, ως η βάση για τα SANs, τα οποία τις επεκτείνουν με δυνατότητες επίγνωσης κατάστασης, προηγμένη χρήση του πλαισίου εκτέλεσης (context) και χρήση γεγονότων. Τα SANs παρουσιάζονται αναλυτικά στη συνέχεια.

5.2 Παρουσίαση των Δικτύων Κατάστασης – Ενέργειας

Όπως προαναφέρθηκε, στην παρούσα διατριβή έχουμε εστιάσει το ενδιαφέρον μας σε διαδικασίες και συστήματα οδηγούμενα από γεγονότα (event-driven systems). Στο πλαίσιο αυτό θεωρούμε ότι οι όποιες πληροφορίες ανταλλάσσονται μεταξύ μιας διαδικασίας και του περιβάλλοντός της γίνεται με τη μορφή γεγονότων (events). Όταν προέρχονται από το περιβάλλον, τα γεγονότα περιέχουν τις τιμές κάποιων (μίας ή μερικών) παραμέτρων του περιβάλλοντος. Όταν προέρχονται μέσα από τη διαδικασία τότε περιέχουν πληροφορίες για την κατάστασή της (process state) ή για τα αποτελέσματα της λειτουργίας της (process output). Με κατάλληλο συνδυασμό διαφόρων τύπων γεγονότων του περιβάλλοντος είναι δυνατό να «συντεθεί» μια εικόνα της διαμόρφωσης του περιβάλλοντος και των καταστάσεων που επικρατούν. Η εικόνα αυτή μπορεί να αφορά είτε την τρέχουσα είτε κάποια παρελθούσα διαμόρφωση του περιβάλλοντος, αναλόγως με το αν το γεγονός που χρησιμοποιήθηκαν είναι πρόσφατα ή παλαιότερης περιόδου. Για τη σύνθεση της συνολικής εικόνας του περιβάλλοντος αλλά και για την ανίχνευση συγκεκριμένων καταστάσεων ενδιαφέροντος χρησιμοποιούνται τεχνικές Επεξεργασίας Σύνθετων Γεγονότων (Complex Event Processing). Το λογισμικό που χρησιμοποιείται για αυτόν το σκοπό αναφέρεται ως CEP Engine. Τα SANs στηρίζουν εν μέρει τη λειτουργία τους στη χρήση CEP engines για την αναγνώριση καταστάσεων ενδιαφέροντος.

5.2.1 Εισαγωγή – Σκοπός

Σκοπός των SANs είναι να παρέχουν έναν εύχρηστο και κατανοητό τρόπο μοντελοποίησης των συμπεριφορών συστημάτων και διαδικασιών. Ο όρος «συμπεριφορά» (behavior), ανάλογα με το επιστημονικό πεδίο που την εξετάζει, μπορεί να ορίζεται με διαφορετικό τρόπο. Συνήθως με τον όρο αυτό εννοούμε ένα σύνολο ενεργειών και ιδιομορφιών που εμφανίζει ένα σύστημα ή οργανισμός σε σχέση με το περιβάλλον στο οποίο βρίσκεται. Αποτελεί δε την αντίδραση του συστήματος ή οργανισμού στα διάφορα ερεθίσματα του περιβάλλοντος. Στην επιστήμη των υπολογιστών η συμπεριφορά ενός συστήματος κωδικοποιείται συνήθως ως επαναχρησιμοποιήσιμες μονάδες κώδικα που του επιτρέπουν να ανταποκρίνεται στις αλλαγές του περιβάλλοντος και σε εσωτερικά αίτια, με τρόπους που έχουν νόημα και παράγουν κάποιο αποτέλεσμα.

Σύμφωνα με τη δική μας οπτική, κάθε διαδικασία ή σύστημα ή μονάδα αυτών εμφανίζει συγκεκριμένες «συμπεριφορές» (λειτουργίες, αντιδράσεις) κάτω από διαφορετικές συνθήκες του περιβάλλοντος. Αυτές οι συμπεριφορές αποβλέπουν στην επίτευξη κάποιων στόχων. Προφανώς ανάλογα με τις συνθήκες (κατάσταση) που επικρατούν είναι δυνατό οι στόχοι του συστήματος ή διαδικασίας να επιτυγχάνονται από διαφορετικές συμπεριφορές. Δηλαδή ένα σύστημα ή διαδικασία έχει σκοπούς που επιδιώκει να επιτύχει επιδεικνύοντας συγκεκριμένες συμπεριφορές που ανάλογα με την κατάσταση του περιβάλλοντος αυτές ενδέχεται να αλλάζουν.

Σκοπός λοιπόν των SANs είναι να εκφράσουν αυτές τις συμπεριφορές σε σχέση με τους στόχους που επιδιώκουν αλλά και τις καταστάσεις που πρέπει να συντρέχουν προκειμένου αυτές οι συμπεριφορές να ενεργοποιηθούν. Επιπλέον, τα SANs αποβλέπουν και στην καταγραφή της ανάλυσης των συμπεριφορών σε απλούστερες καθώς και στις στοιχειώδεις λειτουργίες που τις συνθέτουν. Τα SANs δεν καταγράφουν τη δομή ενός συστήματος, ούτε τη λειτουργία του από καθαρά τεχνικής απόψεως. Αντιθέτως καταγράφουν τους στόχους του και τις συμπεριφορές που οδηγούν στην επίτευξη αυτών σε συνάρτηση με την εκάστοτε κατάσταση που επικρατεί.



Σχήμα 5-1. Συσχέτιση στόχων-συμπεριφορών και καταστάσεων

5.2.2 Ορισμός των Δικτύων Κατάστασης – Ενέργειας

Τα SANs είναι κατευθυντικοί ακυκλικοί γράφοι (Directed Acyclic Graphs, DAGs) των οποίων οι κόμβοι έχουν ειδικές λειτουργίες και σημασιολογία, και χρησιμοποιούνται για την ανάλυση και μοντελοποίηση των στόχων συστημάτων (goal decompositions). Δίνουν δε μια εικόνα του πως οι στόχοι ενός συστήματος αναλύονται **ιεραρχικά** σε υποστόχους, υποστόχους των υποστόχων και ούτω καθεξής μέχρις ότου αναλυθούν όλοι σε απλές ενέργειες που δεν επιδέχονται περαιτέρω ανάλυσης (δείτε Σχήμα 5-2 και Σχήμα 5-3). Σε κάθε επίπεδο του γράφου οι στόχοι του προηγούμενου επιπέδου εμφανίζονται αναλυμένοι στους υποστόχους τους ή σε απλές ενέργειες. Η εκτέλεση των απλών ενεργειών οδηγεί στην επίτευξη των (υπο)στόχων του προηγούμενου επιπέδου. Ομοίως η επίτευξη των υποστόχων ενός επιπέδου οδηγεί στην επίτευξη των στόχων του αμέσως ανώτερου επιπέδου. Έτσι τα SANs παρέχουν το σχέδιο εκτέλεσης (execution plan) για την επίτευξη των στόχων. Ουσιαστικά το σχέδιο είναι η σειρά εκτέλεσης όλων των απλών ενεργειών ενός SAN καθώς και ο τρόπος που αυτές πρέπει να συνδυαστούν προκειμένου να επιτευχθεί ο στόχος του ανώτατου επιπέδου.

Ως κόμβοι ενός γράφου, οι στόχοι επιτρέπεται να έχουν περισσότερους από έναν γονείς. Αυτό σημαίνει ότι ένας υποστόχος μπορεί να αποτελεί μέρος της ανάλυσης δύο ή

περισσότερων διαφορετικών στόχων υψηλότερου επιπέδου, αρκεί να μη δημιουργούνται κυκλικές εξαρτήσεις, δηλαδή ο υποστόχος να περιλαμβάνει κάποιον στόχο υψηλότερου επιπέδου ως δικό του υποστόχο. Όλοι οι στόχοι θα πρέπει να έχουν τουλάχιστο έναν στόχο-γονέα εκτός φυσικά από τον στόχο του πρώτου επιπέδου που είναι και η ρίζα του γράφου. Τα SANs έχουν ακριβώς έναν στόχο-ρίζα (Root Goal).

Οι στόχοι διακρίνονται σε δύο βασικές κατηγορίες, αυτούς που αποβλέπουν στην επίτευξη κάποιου επιθυμητού αποτελέσματος και σε αυτούς που αποβλέπουν στην αποφυγή κάποιου ανεπιθύμητου αποτελέσματος. Αναφέρονται δε ως στόχοι “Επίτευξης” (Achieve) και στόχοι “Αποφυγής” (Avoid) αντίστοιχα. Η πρώτη κατηγορία περιλαμβάνει στόχους που επιδιώκουν να διατηρούν κάποιες παραμέτρους του συστήματος ή του περιβάλλοντος σε κάποια επιθυμητά εύρη τιμών (ή κατάσταση) και ενεργοποιούν το σχέδιο επίτευξής τους όταν οι τιμές αποκλίνουν από τα αναμενόμενα εύρη. Αντίστροφα, η δεύτερη κατηγορία περιλαμβάνει στόχους που επιδιώκουν οι παράμετροι του συστήματος/διαδικασίας ή του περιβάλλοντος να **μην** λαμβάνουν κάποια εύρη τιμών και αντιδρούν όταν συμβεί κάτι τέτοιο. Ένας στόχος επίτευξης είναι δυνατό να αναδιατυπωθεί ως στόχος αποφυγής και αντίστροφα. Σε ορισμένες εργασίες (Nickols, 2003) ορίζονται ακόμη και οι κατηγορίες των στόχων “Διατήρησης” (Preserve), για διατήρηση μιας κατάστασης που ήδη υφίσταται, και “Εξάλειψης” (Eliminate), για αλλαγή μιας υφιστάμενης ανεπιθύμητης κατάστασης. Ωστόσο εμείς θεωρούμε ότι αυτές καλύπτονται από τους στόχους επίτευξης και αποφυγής αντίστοιχα.

Στα SANs κάθε στόχος του συστήματος χαρακτηρίζεται από τρεις παραμέτρους. Πρώτα από την κατάσταση του περιβάλλοντος που ενεργοποιεί το σχέδιο επίτευξής του, την οποία συνάγει από τα γεγονότα που θα ληφθούν από το περιβάλλον³⁷. Κατόπιν, από μία συνθήκη ελέγχου που χρησιμοποιείται για να προσθέσει επιπλέον δυνατότητες έκφρασης και ελέγχου στα SANs. Τέλος από την ενέργεια που υλοποιεί τη συμπεριφορά που θα οδηγήσει στην επίτευξη του στόχου.

Σημειώνουμε ότι δεν είναι απαραίτητο να επιδιώκονται όλοι οι στόχοι μιας διαδικασίας ή ενός συστήματος ταυτόχρονα, ιδίως αν αυτό είναι πολυσύνθετο. Εξάλλου κάποιος από αυτούς μπορεί να έχουν νόημα μόνο σε συγκεκριμένες περιστάσεις. Συνεπώς οι στόχοι μιας διαδικασίας ή ενός συστήματος μπορούν να ενεργοποιούνται και απενεργοποιούνται κατά τη λειτουργία του. Όταν ενεργοποιούνται τότε εξετάζεται αν συντρέχουν οι συνθήκες για να ξεκινήσει το σχέδιο επιδίωξής τους (συμπεριφορά). Αν δεν συντρέχουν οι συνθήκες τότε αναμένεται αυτές να μεταβληθούν κατάλληλα προκειμένου να ξεκινήσει η επιδίωξη των στόχων.

Το απλούστερο δυνατό SAN είναι ένα δένδρο δύο επιπέδων που αποτελείται από τον κόμβο-ρίζα και τρεις κόμβους-παιδιά, όπου ο καθένας έχει τον δικό του ρόλο (Σχήμα 5-2). Ο κόμβος-ρίζα μοντελοποιεί τον προς επίτευξη στόχο και ονομάζεται Κόμβος Στόχου.

³⁷ Για την ακρίβεια συνάγει την κατάσταση από ένα σύνθετο γεγονός (complex event) που λαμβάνει από έναν CEP engine, ο οποίος την ανιχνεύει επεξεργαζόμενος γεγονότα του περιβάλλοντος.

- Ο αριστερός κόμβος-παιδί περιγράφει την κατάσταση που θα πρέπει να συντρέχει στο περιβάλλον του συστήματος ούτως ώστε να ενεργοποιηθεί το σχέδιο επίτευξης του στόχου. Ονομάζεται δε Κόμβος Κατάστασης (Situation node). Όταν ενεργοποιείται ένας στόχος, πρώτα αναμένει να συμβεί η κατάσταση που περιγράφεται στον αντίστοιχο κόμβο κατάστασης (αν δεν συμβαίνει ήδη) και μόνο τότε συνεχίζει στην εκτέλεση των δύο επόμενων κόμβων.

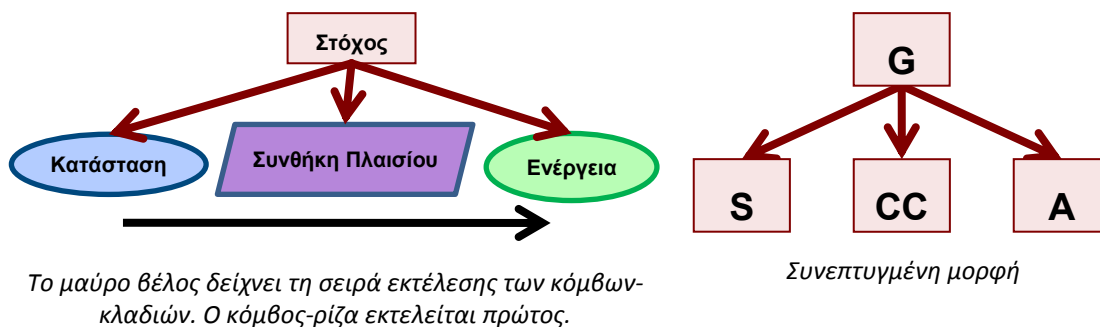
Ως κατάσταση (situation) θεωρούμε ένα σύνολο συνθηκών (ή τιμών κάποιων παραμέτρων) του περιβάλλοντος, που ισχύουν σε ορισμένο χωρικό και χρονικό πλαίσιο, και αντιστοιχούν σε μια κοινή ερμηνεία των τεκταινομένων μέσα σε αυτό. Μια κατάσταση είναι διακριτή από τις άλλες και αφορά συνήθως ένα υποσύνολο των παραμέτρων του περιβάλλοντος. Συνεπώς μπορεί να συντρέχουν περισσότερες από μία καταστάσεις ταυτόχρονα. Επίσης, η κατάσταση ως συνδυασμός τιμών των παραμέτρων, έχει μια διάρκεια ζωής και άρα αρχή και τέλος.

Με την ενεργοποίηση του στόχου, ο κόμβος κατάστασης προβαίνει σε μια σειρά από ενέργειες προκειμένου να διασφαλίσει ότι θα λάβει την απαραίτητη ενημέρωση για την ύπαρξη ή τη μετάβαση στην αναμενόμενη κατάσταση. Σε συστήματα οδηγούμενα από γεγονότα (event-driven systems) απαιτείται η χρήση ενός κατάλληλου προτύπου σύνθετου γεγονότος (Complex Event Pattern ή CEPat) προκειμένου να ανιχνευθεί η αναμενόμενη κατάσταση από μια κατάλληλη αλληλουχία γεγονότων του περιβάλλοντος. Το πρότυπο σύνθετου γεγονότος ορίζεται στον κόμβο κατάστασης, ο οποίος κατά την ενεργοποίηση του στόχου το αναθέτει για εκτέλεση σε έναν κατάλληλο μηχανισμό επεξεργασία σύνθετων γεγονότων (CEP engine). Ο CEP engine αναλύει όλα τα γεγονότα που λαμβάνονται από το περιβάλλον κι αν κάποια αλληλουχία τους ταιριάζει με το πρότυπο τότε δημιουργεί ένα «σύνθετο γεγονός» το οποίο σηματοδοτεί την έναρξη ισχύος της αναμενόμενης κατάστασης. Κατόπιν, το γεγονός αυτό αποστέλλεται στον κόμβο κατάστασης για να συνεχίσει τη λειτουργία του. Περισσότερα γι' αυτόν τον κόμβο θα αναφερθούν στη συνέχεια.

- Ο μεσαίος κόμβος-παιδί απαιτεί να ισχύει μια συγκεκριμένη συνθήκη στα δεδομένα του πλαισίου εκτέλεσης (execution context) προτού ξεκινήσει η προσπάθεια επίτευξης του στόχου. Εκτελείται αφού έχει ανιχνευθεί ότι συντρέχει η κατάσταση του αριστερού κόμβου. Ο κόμβος αυτός καλείται Κόμβος Συνθήκης Πλαισίου (Context Condition node) και ουσιαστικά αποτιμά μια λογική έκφραση σε αληθή ή ψευδή (true/false) προκειμένου να ελέγξει αν πρέπει να προχωρήσει στην επίτευξη του στόχου ή όχι. Το πλαίσιο εκτέλεσης ως έννοια αφορά κυρίως την εσωτερική κατάσταση του SAN (δείτε ενότητα 5.2.4).
- Ο δεξιός κόμβος-παιδί ορίζει την ενέργεια που πρέπει να εκτελεστεί προκειμένου να επιτευχθεί ο στόχος και ονομάζεται Κόμβος Ενέργειας (Action node). Μπορεί επίσης να είναι ένας υποστόχος (Κόμβος Υποστόχου ή Subgoal node), ή επίσης μπορεί να είναι μια σύνθετη ενέργεια (Κόμβος Σύνθετης Ενέργειας ή Composite Action node) που συνδυάζει πολλούς υποστόχους μαζί, οι οποίοι από κοινού συνθέτουν τη συμπεριφορά του στόχου. Φυσικά οι υποστόχοι χαρακτηρίζονται από

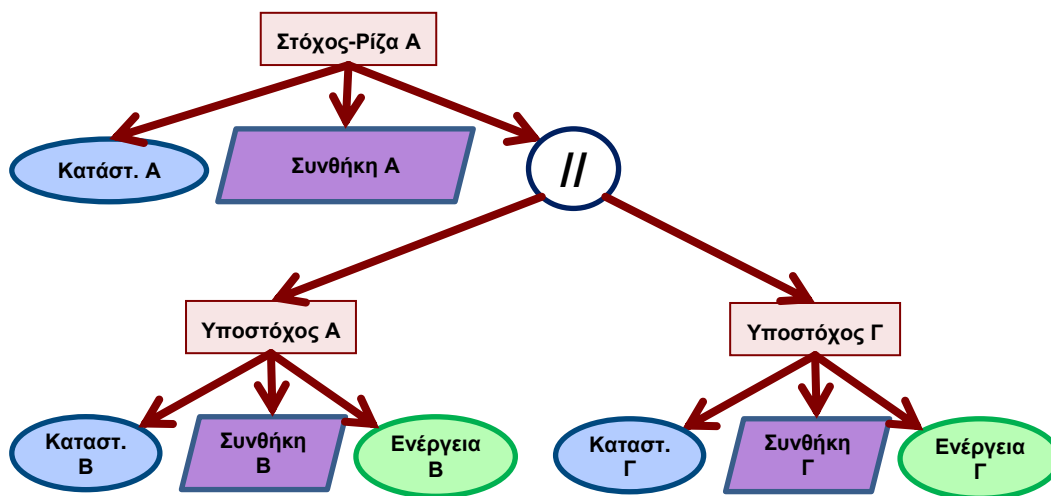
τις ίδιες τρεις παραμέτρους με τους στόχους, οπότε όταν χρειαστεί να εκτελεστεί η ενέργεια (υλοποιηθεί η συμπεριφορά) του στόχου-γονέα τότε ενεργοποιούνται οι υποστόχοι και οι δικοί τους κόμβοι κατάστασης, συνθήκης πλαίσιου και ενέργειας.

Σημειώνουμε ότι ο στόχος που βρίσκεται στο ανώτατο επίπεδο της ιεραρχίας στόχων ενός SAN, ονομάζεται **στόχος-ρίζα** ή **Root Goal** και αποτελεί την αρχή (ρίζα) του γράφου του SAN. Εκφράζει δε έναν κύριο στόχο του συστήματος. Κάθε SAN έχει ακριβώς έναν στόχο-ρίζα. Ακολούθως δίνεται μια γραφική αναπαράσταση της βασικής δομής ενός απλού SAN.



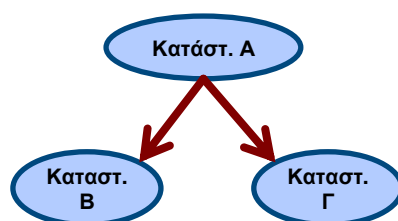
Σχήμα 5-2. Βασική δομή ενός SAN

Ο δεξιός κλάδος (A) ενός κόμβου στόχου (G), είτε πρόκειται για απλή ενέργεια είτε για υποδίκτυο υποστόχων, εκφράζει τη συμπεριφορά του συστήματος που απαιτείται για την επίτευξη του στόχου (G), όταν συντρέχει η κατάσταση περιβάλλοντος που κωδικοποιείται από τον κόμβο κατάστασης (S) και όταν η συνθήκη του κόμβου συνθήκης πλαίσιου (CC) αποτιμάται ως αληθής (δηλαδή επιτρέπει την εκτέλεση του κόμβου ενέργειας A). Αν ο δεξιός κλάδος πρόκειται για δίκτυο υποστόχων τότε η συμπεριφορά αυτή αναλύεται περαιτέρω σε επιμέρους συμπεριφορές σχετικές με τους αντίστοιχους υποστόχους. Όταν αυτές συνδυαστούν κατάλληλα έχουν ως αποτέλεσμα τη συμπεριφορά του αρχικού κόμβου στόχου. Σημειώνουμε ότι οι κόμβοι απλών ενεργειών είναι τερματικοί κόμβοι του δικτύου (leaf nodes) και δεν έχουν κάποια σχετιζόμενη κατάσταση ή συνθήκη ελέγχου.



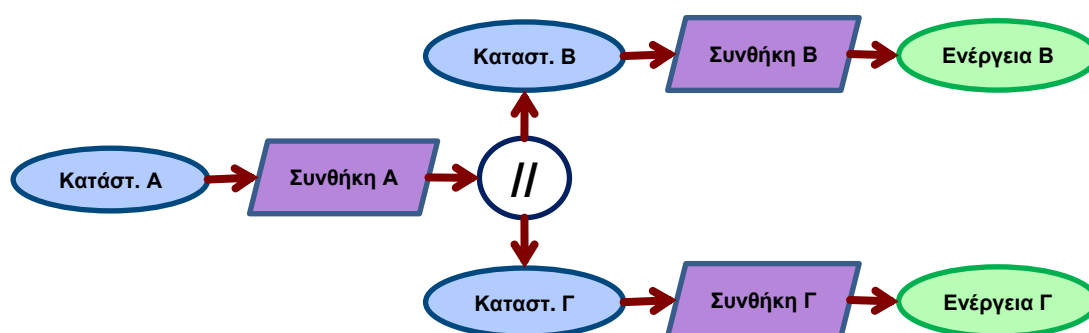
Σχήμα 5-3. SAN δύο επιπέδων με ανάλυση σε υποστόχους

Επισημαίνουμε ακόμη ότι η κατάσταση ενός στόχου ελέγχεται μόνο όταν αυτός ενεργοποιείται. Φυσικά μετά την επίτευξη του στόχου παύει να ελέγχεται η κατάσταση αυτή. Όταν ένας στόχος αναλύεται σε υποστόχους τότε απαιτείται να συντρέχουν ταυτόχρονα οι αντίστοιχες καταστάσεις τόσο του στόχου όσο και των υποστόχων του. Κατ' αυτόν τον τρόπο δημιουργείται και μία ιεραρχία καταστάσεων που πρέπει να συμβαίνουν ταυτόχρονα, παράλληλα με την ιεραρχία των στόχων. Οι καταστάσεις των υποστόχων εξαρτώνται και μπορούν να ελεγχθούν (δηλαδή να μας ενδιαφέρουν) μόνο όταν συντρέχουν οι καταστάσεις όλων των στόχων υψηλότερου επιπέδου. Ακολούθως φαίνεται η ιεραρχία των καταστάσεων του παραδείγματος στο Σχήμα 5-3 που εκφράζει ότι πρέπει να έχει συμβεί η κατάσταση Α (του στόχου-ρίζα) προκειμένου να μας ενδιαφέρουν οι καταστάσεις Β και Γ των υποστόχων.



Σχήμα 5-4. Ιεραρχία καταστάσεων του SAN στο Σχήμα 5-3

Κλείνοντας την εισαγωγή στα SANs δίνουμε ένα παράδειγμα του σχεδίου εκτέλεσης του παραδείγματος στο Σχήμα 5-3. Περιλαμβάνει μόνο τερματικούς κόμβους.



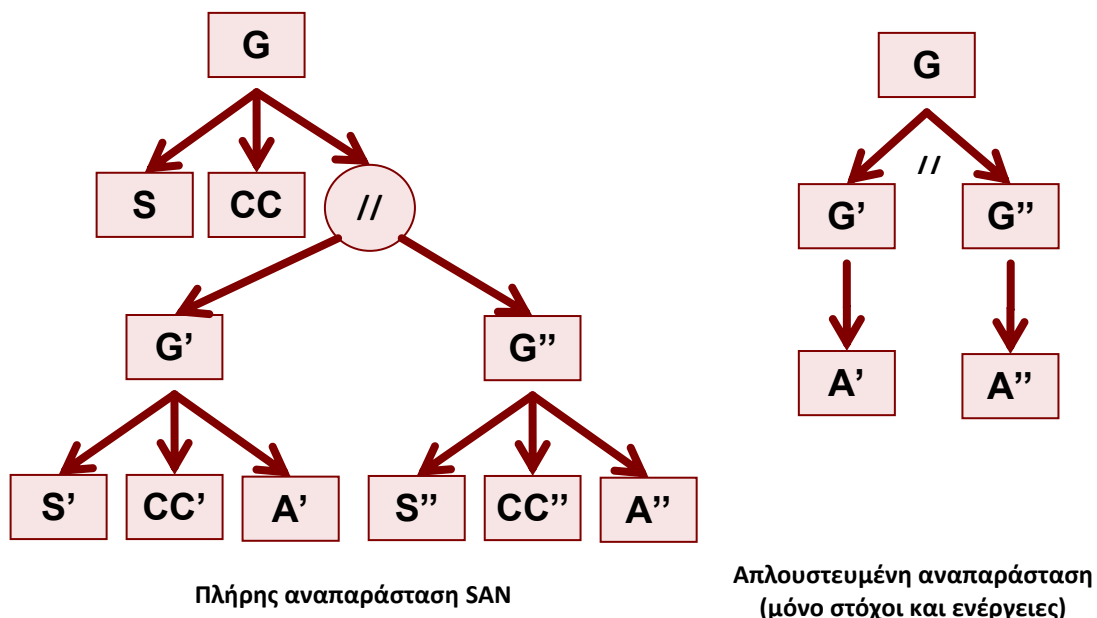
Σχήμα 5-5. Σχέδιο εκτέλεσης του SAN στο Σχήμα 5-3, ως ροή εργασιών

5.2.3 Τρόποι αναπαράστασης

Τα SANs ως δομή δεδομένων μπορεί να αναπαρασταθεί με διάφορες σημειογραφίες, είτε συμβολικές (symbolic notations) είτε γραφικές (graphical notations). Είναι δυνατό να χρησιμοποιηθούν οι ίδιες που χρησιμοποιούνται για τους κατευθυντικούς γράφους (directed graphs) με κατάλληλες επισημάνσεις (annotations) που αφορούν τα ιδιαίτερα σημασιολογικά χαρακτηριστικά των κόμβων των SANs. Στην παρούσα εργασία υιοθετούμε δύο (παρεμφερείς) γραφικούς τρόπους αναπαράστασης που προέρχονται από τη συνήθη σημειογραφία των δένδρων συμπεριφορών. Ο πρώτος είναι πιο αναλυτικός και στηρίζεται στη χρήση κατάλληλων σχημάτων και χρωμάτων για να επισημάνει το είδος του κάθε κόμβου, ενώ ο δεύτερος χρησιμοποιεί το ίδιο σχήμα για όλους τους κόμβους, οπότε το

είδος του θα πρέπει να επισημαίνεται με αντίστοιχη λεκτική ένδειξη (π.χ. G για κόμβο στόχου ή υποστόχου, S για κόμβο κατάστασης, C ή CC για κόμβο συνθήκης και A για κόμβο ενέργειας). Ο δεύτερος τρόπος παράγει πιο συνεπτυγμένα σχήματα απ' ό,τι ο πρώτος. Παραδείγματα των δύο αυτών αναπαραστάσεων δείτε στο Σχήμα 5-2.

Πολλές φορές κατά την μοντελοποίηση των στόχων με SANs, εξυπηρετεί η απόκρυψη των κόμβων κατάστασης και συνθήκης πλαισίου από τις (γραφικές) αναπαραστάσεις των SANs. Κατ' αυτόν τον τρόπο, επικεντρώνονται ειδικά στην ανάλυση των στόχων σε υποστόχους και ενέργειες κάνοντας έτσι πιο εύκολη την εργασία του αναλυτή και πιο απλές και κατανοητές τις αναπαραστάσεις. Επιπλέον, όταν δεν εμφανίζονται οι παραπάνω κόμβοι, τα SANs αρχίζουν να προσομοιάζουν στα Ιεραρχικά Δίκτυα Εργασιών (HTNs). Η αναπαράσταση των SANs χωρίς κόμβους κατάστασης και συνθήκης πλαισίου καλείται «απλουστευμένη» ενώ με αυτούς «πλήρης».



Σχήμα 5-6. Πλήρης και απλουστευμένη αναπαράσταση ενός SAN

Επίσης αν αποκρυβούν όλοι οι κόμβοι ενός SAN εκτός των κόμβων κατάστασης και διατηρηθεί φυσικά η δομή του γράφου, τότε προκύπτει μια αναπαράσταση της ιεραρχίας των καταστάσεων (δείτε ένα παράδειγμα στο Σχήμα 5-4 παραπάνω).

5.2.4 Αλληλεπίδραση με το περιβάλλον

Στο πλαίσιο των οδηγούμενων από γεγονότα συστημάτων, το περιβάλλον θα πρέπει να στέλνει γεγονότα προς το σύστημα. Τα γεγονότα αυτά μεταφέρουν συνήθως τις τιμές μίας ή ενός μικρού υποσυνόλου των παραμέτρων του περιβάλλοντος. Με συνδυασμό των κατάλληλων γεγονότων είναι δυνατό να συμπεραίνει κάποιος ποια ή ποιες καταστάσεις επικρατούν μια δεδομένη στιγμή. Τη στιγμή που αρχίζει να συντρέχει μια κατάσταση ενδιαφέροντος θεωρούμε ότι θα δημιουργείται και αποστέλλεται ένα σύνθετο γεγονός που θα εκφράζει αυτό ακριβώς το συμβάν (**γεγονός αρχής** ή **start event**), δηλαδή την έναρξη της κατάστασης ενδιαφέροντος. Αυτό θα περιέχει και τις τιμές των παραμέτρων που αφορούν την κατάσταση αυτή.

Σημειώνουμε ότι τα γεγονότα αρχής δημιουργούνται στην πράξη από έναν CEP engine με βάση τα CEPats που του στέλνουν οι κόμβοι-καταστάσεων. Ακόμη, μια ενδιαφέρουσα επισήμανση είναι ότι όλες οι τεχνικές και μέθοδοι που χρησιμοποιούνται στα γνωστικά πεδία της Επεξεργασίας Γεγονότων (Event Processing) καθώς και της Επίγνωσης Κατάστασης (Situation Awareness), μπορούν να χρησιμοποιηθούν και στα SANs προκειμένου να οδηγήσουν στην ανίχνευση και κατανόηση της εκάστοτε κατάστασης του περιβάλλοντος.

Στο πλαίσιο των SANs δεν υποδεικνύεται ή απαιτείται κάποιος συγκεκριμένος τρόπος έκφρασης μιας κατάστασης ή πιο σωστά του γεγονότος αρχής της. Αντίθετα μπορεί να χρησιμοποιηθεί οποιαδήποτε γλώσσα έκφρασης ή ορισμού (σύνθετων) γεγονότων. Σε μια οριακή περίπτωση θα μπορούσε να είναι ένα ερώτημα (query) για την κατάσταση του περιβάλλοντος. Στην παρούσα εργασία έχουμε επιλέξει να χρησιμοποιούμε την γλώσσα EP-SPARQL (Anićić et al., 2011), η οποία είναι μια επέκταση της SPARQL³⁸ για τον ορισμό σύνθετων γεγονότων. Με την EP-SPARQL μπορεί να ζητείται από ένα συμβατό CEP engine να στέλνει σε κάθε στόχο ένα κατάλληλο σύνθετο γεγονός αρχής, όταν φυσικά συμβεί η αντίστοιχη κατάσταση του περιβάλλοντος.

5.3 Λειτουργία και Εκτέλεση των SANs

Τα SANs είναι μεν μοντέλα συμπεριφορών συστημάτων αλλά περιέχουν και «ενεργά – εκτελέσιμα» στοιχεία. Εξάλλου όπως έχει ήδη αναφερθεί, τα SANs λειτουργούν και ως σχέδια εκτέλεσης (execution plans) για την επίτευξη των στόχων. Συνεπώς εκτός από την στατική όψη της μοντελοποίησης των συμπεριφορών, τα SANs διαθέτουν και μια δυναμική όψη, εκείνη της εκτέλεσης σχεδίων ενεργειών (οιονεί προγραμμάτων).

Ως **εκτέλεση** ή **αποτίμηση** ενός SAN νοείται η διάσχιση του αντίστοιχου ιεραρχικού γράφου, ή αλλιώς η επίσκεψη των κόμβων αυτού και η εκτέλεση των λειτουργιών που σχετίζονται με τον καθένα από αυτούς, χρησιμοποιώντας τις εκάστοτε παραμέτρους που έχουν οριστεί. Ο τρόπος διάσχισης καθώς και οι παράμετροι εκτέλεσης είναι το θέμα αυτής της ενότητας.

5.3.1 Οργάνωση και αποθήκευση των SANs

Ένα σύστημα συνήθως αποτελείται από αρκετές λογικές μονάδες (conceptual units) που επιτελούν διακριτές μεταξύ τους λειτουργίες και που εξυπηρετούν διαφορετικούς στόχους. Αντίστοιχα και οι διαδικασίες αποτελούνται από λογικές ενότητες και υποδιαδικασίες. Για λόγους καλύτερης οργάνωσης των SANs εισάγεται η έννοια των **Οντοτήτων (Entities)**, οι οποίες αντιπροσωπεύουν της λογικές μονάδες του συστήματος και χρησιμοποιούνται για την ομαδοποίηση των SANs που αφορούν την ίδια λογική μονάδα. Οι οντότητες δεν θέτουν στεγανά μεταξύ των SANs, τα οποία μπορούν να αλληλεπιδρούν μεταξύ τους ακόμη κι αν ανήκουν σε άλλες οντότητες.

³⁸ Η SPARQL είναι μια γλώσσα ερωτημάτων (σε βάσεις δεδομένων) κατάλληλη για ανάκτηση και διαχείριση δεδομένων αποθηκευμένων σε μορφή RDF. Αποτελεί δε μια βασική τεχνολογία του semantic web ενώ είναι και πρότυπο (standard) του W3C. Η έκδοση 1.0 της SPARQL εγκρίθηκε στις 15 Ιανουαρίου 2008 ενώ η έκδοση 1.1 στις 21 Μαρτίου 2013.

Τα SANs και οι σχετικές οντότητες, αφού κωδικοποιηθούν με χρήση κατάλληλης γλώσσας, αποθηκεύονται στα «αποθετήρια SANs» ή SAN repositories. Στην παρούσα εργασία έχουμε δημιουργήσει ένα συντακτικό RDF³⁹ για την κωδικοποίηση των SANs, το οποίο και χρησιμοποιήσαμε στις δοκιμές μας. Αντιστοίχως, ως SAN repository χρησιμοποιήσαμε το Sesame⁴⁰, ένα ανοιχτού κώδικα αποθετήριο RDF. Φυσικά είναι δυνατό να χρησιμοποιηθούν κι άλλες εναλλακτικές γλώσσες (π.χ. κάποιο συντακτικό XML για γράφους), και εναλλακτικά αποθετήρια (π.χ. συλλογές αρχείων ή σχεσιακές βάσεις δεδομένων).

5.3.2 Εκτέλεση των SANs

Σύμφωνα με την προσέγγισή μας, ένα σύστημα αποτελείται από ένα σύνολο από SANs. Ορισμένα από αυτά θα πρέπει να ενεργοποιούνται άμεσα με την εκκίνηση του συστήματος έτσι ώστε να ξεκινά η εκτέλεση των αρχικών λειτουργιών του συστήματος. Τα SANs αυτά θα πρέπει να επισημαίνονται ως SAN «αυτόματης εκκίνησης» (auto-start). Ομοίως και οι αντίστοιχες οντότητες. Κατά την εκκίνηση του συστήματος ενεργοποιούνται αυτόματα οι στόχοι-ρίζες όσων SANs είναι auto-start και εκδηλώνεται έτσι το ενδιαφέρον τους (και του συστήματος) για ενημέρωσή τους για την εμφάνιση των καταστάσεων του περιβάλλοντος που ενεργοποιούν την επιδίωξη των αντίστοιχων στόχων-ριζών.

Όταν εμφανιστεί η κατάσταση που αναμένει ένας στόχος-ρίζας, δηλαδή όταν ληφθεί το αντίστοιχο γεγονός αρχής, τότε δημιουργείται ένα νέο στιγμιότυπο του SAN, ένα νέο τοπικό πλαίσιο εκτέλεσης (δείτε σχετικά στην ενότητα 5.3.3) κι έπειτα αρχίζει η διάσχιση του γράφου (δείτε σχετικά στην ενότητα 5.3.4). Η δημιουργία του στιγμιότυπου έχει την έννοια ότι αν μια κατάσταση αφορά ένα τοπικό φαινόμενο ή μια συγκεκριμένη οντότητα του περιβάλλοντος, τότε η ίδια (ή καλύτερα μία παρόμοια) κατάσταση μπορεί να επαναληφθεί και για κάποια άλλη παρεμφερή οντότητα. Για παράδειγμα σε ένα σύστημα ελέγχου της καλής λειτουργίας των μηχανών A και B μιας μονάδας παραγωγής, ένας κύριος στόχος του είναι η «διατήρηση φυσιολογικής θερμοκρασίας στις μηχανές» και η αντίστοιχη κατάσταση ενδιαφέροντος είναι φυσικά η ανίχνευση «υπερθέρμανσης της μηχανής». Η κατάσταση αυτή μπορεί να συμβεί είτε όταν υπερθερμανθεί η μηχανή A είτε η B. Αν τυχόν υπερθερμανθούν και οι δύο ταυτόχρονα τότε δύο παρόμοιες καταστάσεις θα συμβαίνουν ταυτόχρονα. Τότε θα πρέπει να σταλούν δύο γεγονότα αρχής όπου το καθένα θα επισημαίνει σε ποια μηχανή αναφέρεται. Κατά συνέπεια θα ξεκινήσουν δύο εκτελέσεις του ίδιου SAN (δύο στιγμιότυπα) για την ψύξη και των δύο μηχανών.

³⁹ Η γλώσσα RDF (Resource Description Framework) αποτελεί ένα πρότυπο μοντέλο (standard model) ανταλλαγής δεδομένων στο Web. Παρέχει χαρακτηριστικά που διευκολύνουν τον συνδυασμό και συγκεκριασμό δεδομένων ακόμη κι όταν αυτά έχουν διαφορετικές μορφές (schemas). Επίσης υποστηρίζει τη (χρονική) εξέλιξη των μορφών των δεδομένων χωρίς να απαιτείται ανάλογη προσαρμογή και από τους χρήστες-καταναλωτές των δεδομένων αυτών. [Πηγή: <http://www.w3.org/RDF/>]

⁴⁰ Το Sesame είναι ελεύθερο λογισμικό ανοιχτού κώδικα που παρέχει ένα πλαίσιο εργασίας (framework) για την επεξεργασία δεδομένων σε RDF. [<http://www.openrdf.org/index.jsp>]

5.3.3 Πλαίσιο εκτέλεσης

Ο όρος context μεταφράζεται στα Ελληνικά ως περιβάλλον, πλαίσιο ή συμφραζόμενα. Ίσως ο πιο διαδεδομένος ορισμός του στην επιστήμη των υπολογιστών είναι εκείνος των Dey and Abowd (2000) που αναφέρει ότι “Πλαίσιο είναι οποιαδήποτε πληροφορία μπορεί να χρησιμοποιηθεί για να χαρακτηρίσει μια οντότητα. Μια οντότητα μπορεί να είναι ένα πρόσωπο, ένα μέρος ή ένα αντικείμενο που σχετίζεται με την αλληλεπίδραση μεταξύ ενός χρήστη και μίας εφαρμογής, συμπεριλαμβανομένων του χρήστη και της εφαρμογής”⁴¹. Οι πληροφορίες του πλαισίου αναφέρονται και ως μετα-δεδομένα ή metadata. Στο πλαίσιο της παρούσας εργασίας υιοθετούμε την άποψη των Etzion et al. (2010) ότι τα γεγονότα είναι «πηγές πλαισίου» αφού μεταφέρουν πληροφορίες για δραστηριότητες που έλαβαν χώρα στο περιβάλλον ενός συστήματος (ή και μέσα σε αυτό)⁴². Η επεξεργασία δε των γεγονότων μπορεί να έχει ως αποτέλεσμα την εξαγωγή μετα-πληροφοριών για το πλαίσιο του συστήματος.

Ειδικότερα, όσον αφορά την εκτέλεση των SANs ορίζεται η έννοια του **πλαϊσίου εκτέλεσης (execution context)** ή **πλαϊσίου λειτουργίας (operational context)**. Αυτό περιλαμβάνει όλες εκείνες τις πληροφορίες που προέρχονται από το περιβάλλον του συστήματος με τη μορφή γεγονότων, καθώς και πληροφορίες που παράγουν τα ίδια τα SANs κατά την εκτέλεσή τους. Επίσης μπορεί να περιλαμβάνονται και δεδομένα εκτέλεσης όπως μεταβλητές, μετρητές και άλλα. Οι πληροφορίες που περιλαμβάνονται στο πλαίσιο εκτέλεσης ονομάζονται **Στοιχεία ή Πληροφορίες του πλαισίου (Context elements)** και μπορούν να χρησιμοποιηθούν για τον έλεγχο και την αποτίμηση των συνθηκών πλαισίου των κόμβων συνθήκης (μεσαίοι κόμβοι των SANs), ως παράμετροι για την εκτέλεση κάποιων ενεργειών, ή για την παραγωγή νέων στοιχείων πλαισίου υψηλότερου σημασιολογικού επιπέδου (π.χ. από πληροφορίες χρόνου και απόστασης μπορεί να υπολογιστεί η ταχύτητα, ως παράγωγο μέγεθος).

Η διαδικασία εξαγωγής των πληροφοριών που μεταφέρουν τα γεγονότα αρχής και που ενδιαφέρουν ένα στόχο, καθώς και η εισαγωγή τους σε ένα πλαίσιο εκτέλεσης καλείται **πλαϊσιοποίηση (contextualization)**. Επιτυγχάνεται δε με χρήση κατάλληλων τεχνικών ανάλογα με τη μορφή και τον τρόπο κωδικοποίησης των πληροφοριών των γεγονότων. Στην υλοποίηση του λογισμικού εκτέλεσης των SANs (δείτε ενότητα 6.2) αυτό γίνεται με χρήση της γλώσσας SPARQL, η οποία είναι κατάλληλη για αναζήτηση, εξαγωγή και επεξεργασία πληροφοριών που έχουν μοντελοποιηθεί με RDF.

Τα στοιχεία πλαισίου αποθηκεύονται στα πλαίσια εκτέλεσης ως ζεύγη ονομάτων – τιμών (name-value pairs). Δεν προδιαγράφεται κάποιος συγκεκριμένος τύπος δεδομένων για τις τιμές οπότε αυτός εξαρτάται από την εκάστοτε υλοποίηση του λογισμικού εκτέλεσης των SANs. Υπάρχει όμως η απαίτηση να υποστηρίζεται ιστορικότητα για τις τιμές των στοιχείων πλαισίου, δηλαδή να διατηρούνται όλες οι τιμές τους (τρέχουσα και παρελθούσες). Στην

⁴¹ Context is “any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves”. (Dey & Abowd, 2000)

⁴² “Events are used as a source of context, since they are snippets of the past activities”. (Etzion et al., 2010)

υλοποίηση (δείτε στην ενότητα 6.2) οι τιμές των context elements είναι πάντοτε πίνακες αλφαριθμητικών (string arrays) όπου στο πρώτο στοιχείο εμφανίζεται η τρέχουσα τιμή ενώ στις επόμενες θέσεις αποθηκεύονται με χρονολογική σειρά οι προηγούμενες τιμές από την πιο πρόσφατη στην πιο παλιά. Ορίζονται οι εξής τύποι πλαισίων εκτέλεσης:

- **Τοπικό πλαίσιο εκτέλεσης (Local context).** Περιλαμβάνει πληροφορίες που αφορούν ένα συγκεκριμένο στιγμιότυπο ενός SAN. Αν σε ένα στιγμιότυπο τύχει να υπάρχουν δύο ή περισσότερες ταυτόχρονες διασχίσεις, τότε όλες τους μοιράζονται το ίδιο τοπικό πλαίσιο εκτέλεσης. Αυτό μπορεί να συμβεί αν εκτελεστεί κάποια σύνθετη ενέργεια παράλληλης διάσχησης πολλών υποστόχων.
- **Πλαίσιο εκτέλεσης Οντότητας (Entity context).** Περιλαμβάνει πληροφορίες που χαρακτηρίζουν μια λογική μονάδα του συστήματος και κατά συνέπεια την αντίστοιχη οντότητα (entity). Τις πληροφορίες αυτές μοιράζονται όλα τα SANs που ανήκουν στη συγκεκριμένη οντότητα.
- **Γενικό πλαίσιο εκτέλεσης (Global context).** Περιλαμβάνει πληροφορίες που αφορούν ολόκληρο το σύστημα ή το περιβάλλον του ή που χρειάζεται να τις μοιράζονται όλα τα SANs ανεξαρτήτως σε ποια οντότητα ανήκουν.

Τα πλαίσια εκτέλεσης δημιουργούνται δυναμικά και λαμβάνουν (αυτόματα) ένα μοναδικό αναγνωριστικό (π.χ. ένα Universal Resource Identifier, URI⁴³) προκειμένου να είναι δυνατή η αναφορά σε αυτά και η αλληλεπίδραση με αυτά. Συγκεκριμένα, προτού αρχίζει η εκτέλεση ενός στιγμιότυπου ενός SAN και εκτελεστεί ο αρχικός στόχος αυτού (στόχος – ρίζα), δημιουργείται ένα νέο τοπικό πλαίσιο. Κάθε φορά που ξεκινά μια νέα εκτέλεση του ίδιου SAN (νέο στιγμιότυπο) δημιουργείται και ένα νέο αντίστοιχο τοπικό πλαίσιο. Αν όμως αργότερα κατά τη διάσχιση του γράφου του SAN δημιουργηθούν περισσότερες από μία παράλληλες ροές διάσχησης (flows) αυτό δεν οδηγεί στην δημιουργία νέων τοπικών πλαισίων. Ένα πλαίσιο οντότητας δημιουργείται αυτόματα την πρώτη φορά που θα εκτελεστεί ένα SAN που ανήκει σε αυτήν. Αντίστοιχα το γενικό πλαίσιο δημιουργείται αυτόματα με την εκτέλεση του πρώτου SAN κατά την εκκίνηση του συστήματος. Τα πλαίσια εκτέλεσης δεν έχουν στεγανά και είναι προσβάσιμα (με το μοναδικό αναγνωριστικό τους) από όλα τα εκτελούμενα στιγμιότυπα SANs, σε όποια οντότητα κι αν ανήκουν αυτά.

5.3.4 Αλγόριθμος διάσχησης

Ο τρόπος διάσχησης των SANs βασίζεται στον αλγόριθμο προδιατεταγμένης, κατά βάθος διάσχησης γράφων (depth-first preorder graph traversal) (Cormen et al., 2001; Goodrich, 2001; Knuth, 1997), με κατάλληλες φυσικά επεκτάσεις για να υποστηρίξει την επιδίωξη και επίτευξη στόχων καθώς και μια διαδικασιακή λογική (procedural logic). Καλείται δε **αλγόριθμος εκτέλεσης των SANs** (SAN execution algorithm). Επίσης το λογισμικό που εκτελεί τα SANs υλοποιώντας τον αλγόριθμο εκτέλεσης καλείται **Μηχανισμός εκτέλεσης SANs** ή **Μηχανισμός SANs** (SAN execution engine ή **SAN engine**).

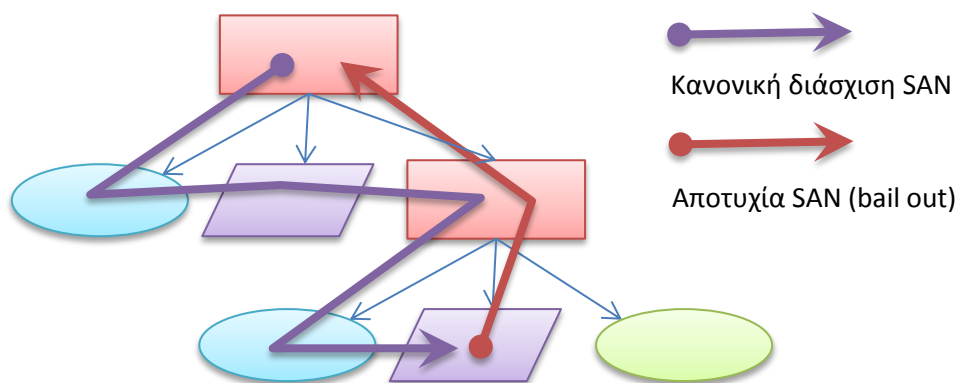
⁴³ Ένα uniform resource identifier (URI) είναι μια συμβολοσειρά (string) που χρησιμοποιείται για να προσδιορίσει μοναδικά έναν πόρο του παγκόσμιου ιστού. Τα URIs ορίζονται και περιγράφονται στο RFC 3986 / 2009 του IETF καθώς και στο Internet Protocol STD 66.

Όπως έχει ήδη αναφερθεί ο αλγόριθμος εκτέλεσης των SANs έχει κάποιες επεκτάσεις σε σχέση με τον κλασικό αλγόριθμο διάσχισης των δυαδικών δένδρων. Μια προφανής επέκταση είναι η χρήση τριών κόμβων-παιδιών αντί για δύο όπως συμβαίνει στα δυαδικά δένδρα. Συνεπώς η διάσχιση ενός SAN έχει την εξής σειρά: Κόμβος Στόχου → Κόμβος Κατάστασης (αριστερό κλαδί) → Κόμβος Συνθήκης Πλαισίου (μεσαίο κλαδί) → Κόμβος Ενέργειας ή Υποστόχου (δεξί κλαδί). Οι κόμβοι στο παράδειγμα στο Σχήμα 5-8 (δείτε παρακάτω) διασχίζονται με την εξής σειρά: R, S1, CC1, G, S2, CC2, D, A. Αν αγνοηθούν οι μη εκτελέσιμοι κόμβοι (εδώ οι κόμβοι-στόχων) τότε λαμβάνουμε το σχέδιο εκτέλεσης του SAN.

Μια δεύτερη αξιοπρόσεκτη επέκταση αφορά το γεγονός ότι οι κόμβοι των SANs είναι διαφόρων τύπων, ήτοι κόμβοι Στόχου, Κατάστασης, Συνθήκης Πλαισίου, Ενέργειας και άλλοι που θα παρουσιάσουμε αργότερα. Κάθε κατηγορία κόμβων, εκτός από ένα όνομα και ένα μοναδικό αναγνωριστικό, απαιτεί και ορισμένες επιπλέον πληροφορίες που αφορούν τη συγκεκριμένη κατηγορία κόμβων. Αυτές είναι οι **παράμετροι εκτέλεσης** του κόμβου.

Άλλη σημαντική επέκταση είναι ότι οι κόμβοι των SANs είναι «εκτελέσιμα στοιχεία». Αυτό σημαίνει πως όταν κατά τη διάσχιση ενός SAN γίνεται επίσκεψη ενός κόμβου, τότε εκτελούνται ορισμένες λειτουργίες ανάλογα με το είδος του κόμβου. Χρησιμοποιούνται δε και οι αντίστοιχες παράμετροι εκτέλεσης. Αξίζει ακόμη να επισημανθεί ότι η διάσχιση ενός SAN διακόπτεται καθ' όσο χρόνο εκτελούνται οι λειτουργίες ενός κόμβου. Εξαιρεση σε αυτόν τον κανόνα αποτελούν οι Παράλληλες ενέργειες που θα παρουσιαστούν αργότερα.

Επιπλέον, ο αλγόριθμος εκτέλεσης των SANs απαιτεί μετά την ολοκλήρωση της εκτέλεσης ενός κόμβου, να επιστρέφεται μια ένδειξη Επιτυχίας ή Αποτυχίας (SUCCESS ή FAILURE) που καλείται αποτέλεσμα του κόμβου (ή node return status). Όταν η εκτέλεση του κόμβου είναι επιτυχής τότε η διάσχιση συνεχίζει κανονικά όπως έχει ήδη περιγραφεί. Όταν όμως η εκτέλεση δεν είναι επιτυχής τότε η διάσχιση διακόπτεται με επιστρεφόμενο αποτέλεσμα «Αποτυχία» ενώ και οι κόμβοι-γονείς του κόμβου αποτυγχάνουν διαδοχικά (ήτοι αποτυχία επίτευξης όλων των στόχων-υποστόχων) από τον πιο χαμηλά μέχρι την κορυφή (ρίζα). Κατά συνέπεια και ολόκληρο το SAN αποτυγχάνει. Στα δένδρα συμπεριφορών αυτό αναφέρεται ως bail out. Εξαιρεση σε αυτόν τον κανόνα αποτελούν οι Σύνθετες ενέργειες και ορισμένοι decorators που έχουν στρατηγικές αντιμετώπισης αυτής της κατάστασης.



Σχήμα 5-7. Κανονική και bail out διάσχιση ενός SAN

Ακολουθως παρουσιάζεται ο αλγόριθμος εκτέλεσης των SANS σε ψευδοκώδικα (Πίνακας 5-1). Αρκετές από τις λεπτομέρειές του θα γίνουν πιο κατανοητές αργότερα όταν θα περιγραφούν όλοι οι τύποι κόμβων (στην ενότητα 5.4), ωστόσο η κύρια δομή του και η μέθοδος διάσχισης πρέπει ήδη να είναι εύκολα αναγνωρίσιμες από τον αναγνώστη.

Πίνακας 5-1. Αλγόριθμος εκτέλεσης των SANS σε ψευδοκώδικα

```

-- SAN execution algorithm in pseudocode

-- Executes all auto-start SANS with Root Goal
-- Each SAN is executed in a separate flow
procedure start_system_execution() is
begin
  for all RootGoals : rg do
    if rg.auto-start==TRUE then
      ctx ← create_local_context
      create_parallel_flow( traverse(rg, ctx) )
    end if
  end for
end procedure

-- Executes a Goal node (ie waits on situation sub-node,
-- checks context condition sub-node and traverses the
-- action sub-node)
-- If it is a Root Goal node then it spawns a new flow
-- each time the RG situation is detected
function traverse(Node ND, Context ctx) returns STATUS is
begin
  -- if ND is a Root Goal node (ie tree root)
  if ND is_A RootGoal then
    waitForSituation( ND.situation, ctx )
    if checkCondition(ND.contextCondition,ctx)==TRUE then
      create_parallel_flow(traverseAction(ND.action,ctx))
      ctx ← create_local_context
    end if
    traverse(ND, ctx) -- iterate for ever on RG situation
  end if

  -- if we get here, ND is a normal Goal node
  waitForSituation( ND.situation, ctx )
  if checkCondition(ND.contextCondition, ctx) == TRUE then
    return traverseAction( ND.action, ctx )
  else
    return FAILURE
  end if
end function

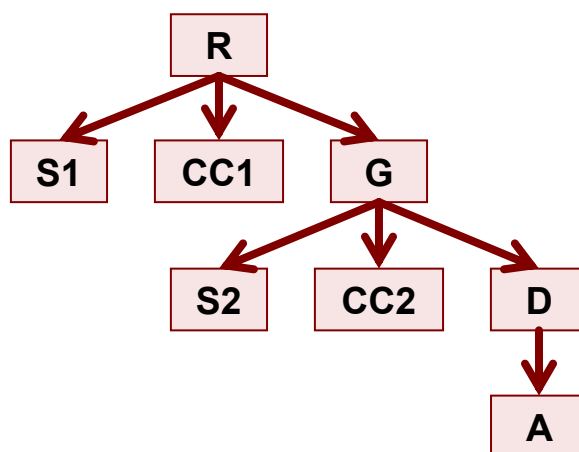
```

```

-- Checks the type of Action node and acts accordingly
function traverseAction(Node action, Context ctx)
  returns STATUS is
begin
  if action is_A PrimitveAction then
    return doPrimitiveAction( action, ctx )
  else if action is_A ComplexAction then
    return doComplexAction( action, ctx )
  else if action is_A MountAction then
    return doMountAction( action, ctx )
  else if action is_A Decorator then
    return doDecorator( action, ctx )
  else if action is_A AbstractAction then
    return doAbstractAction( action, ctx )
  else -- it is a subgoal node
    return traverse( cast_action_to_goal(action), ctx )
  end if
end function

```

Ακολούθως παρουσιάζουμε ένα παράδειγμα SAN που περιλαμβάνει αρκετούς διαφορετικούς τύπους κόμβων. Η σημασία τους θα εξηγηθεί στην επόμενη ενότητα.



Σχήμα 5-8. Παράδειγμα SAN με Στόχο, Υποστόχο και Decorator

5.4 Δομικά στοιχεία των SANs

Σε αυτή την ενότητα παρουσιάζονται συνοπτικά οι διάφοροι τύποι κόμβων των SANs. Συμπεριλαμβάνονται επίσης και οι γραφικές τους αναπαραστάσεις.

- Κόμβος Στόχου (Goal node), Κόμβος Υποστόχου (Subgoal node), Κόμβος Στόχου Ρίζας ή Κορυφής (Root Goal node).** Ουσιαστικά πρόκειται για τον ίδιο τύπο κόμβου. Οι κόμβοι στόχου μοντελοποιούν τους στόχους κορυφής (πρώτου επιπέδου) καθώς και τους υποστόχους στους οποίους αναλύονται. Κατ' αυτόν τον τρόπο επιτυγχάνεται η ιεραρχική ανάλυση των στόχων σε υποστόχους. Προφανώς οι κόμβοι υποστόχων έχουν έναν κόμβο γονέα. Αντίθετα οι κόμβοι στόχων κορυφής δεν έχουν κόμβο γονέα.

Όταν ένας κόμβος στόχου εκτελείται, η διάσχιση του SAN συνεχίζει στους κόμβους-κλαδιά του από το αριστερότερο στο δεξιότερο. Αν όλοι οι κόμβοι-κλαδιά εκτελεστούν επιτυχώς τότε και ο κόμβος στόχος θεωρείται ότι εκτελέστηκε επιτυχώς. Λογικά οι στόχοι μπορούν να έχουν ως κόμβους-γονείς μόνο άλλους στόχους (ανώτερου επιπέδου), ωστόσο στην πράξη κι άλλοι τύποι κόμβων είναι δυνατό να είναι γονείς ενός κόμβου στόχου (Decorators και Σύνθετες ενέργειες). Οι κόμβοι στόχοι κορυφής (Root Goal nodes) δεν έχουν κόμβο-γονέα καθότι είναι οι ρίζες των αντίστοιχων SANs. Ενεργούν δε ως σημεία εισόδου σε αυτά ενώ η εκτέλεσή τους ουσιαστικά ισοδυναμεί με την εκτέλεση ή αποτίμηση ολόκληρων των αντίστοιχων SANs. Παρότι σημασιολογικά οι Στόχοι Κορυφής είναι όμοιοι με τους Στόχους κατώτερων επιπέδων, έχουν ωστόσο κάποιες διαφορές κατά την εκτέλεση. Συγκεκριμένα:

- Κατά την εκκίνησή του, ο μηχανισμός εκτέλεσης των SANs ενεργοποιεί άμεσα τους Στόχους Κορυφής που έχουν σημειωθεί ως auto-start. Ομοίως και όταν προστεθεί ένας νέος στόχος κορυφής κάποια χρονική στιγμή αργότερα μετά την εκκίνηση.
- Οι κόμβοι κατάστασης των Στόχων Κορυφής έχουν μια λίγο διαφορετική μεταχείριση από τους κόμβους κατάστασης των υπολοίπων κόμβων στόχου. Οι κόμβοι κατάστασης των στόχων κορυφής ενεργοποιούνται μια φορά όταν ξεκινά η εκτέλεση του στόχου κορυφής και παραμένουν ενεργοί εσαεί ή μέχρι την απενεργοποίηση του SAN. Κατ' αυτόν τον τρόπο ο μηχανισμός των SANs μπορεί να γνωρίζει κάθε φορά που (ξανα-)συμβαίνει η αντίστοιχη κατάσταση ενδιαφέροντος και έτσι να δημιουργεί νέα στιγμιότυπα (instances) του ίδιου SAN.

Συγκεκριμένα, όταν συμβαίνει η κατάσταση που περιγράφει ο κόμβος κατάστασης και θα πρέπει φυσικά η διάσχιση να προχωρήσει στον επόμενο κόμβο, τότε δημιουργείται ένα νέο στιγμιότυπο του SAN στο οποίο και συνεχίζει η διάσχιση από τον κόμβο συνθήκης. Αντιθέτως, στο αρχικό στιγμιότυπο του SAN η διάσχιση επιστρέφει στον κόμβο κατάστασης όπου αναμένει να λάβει ένα νέο γεγονός αρχής που θα υποδηλώνει ότι η ίδια κατάσταση συνέβη ξανά. Προφανώς οι διασχίσεις των δύο στιγμιότυπων του SAN εξελίσσονται ανεξάρτητα.

- **Κόμβοι Κατάστασης (Situation nodes)** είναι τερματικοί κόμβοι (leaf nodes) με κόμβο γονέα ένα κόμβο στόχου. Μοντελοποιούν μία κατάσταση του περιβάλλοντος (που αφορά την ενεργοποίηση του στόχου-γονέα), η οποία στην παρούσα εργασία εκφράζεται ως ένα *πρότυπο σύνθετου γεγονότος* (CEPat), το οποίο καταχωρείται σε ένα μηχανισμό επεξεργασίας σύνθετων γεγονότων (CEP engine) αμέσως μόλις ο κόμβος αρχίσει να εκτελείται. Κατόπιν ο μηχανισμός των SANs εγγράφεται στην αντίστοιχη ροή γεγονότων και αναμένει για το γεγονός αρχής που θα προέρθει από το συγκεκριμένο CEPat. Μόλις ληφθεί ένα τέτοιο



γεγονός γνωρίζει πλέον ότι η κατάσταση που μοντελοποιεί ο κόμβος κατάστασης έχει συμβεί.

Ο αλγόριθμος εκτέλεσης SANs δεν κάνει καμία υπόθεση για τη γλώσσα περιγραφής της κατάστασης ή του προτύπου σύνθετου γεγονότος (CEPat). Κατ' αυτόν τον τρόπο μπορεί να χρησιμοποιηθεί οποιαδήποτε γλώσσα και οποιοσδήποτε CEP engine. Η εκτέλεση των στιγμιότυπων των SANs αναστέλλεται στους κόμβους κατάστασης μέχρι να συμβεί η αναμενόμενη κατάσταση. Μόλις όμως αυτή ανιχνευθεί τότε το αντίστοιχο πρότυπο (CEPat) καταργείται από τον CEP engine και η εγγραφή στη ροή γεγονότων αυτού ανακαλείται. Κατόπιν η εκτέλεση συνεχίζεται κανονικά και ο κόμβος ολοκληρώνει τη λειτουργία του με επιτυχία.

Οι πληροφορίες που φέρει το γεγονός αρχής και που ενδιαφέρουν το SAN, εξαγονται και αποθηκεύονται στο πλαίσιο εκτέλεσης (execution context). Η διαδικασία αυτή αναφέρεται ως πλαίσιοποίηση (contextualization) αυτών των πληροφοριών, η δε εξαγωγή τους μπορεί να γίνει με οποιαδήποτε πρόσφορη τεχνική υποστηρίζει ο SAN engine. Στη δική μας υλοποίηση του SAN engine αυτό γίνεται με χρήση της γλώσσας ερωτημάτων SPARQL, η οποία είναι κατάλληλη για εξαγωγή και επεξεργασία πληροφοριών που έχουν μοντελοποιηθεί με RDF.

- **Κόμβοι Συνθήκης Πλαισίου (Context Condition nodes)** είναι τερματικοί κόμβοι (leaf nodes) με γονέα έναν κόμβο στόχου. Εκφράζουν μια συγκεκριμένη συνθήκη που θα πρέπει να ισχύει στο πλαίσιο εκτέλεσης (context) προκειμένου να επιτραπεί η εκτέλεση της ενέργειας που περιγράφεται στον επόμενο κόμβο-παιδί (του κοινού τους κόμβου-γονέα). Οι κόμβοι συνθήκης εκτελούνται δεύτεροι μετά τους κόμβους κατάστασης. Περιέχουν δε εκφράσεις (expressions) που αποτιμώνται σε κάποια λογική τιμή, ήτοι Αληθής ή Ψευδής (true ή false), και οι οποίες εκφράσεις δύνανται να χρησιμοποιούν πληροφορίες του πλαισίου εκτέλεσης (context elements) κατά την αποτίμησή τους. Μπορούν επιπλέον να υποβάλλουν ερωτήματα (queries) σε βάσεις δεδομένων, να χρησιμοποιούν τις διαθέσιμες λειτουργίες και τελεστές, να καλούν εξωτερικά APIs ή άλλα διαθέσιμα προγραμματιστικά «αντικείμενα».

Όταν ένας κόμβος συνθήκης πλαισίου εκτελείται τότε η αντίστοιχη έκφραση αποτιμάται αμέσως και αν είναι αληθής η εκτέλεση του κόμβου τελειώνει επιτυχώς και η διάσχιση του SAN συνεχίζει στον επόμενο κόμβο. Διαφορετικά ο κόμβος τερματίζει με αποτυχία και η διάσχιση διακόπτεται.

- **Οι Κόμβοι Ενέργειας (Action nodes)** είναι τελικοί κόμβοι με γονέα έναν κόμβο στόχου. Περιγράφουν απλές ενέργειες που πρέπει να γίνουν αμέσως μόλις η διάσχιση του SAN φτάσει σε αυτούς, ήτοι όταν συμβεί η αναμενόμενη κατάσταση και αποτιμηθεί η συνθήκη πλαισίου ως αληθής, στους δύο προηγούμενους κόμβους. Οι ενέργειες πρέπει να ολοκληρωθούν με επιτυχία προτού ο αλγόριθμος εκτέλεσης συνεχίσει τη διάσχιση του SAN. Οι κόμβοι ενέργειας καλούνται επίσης και κόμβοι Απλών ενεργειών (Primitive Actions) επειδή δεν μπορούν να

αναλυθούν σε απλούστερες ενέργειες ή υποστόχους όπως συμβαίνει με τις σύνθετες ενέργειες.

Ο αλγόριθμος εκτέλεσης των SANs δεν κάνει καμία υπόθεση για τη φύση και το σκοπό των ενεργειών. Συνεπώς μπορεί να χρησιμοποιηθεί οποιαδήποτε γλώσσα περιγραφής ενεργειών, για οποιονδήποτε σκοπό και με οποιοδήποτε μηχανισμό εκτέλεσης ή εφαρμογής τους (task execution engine). Στη δική μας υλοποίηση του μηχανισμού των SANs, μια απλή ενέργεια μπορεί να χρησιμοποιηθεί για:

- αποστολή γεγονότων,
- τροποποίηση του πλαισίου εκτέλεσης,
- αποτίμηση εκφράσεων (σε Java ή JavaScript),
- εκτύπωση μηνυμάτων,
- debugging (ολοκλήρωση με επιτυχία/αποτυχία, πρόκληση εξαιρέσεων), και
- εισαγωγή χρονικών καθυστερήσεων.

- **Οι Κόμβοι Προσάρτησης (Mount Action nodes)** είναι ένας ειδικός τύπος κόμβων ενέργειας. Κάθε τέτοιος κόμβος περιέχει μια αναφορά σε ένα άλλο SAN (το μοναδικό αναγνωριστικό του), το οποίο εισάγεται στη θέση του κόμβου προσάρτησης και τον αντικαθιστά δημιουργώντας έτσι ένα νέο, συνθετότερο SAN. Κατ' αυτόν τον τρόπο γίνεται εφικτή η δημιουργία συλλογών ή βιβλιοθηκών από SANs, τα οποία μπορεί να επαναχρησιμοποιούνται. Υπογραμμίζουμε ότι η τροποποίηση αυτή αφορά μόνο τα στιγμιότυπα SANs που εκτελούν αυτή την ενέργεια και δεν επηρεάζει τα υπόλοιπα.



Τα SANs που ανήκουν σε βιβλιοθήκες ή συλλογές θα πρέπει να έχουν κόμβους στόχων κορυφής που δεν θα είναι σημειωμένοι ως auto-start. Έτσι δεν θα εκτελούνται κατά την εκκίνηση του μηχανισμού εκτέλεσης των SANs παρά μόνο όταν εισάγονται σε ένα άλλο SAN, με χρήση των κόμβων προσάρτησης. Τα SANs κάποιας βιβλιοθήκης ή συλλογής μπορούν κι αυτά με τη σειρά τους να εισάγουν άλλα SANs από τη βιβλιοθήκη ή ακόμα και τον εαυτό τους (αναδρομική προσάρτηση ή recursive mount).

- **Οι κόμβοι Decorators⁴⁴ (Decorator nodes)** είναι ενδιάμεσοι κόμβοι των SANs. Τοποθετούνται μεταξύ κόμβων στόχου και κόμβων ενεργειών ή υποστόχων, και μεταφέρουν οδηγίες συμπεριφορών (behavioural specifications) για το πως πρέπει να τροποποιηθεί η εκτέλεση των κόμβων ενέργειας ή υποστόχων. Ένας κόμβος decorator έχει ακριβώς ένα κόμβο-παιδί, το οποίο επίσης μπορεί να είναι decorator, επιτρέποντας έτσι τη δημιουργία «αλυσίδων από decorators» (decorator chaining). Άρα μια αλυσίδα από decorators μπορεί να τοποθετηθεί ανάμεσα από έναν κόμβο στόχου και ένα κόμβο ενέργειας ή υποστόχου. Σε αυτή την περίπτωση οι οδηγίες όλων των decorators επηρεάζουν τον τρόπο εκτέλεσης της ενέργειας ή του υποστόχου. Προτεραιότητα έχουν οι decorators που



⁴⁴ Οι decorators προέρχονται από το Decorator Pattern του αντικειμενοστραφούς προγραμματισμού. Χρησιμοποιούνται για να προσθέτουν και αφαιρούν λειτουργικότητες και ιδιότητες σε μεμονωμένα αντικείμενα κατά το χρόνο εκτέλεσης, ή σε κλάσεις όταν δεν είναι δυνατό το subclassing.

βρίσκονται πιο ψηλά στην αλυσίδα (πιο κοντά στον κόμβο στόχου), των οποίων οι οδηγίες εφαρμόζονται πρώτες ενώ οι decorators που βρίσκονται πιο χαμηλά στην αλυσίδα (πιο κοντά στον κόμβο ενέργειας ή υποστόχου) εκτελούνται μετά τους προηγούμενούς τους.

Ανάλογα με τον τύπο του decorator είναι δυνατό είτε να τροποποιηθεί η διάσχιση ενός SAN, είτε να μεταβληθούν οι πληροφορίες του πλαισίου εκτέλεσης (contextual information). Η πρώτη κατηγορία decorators εισάγει δομές ελέγχου της διάσχισης, όπως βρόγχους (loops) επαναλήψεων, υπό συνθήκη εκτέλεση (if-then), χρονικά περιορισμένη εκτέλεση (timeouts), δομές χειρισμού εξαιρέσεων (exception handlers), «κάλυψη» (masking) του αποτελέσματος μιας ενέργειας ούτως ώστε αυτή να είναι πάντα επιτυχής ή ανεπιτυχής. Η άλλη κατηγορία decorators προσφέρει μια μέθοδο ανάκτησης και τροποποίησης (get-set) των πληροφοριών του πλαισίου εκτέλεσης (contextual information), όπως για παράδειγμα η αύξηση της τιμής ενός μετρητή. Τέλος, παρέχονται και κάποιοι ειδικοί decorators που χρησιμεύουν κατά το χρόνο ανάπτυξης των SANs για τον έλεγχο της ορθής λειτουργίας τους (εκτύπωση μηνυμάτων και διακοπή της διάσχισης).

- **Decorator βρόχου (Loop decorator)** προκαλεί την επαναληπτική εκτέλεση του κόμβου-παιδιού του (ήτοι κόμβου ενέργειας ή υποστόχου ή σύνθετης ενέργειας ή του υπόλοιπου τμήματος της αλυσίδας από decorators). Οι επαναλήψεις συνεχίζονται καθόσον χρόνο η συνθήκη ελέγχου του βρόχου είναι αληθής. Η συνθήκη αυτή είναι μια λογική έκφραση (αποτιμάται σε Αληθής ή Ψευδής) και μπορεί να χρησιμοποιεί πληροφορίες του πλαισίου εκτέλεσης (contextual information) για την αποτίμησή της. Αποτιμάται δε σε κάθε επανάληψη πριν την εκτέλεση του κόμβου-παιδιού του. Αν σε κάποια επανάληψη η εκτέλεση του κόμβου-παιδιού αποτύχει, τότε και ο decorator βρόχου αποτυγχάνει επίσης. Αν όμως η συνθήκη βρόχου αποτιμηθεί σε ψευδής, οι επαναλήψεις μεν διακόπτονται αλλά ο decorator βρόχου τερματίζει με επιτυχία (ακόμη κι αν ο κόμβος-παιδί δεν εκτελέστηκε ούτε μία φορά). Κατόπιν η διάσχιση του SAN συνεχίζει κανονικά.

Οι decorator βρόχοι ενεργούν ως δομές ελέγχου τύπου “while-loop”, ωστόσο αν συνδυαστούν κατάλληλα με decorators μετρητή ή ελέγχου συνθήκης μπορούν επίσης να ενεργούν και ως δομές ελέγχου τύπου “for-loop” και “repeat-until”.

- **Decorator Μετρητή (Counter decorator)** απλά εκτελεί τον κόμβο-παιδί του αφού όμως πρώτα αυξήσει ή ελαττώσει μια μεταβλητή (μετρητή) κατά μία προκαθορισμένη τιμή. Η αρχική τιμή της μεταβλητής και το βήμα αύξησης ή ελάττωσης πρέπει να καθορίζονται. Η μεταβλητή-μετρητής αποθηκεύεται ως πληροφορία του πλαισίου εκτέλεσης και παραμένει διαθέσιμη και μετά την ολοκλήρωση της εκτέλεσης του decorator. Αυτός ο decorator διευκολύνει την υλοποίηση βρόχων τύπου “for-loop”.

- **Decorator Ελέγχου Συνθήκης (Condition decorator)** ελέγχει αν ισχύει μια προκαθορισμένη συνθήκη (ήτοι αν μια λογική έκφραση αποτιμάται σε αληθής) και τότε μόνο εκτελεί τον κόμβο-παιδί του, του οποίου η επιτυχής ή ανεπιτυχής εκτέλεση είναι ταυτόχρονα και το αποτέλεσμα της εκτέλεσης του decorator. Αν η συνθήκη δεν ισχύει τότε ο decorator αποτυγχάνει χωρίς να εκτελέσει τον κόμβο-κλαδί. Ο decorator ελέγχου συνθήκης ενεργεί ως δομή ελέγχου “if-then”.
- **Decorator χρονικού περιορισμού (Timer Decorator)** εκτελεί άμεσα τον κόμβο-παιδί του αλλά του επιτρέπει να εκτελεστεί μέσα σε ένα συγκεκριμένο χρονικό διάστημα. Αν η εκτέλεση του κόμβου-παιδιού ολοκληρωθεί εντός χρόνου τότε το αποτέλεσμά της (επιτυχία / αποτυχία) είναι επίσης και το αποτέλεσμα του decorator. Αν όμως παρέλθει ο προκαθορισμένος χρόνος τότε διακόπτεται η εκτέλεση του κόμβου-παιδιού και ο decorator αποτυγχάνει.
- **Decorator Χειρισμού Εξαίρεσης (Exception Handler Decorator)** εκτελεί άμεσα τον κόμβο-παιδί του. Στην περίπτωση που κατά την εκτέλεσή του ο κόμβος-παιδί παρουσιάσει κάποια τεχνική δυσλειτουργία που γίνεται προγραμματιστικά αντιληπτή ως μία εξαίρεση (exception throwing), ο decorator χειρισμού εξαίρεσης την παγιδεύει και εξετάζει αν ο τύπος της ταιριάζει με έναν προκαθορισμένο τύπο. Αν συμβαίνει αυτό τότε εκτελεί ένα δεύτερο κόμβο-παιδί, ο οποίος θα πρέπει να διαχειριστεί το πρόβλημα που προέκυψε (exception handler). Αυτός είναι ο μόνος τύπος decorator που διαθέτει δύο κόμβους-παιδιά. Αν και ο δεύτερος κόμβος-παιδί αποτύχει ή αν ο τύπος της εξαίρεσης δεν ταιριάζει με τον προκαθορισμένο τότε ο decorator χειρισμού εξαίρεσης αποτυγχάνει. Ο decorator χειρισμού εξαίρεσης ενεργεί ως μια δομή “try-catch”. Παρότι μπορεί να θεωρησει κανείς ότι η λειτουργικότητα αυτού του decorator καλύπτεται από τη σύνθετη ενέργεια «Κατ’ επιλογή εκτέλεσης» (Selector Complex Action) (δείτε παρακάτω), ο decorator χειρισμού εξαίρεσης ωστόσο, είναι χρήσιμος στην πράξη αφού μπορεί να εντοπίζει και απομονώνει σοβαρές τεχνικές δυσλειτουργίες κατά την εκτέλεση των κόμβων ενέργειας ή του μηχανισμού των SANs ή άλλου λογισμικού. Αυτές θα προκαλούσαν τη δημιουργία εξαιρέσεων και θα οδηγούσαν στον απρόσμενο τερματισμό της λειτουργίας του μηχανισμού των SANs ή ακόμα χειρότερα στην ελαττωματική λειτουργία του.
- **Decorator Επιτυχίας / Αποτυχίας (Success / Failure Decorator)** εκτελεί άμεσα τον κόμβο-παιδί του αλλά όταν αυτός ολοκληρώσει την εκτέλεσή του, ο decorator επιστρέφει πάντα «επιτυχία / αποτυχία» ως αποτέλεσμα της εκτέλεσής του, ανεξάρτητα από το αποτέλεσμα του κόμβου-παιδιού.
- **Decorator Αντιστροφής (Negation Decorator)** εκτελεί άμεσα τον κόμβο-παιδί του και αντιστρέφει το αποτέλεσμα της εκτέλεσής του (λογική πράξη

NOT). Δηλαδή αν ο κόμβος-παιδί ολοκληρώσει την εκτέλεσή του με επιτυχία τότε ο decorator αποτυγχάνει και αντίστροφα.

- **Decorator Διακοπής (Break Decorator)** όταν εκτελείται προκαλεί τη διακοπή της διάσχισης του SAN και εμφανίζει ένα μήνυμα (ή το καταγράφει στο log). Αν η διάσχιση επανεκκινηθεί τότε συνεχίζει με την εκτέλεση του κόμβου-παιδιού του, του οποίου το αποτέλεσμα είναι επίσης και το αποτέλεσμα του decorator διακοπής. Ο decorator διακοπής χρησιμεύει κατά την δημιουργία των SANs για τον έλεγχο της ορθότητάς τους και επιτρέπει την βηματική (step-by-step) εκτέλεση καθώς και για την εξέταση πληροφοριών του πλαισίου (context element inspection).
- **Decorator Εκτύπωσης (Print Decorator)** όταν εκτελείται εμφανίζει ή καταχωρεί στο log ένα μήνυμα και αμέσως μετά ξεκινά την εκτέλεση του κόμβου-παιδιού του, του οποίου το αποτέλεσμα είναι επίσης και αποτέλεσμα του decorator εκτύπωσης. Ο decorator εκτύπωσης χρησιμοποιείται για την εμφάνιση μηνυμάτων με σκοπό τον έλεγχο της ορθής υλοποίησης και λειτουργίας των SANs.
- **Κόμβοι Σύνθετης Ενέργειας (Complex Action nodes)** είναι ενδιάμεσοι κόμβοι των SANs. Εισάγουν σύνθετες δομές εκτέλεσης όπως η σειριακή, η κατ' επιλογή και η παράλληλη εκτέλεση απλών ενεργειών ή υποστόχων. Οι κόμβοι σύνθετης ενέργειας μπορούν να έχουν οποιονδήποτε αριθμό κόμβων-παιδιών, τα οποία μπορεί να είναι είτε απλές ενέργειες, είτε υποστόχοι, είτε decorators, είτε άλλοι κόμβοι σύνθετης ενέργειας. Οι κόμβοι σύνθετης ενέργειας είναι ουσιαστικά το μέσο ανάλυσης του στόχου-γονέα σε πολλαπλούς υποστόχους (goal decomposition). Επιπρόσθετα, καθορίζουν και τη στρατηγική εκτέλεσης των υποστόχων (του στόχου), αν φυσικά υπάρχουν περισσότεροι του ενός. Οι διάφοροι τύποι Σύνθετων Ενεργειών περιγράφονται ακολούθως.
 - **Κόμβος Σειριακής εκτέλεσης (Sequence node)** εκτελεί όλους τους κόμβους-παιδιά του διαδοχικά τον ένα μετά τον άλλο. Αν ένας κόμβος-παιδί αποτύχει τότε ολόκληρος ο κόμβος σειριακής εκτέλεσης αποτυγχάνει και οι υπόλοιποι κόμβοι-παιδιά (αν έχουν απομείνει κάποιοι) δεν εκτελούνται. Ουσιαστικά ο κόμβος σειριακής εκτέλεσης «απαιτεί» όλοι οι κόμβοι-παιδιά του να επιτύχουν. Άρα λειτουργεί ως τελεστής AND, όσον αφορά το αποτέλεσμα της εκτέλεσης.
 - **Κόμβος Κατ' Επιλογή εκτέλεσης (Selector node)** εκτελεί τους κόμβους-παιδιά του διαδοχικά μέχρι κάποιος να εκτελεστεί επιτυχώς. Αυτό σημαίνει πως αν ο πρώτος κόμβος-παιδί επιτύχει τότε ολόκληρος ο κόμβος κατ' επιλογή εκτέλεσης επίσης επιτυγχάνει κι έτσι δεν εκτελούνται οι επόμενοι κόμβοι. Διαφορετικά εκτελείται ο δεύτερος κόμβος-παιδί και αυτό συνεχίζεται μέχρις ότου κάποιος κόμβος-παιδί επιτύχει. Αν όλοι οι κόμβοι-παιδιά αποτύχουν τότε και ο κόμβος κατ' επιλογή εκτέλεσης αποτυγχάνει.



Ουσιαστικά ο κόμβος κατ' επιλογή εκτέλεσης «απαιτεί» (ακριβώς) ένας κόμβος-παιδί να επιτύχει. Άρα λειτουργεί ως τελεστής OR.



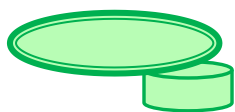
- **Κόμβος Παράλληλης εκτέλεσης τύπου ANY (Parallel Any node)** εκτελεί ταυτόχρονα όλους τους κόμβους-παιδιά του με ξεχωριστές ταυτόχρονες διασχίσεις (ανεξάρτητα threads) για τον καθένα τους, οι οποίες όμως μοιράζονται το ίδιο τοπικό πλαίσιο εκτέλεσης (local context). Αν η εκτέλεση οποιουδήποτε από τους κόμβους-παιδιά ολοκληρωθεί με επιτυχία τότε ολόκληρος ο κόμβος επίσης επιτυγχάνει. Κατόπιν στέλνει σήμα στους κόμβους-παιδιά που ακόμη εκτελούνται να διακόψουν. Ουσιαστικά αυτός είναι μια παράλληλη εκδοχή του κόμβου κατ' επιλογή εκτέλεσης.



- **Κόμβος Παράλληλης εκτέλεσης τύπου ALL (Parallel All node)** εκτελεί ταυτόχρονα όλους τους κόμβους-παιδιά του με ξεχωριστές ταυτόχρονες διασχίσεις για τον καθένα τους, οι οποίες όμως μοιράζονται το ίδιο τοπικό πλαίσιο εκτέλεσης (local context). Αν η εκτέλεση οποιουδήποτε από τους κόμβους-παιδιά ολοκληρωθεί ανεπιτυχώς τότε ολόκληρος ο κόμβος επίσης αποτυγχάνει. Κατόπιν στέλνει σήμα στους κόμβους-παιδιά που ακόμη εκτελούνται να διακόψουν. Μόνο αν ολοκληρωθούν επιτυχώς όλοι οι κόμβοι-παιδιά επιτυγχάνει και ο κόμβος Parallel ALL. Ουσιαστικά αυτός είναι μια παράλληλη εκδοχή του κόμβου σειριακής εκτέλεσης.



- **Κόμβος Παράλληλης εκτέλεσης με χρονικό περιορισμό (Parallel Timeout node)** εκτελεί ταυτόχρονα όλους τους κόμβους-παιδιά του με ξεχωριστές ταυτόχρονες διασχίσεις για τον καθένα τους, οι οποίες όμως μοιράζονται το ίδιο τοπικό πλαίσιο εκτέλεσης (local context). Κατόπιν περιμένει για ένα προκαθορισμένο χρονικό διάστημα τους κόμβους-παιδιά να ολοκληρώσουν την εκτέλεσή τους. Κάθε επιτυχής ή ανεπιτυχής ολοκλήρωση, μέσα στο προκαθορισμένο χρονικό διάστημα, καταγράφεται αλλά ο κόμβος παράλληλης εκτέλεσης με χρονικό περιορισμό δεν τερματίζει την εκτέλεσή του (όπως συμβαίνει στους προηγούμενους δύο τύπους) αλλά συνεχίζει να περιμένει μέχρις ότου παρέλθει ο προκαθορισμένος χρόνος ή όλοι οι κόμβοι-παιδιά ολοκληρώσουν την εκτέλεσή τους. Αν έστω ένας κόμβος-παιδί τερματίσει επιτυχώς τότε ολόκληρος ο κόμβος παράλληλης εκτέλεσης με χρονικό περιορισμό είναι επίσης επιτυχής. Όταν εξαντλείται το καθορισμένο χρονικό διάστημα ο κόμβος στέλνει σήμα στους κόμβους-παιδιά του που ακόμη εκτελούνται να διακόψουν.



- **Κόμβοι Αφηρημένης/Γενικής Ενέργειας (Abstract Action nodes)** είναι τερματικοί κόμβοι των SANs. Όταν η διάσχιση φτάνει σε έναν κόμβο αφηρημένης ενέργειας τότε προσδιορίζεται εκείνη τη στιγμή η ενέργεια που θα πρέπει να εκτελεστεί. Αυτό γίνεται επιλέγοντάς την από ένα προκαθορισμένο σύνολο (αποθετήριο) ενεργειών που καλείται «δεξαμενή ενεργειών» (action pool).

Μια δεξαμενή ενεργειών μπορεί να συμπεριλαμβάνει απλές ενέργειες (primitive actions) αλλά και κάθε άλλη έγκυρη δομή SAN, όπως κόμβους σύνθετων ενεργειών (composite action nodes), κόμβους στόχων (goal nodes), μέρη SANs ή και ολόκληρα SANs. Οι ενέργειες μιας δεξαμενής ενεργειών δέχονται προαιρετικά σημασιολογικές ή άλλες επισημάνσεις (annotations) υπό μορφή μετα-δεδομένων προκειμένου να επιτρέψουν τη δημιουργία «ευφύων» αλγορίθμων επιλογής.

Στον κόμβο αφηρημένης ενέργειας ορίζεται επίσης μια μέθοδος «αναζήτησης και επιλογής» (search and selection method). Αυτή χρησιμοποιείται για την εξέταση όλων των εναλλακτικών ενεργειών (ή άλλων δομών) που περιέχει η δεξαμενή ενεργειών και την αναγνώριση εκείνων που είναι σχετικές με την τρέχουσα κατάσταση και που πληρούν ορισμένα κριτήρια επιλογής.

Αν η μέθοδος αναζήτησης και επιλογής εντοπίσει ακριβώς μία ενέργεια που ταιριάζει στην τρέχουσα κατάσταση (και φυσικά τα κριτήρια επιλογής), τότε αυτή εκτελείται και το αποτέλεσμά της (επιτυχία ή αποτυχία) είναι και το αποτέλεσμα του κόμβου αφηρημένης ενέργειας. Αν δεν βρεθεί κάποια ενέργεια που να ταιριάζει τότε ο κόμβος αφηρημένης ενέργειας αποτυγχάνει. Αν όμως ταιριάζουν περισσότερες από μία ενέργειες τότε χρησιμοποιείται μια προκαθορισμένη «πολιτική επίλυσης» (resolution policy) για να αποφασίσει πως θα αντιδράσει ο αλγόριθμος διάσχισης σε αυτή την κατάσταση. Τυπικά θα επιλέξει μία από όλες ή θα δημιουργήσει μια σύνθεση αυτών.

5.5 Σύγκριση με τις σχετικές προσεγγίσεις της βιβλιογραφίας

Στη συνέχεια επιχειρούμε μια συνοπτική σύγκριση των SANs με ορισμένες αρκετά διαδεδομένες προσεγγίσεις και τεχνολογίες. Κυρίως εστιάζουμε στα μειονεκτήματά τους και εξηγούμε πως τα SANs αποφεύγουν αντίστοιχα προβλήματα. Ειδικότερα, συγκρίνουμε τα SANs με τους «Κανόνες ECA» (ECA rules), τη γλώσσα BPMN 2.0, τα ιεραρχικά δίκτυα εργασιών και τους αλγορίθμους σχεδιασμού και δυναμικού σχεδιασμού καθώς και τα δένδρα συμπεριφορών.

5.5.1 Σύγκριση με κανόνες ECA

Ο όρος Event–Condition–Action (Γεγονός – Συνθήκη – Ενέργεια) ή ECA, αφορά έναν τρόπο αποτύπωσης και υλοποίησης των κανόνων λειτουργίας συστημάτων βασισμένων στην οδηγούμενη από γεγονότα αρχιτεκτονική (Event-Driven Architecture ή EDA), καθώς και στις ενεργές βάσεις δεδομένων (active databases) (Dittrich et al., 1995). Ο λόγος που θεωρείται η μοντελοποίηση συστημάτων με κανόνες ECA ως σχετική με τα SANs είναι λόγω αυτής καθεαυτής της δομής των κανόνων. Συγκεκριμένα αποτελούνται από τρία διακριτά μέρη, τα οποία εκτελούνται διαδοχικά: (α) αναμονή και λήψη ενός συγκεκριμένου (τύπου) γεγονότος, (β) όταν παραληφθεί το αναμενόμενο γεγονός, εξετάζεται αν ισχύει μια συνθήκη ελέγχου, και (γ) αν η συνθήκη ελέγχου ισχύει, εκτελείται μια ενέργεια.

Ένα σύστημα μπορεί να μοντελοποιηθεί από ένα σύνολο κανόνων ECA (rule set), αποθηκευμένων σε μια βάση κανόνων (rule base) απ' όπου και εκτελούνται με τη βοήθεια

ενός μηχανισμού εκτέλεσης (ECA rule engine). Κάτι παρόμοιο όμως συμβαίνει και με τα SANs όπου σε κάθε κόμβο-στόχο των SANs έχουμε μια αντίστοιχη δομή, ήτοι (α) αναμονή και λήψη ενός γεγονότος κατάστασης, (β) έλεγχος αν ισχύει μια συνθήκη στο πλαίσιο εκτέλεσης, και (γ) εκτέλεση μιας ενέργειας ή ενός κόμβου-υποστόχου. Δηλαδή θα μπορούσε να παραλληλιστεί κάθε κόμβος-στόχος των SANs με έναν κανόνα ECA αφού προσομοιάζουν τόσο μορφολογικά όσο και λειτουργικά.

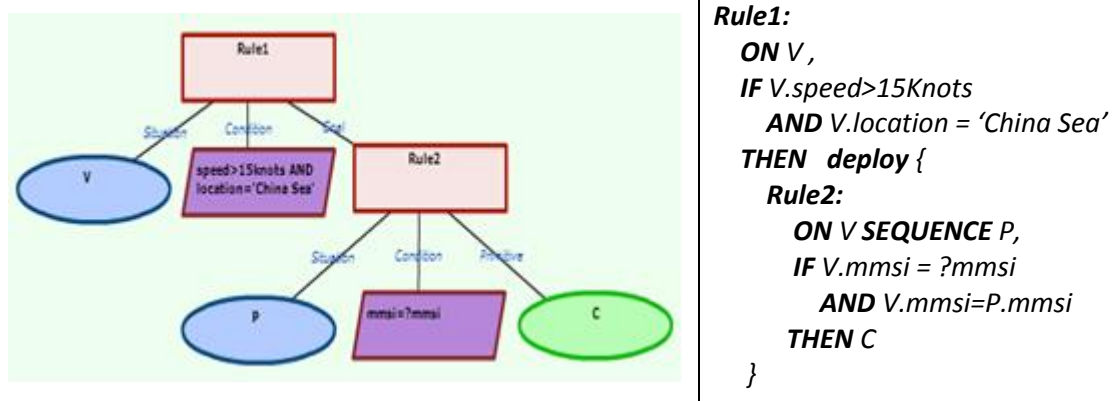
Πρέπει όμως να επισημάνουμε ότι τόσο σημασιολογικά όσο και πρακτικά υπάρχουν και διαφορές μεταξύ των δύο μοντελοποιήσεων. Συγκεκριμένα, στα SANs ως επί τω πλείστο χρησιμοποιούνται γεγονότα κατάστασης (situation events), χωρίς φυσικά να απαγορεύεται και η χρήση άλλων τύπων γεγονότων. Επιπλέον οι κόμβοι κατάστασης των SANs μπορούν να ζητήσουν από έναν CEP engine τη δημιουργία ρών γεγονότων κατάστασης από τις οποίες θα λάβουν τα αναμενόμενα γεγονότα κατάστασης, με χρήση προτύπων σύνθετων γεγονότων (CEPats). Επίσης τα λαμβανόμενα γεγονότα πλαισιοποιούνται έτσι ώστε να είναι διαθέσιμες όσες πληροφορίες τους μας ενδιαφέρουν. Ακόμη, οι συνθήκες ελέγχου στα SANs, σημασιολογικά τουλάχιστο, αφορούν το πλαίσιο εκτέλεσης (context), χωρίς όμως αυτό να είναι περιοριστικό. Τέλος στο τρίτο μέρος, τα SANs μπορούν να εκτελούν είτε απλές ενέργειες (όπως και οι κανόνες ECA) ή να εκτελούν κόμβους-υποστόχων, ενέργεια που ισοδυναμεί με την δευτερογενή εκτέλεση ενός άλλου κανόνα ή ακόμη και σύνθετες ενέργειες.

Τα SANs ως μέθοδος μοντελοποίησης της συμπεριφοράς διαδικασιών οδηγούμενων από γεγονότα υπερέχουν των απλών κανόνων ECA. Αυτό οφείλεται στην εισαγωγή δομής και ιεραρχίας στους κανόνες μέσω της ανάλυσης των στόχων σε υποστόχους. Ακόμη, οι κόμβοι σύνθετων ενεργειών λειτουργούν ως «λογικές ομαδοποιήσεις» των κανόνων, κάτι που τουλάχιστο στις βασικές προσεγγίσεις ECA δεν συμβαίνει. Θα μπορούσε να πει κανείς, ότι τα SANs επεκτείνουν τους κανόνες ECA ή ότι παρέχουν μια προσέγγιση «Ιεραρχικών κανόνων ECA».

Η ιεράρχηση των «κανόνων» στα SANs προσδίδει πολλαπλά οφέλη στη μοντελοποίηση αλλά και στην εκτέλεση αυτών. Ένα πρώτο όφελος είναι το ότι επιτρέπει την επιλεκτική ενεργοποίηση και απενεργοποίηση των κανόνων, όταν αυτοί χρειάζονται ή έχουν νόημα. Αυτό γίνεται όταν εκτελείται ο κόμβος-στόχος στον οποίο ανήκουν. Η ιδιότητα αυτή των SANs μπορεί να αξιοποιηθεί για την επίτευξη οικονομίας. Αναλυτικότερα, αν η εγγραφή σε ροές γεγονότων ή η παραλαβή των γεγονότων επιφέρει κάποιο κόστος, τότε τα SANs συμβάλουν στην ελάττωσή του αφού μειώνεται ο αριθμός των εγγραφών (subscriptions) καθώς και το πλήθος των λαμβανομένων γεγονότων στα απολύτως απαραίτητα. Ενδεικτικά το κόστος μπορεί να αφορά χρεώσεις χρήσης, επιβάρυνση του δικτύου (traffic, bandwidth) ή κατανάλωση ρεύματος (αν τα SANs εκτελούνται σε κινητό). Αντίθετα σε απλές υλοποιήσεις ECA, θα έπρεπε να είναι ταυτόχρονα ενεργοποιημένοι όλοι οι κανόνες κι έτσι θα απαιτείτο να γίνει εγγραφή στις ροές γεγονότων που αφορούν όλους τους κανόνες. Επίσης θα λαμβάνονταν και γεγονότα που δεν θα είχαν πάντα νόημα για ορισμένους κανόνες, αναλόγως βεβαίως της εκτελούμενης λειτουργίας και κατάστασης του συστήματος. Φυσικά σε πιο εξελιγμένες υλοποιήσεις ECA engines αυτό μπορεί να

αποφευχθεί με τη χρήση κατάλληλου κώδικα στις Ενέργειες των κανόνων και με χρήση των ιδιαίτερων χαρακτηριστικών και συντακτικού του εκάστοτε ECA engine. Ωστόσο αυτή είναι μια κατά περίπτωση (ad hoc) αντιμετώπιση του προβλήματος. Αντιθέτως στα SANs το πρόβλημα λύνεται ήδη από τον ορισμό τους ως ιεραρχίες στόχων, όπου ένας κανόνας ενεργοποιείται μόνο όταν εκτελείται ο αντίστοιχος στόχος. Τα SANs υπερέχουν και σε επίπεδο μοντελοποίησης αφού επιτρέπουν τον καθορισμό της σειράς (σχεδίου) ενεργοποίησης και απενεργοποίησης των κανόνων σε επίπεδο μοντέλου κι όχι στον κώδικα.

Στο ακόλουθο σχήμα δίνεται ένα παράδειγμα σύγκρισης ενός SAN με έναν κανόνα ECA για το Drools, ένα γνωστό σύστημα κανόνων ECA (JBoss community, 2014). Εξετάζοντας τον κανόνα διαπιστώνουμε ότι όταν αυτός ενεργοποιηθεί, δημιουργεί δυναμικά έναν δεύτερο κανόνα (υπονοεί μια ιεραρχία). Συνεπώς ο κανόνας είναι μεν λειτουργικά ισοδύναμος με το SAN ωστόσο σε επίπεδο μοντελοποίησης και σημασιολογίας το SAN υπερέχει καθότι παρέχει ένα πολύ πιο ξεκάθαρο «λεξιλόγιο», ανεξάρτητο από συγκεκριμένα συντακτικά, γλώσσες και συστήματα εκτέλεσης. Επίσης όταν η ανάλυση ενός υποστόχου περιλαμβάνει σύνθετες ενέργειες, αφηρημένες ενέργειες ή τμήματα SANs χρησιμοποιούμενα από κοινού σε πολλούς υποστόχους, τότε το συγκεκριμένο συντακτικό δεν επαρκεί ή απαιτεί τη χρήση ad-hoc λύσεων.



Εικόνα 5-1. Παράδειγμα SAN και ο αντίστοιχος κανόνας ECA για το Drools ECA engine

Επίσης, είναι αρκετά εύκολο να “παρακολουθήσει” και να κατανοήσει κανείς τη λειτουργία που μοντελοποιεί ένα SAN. Αφενός οι κόμβοι-στόχων λειτουργούν ως τεκμηρίωση του όλου μοντέλου και μάλιστα αρκετά ενδελεχή αφού κόμβοι-στόχων υπάρχουν σχεδόν σε όλα τα επίπεδα της ιεραρχίας. Αφετέρου η ροή της εκτέλεσης ενός μοντέλου SAN είναι αυστηρά καθορισμένη και προβλέψιμη. Πιο συγκεκριμένα η ιεραρχική ανάλυση των στόχων σε σειρές από απλές ενέργειες διαμέσου των κόμβων-στόχων (που όπως είδαμε ομοιάζουν με κανόνες ECA), αποτελεί το σχέδιο εκτέλεσης (execution plan) του SAN (δείτε στην ενότητα 5.2.2 «Ορισμός των Δικτύων Κατάστασης – Ενέργειας»). Εξάλλου με τη χρήση γραφικών αναπαραστάσεων των SANs η εργασία αυτή γίνεται ακόμη ευκολότερη. Απεναντίας είναι σημαντικά πιο δύσκολο να κατανοήσει κανείς τη λογική ενός έστω και μέτριας πολυπλοκότητας συστήματος που μοντελοποιείται αποκλειστικά από κανόνες ECA. Ειδικά αν σκεφτεί κανείς ότι η εκτέλεση ενός κανόνα ECA μπορεί να προκαλέσει την δευτερογενή εκτέλεση ενός άλλου κανόνα και ούτω καθεξής (σαν φαινόμενο “χιονοστιβάδας”), γίνεται αντιληπτή η πρακτική δυσκολία κατανόησης της λειτουργίας ενός συνόλου κανόνων ECA

και ακόμη χειρότερα της συντήρησής του. Ακόμη, η προσθήκη τεκμηρίωσης (π.χ. σχολίων) επαφίεται στην ευχέρεια του προγραμματιστή.

Μία άλλη πολύ σημαντική παράμετρος είναι το κατά πόσο οι παραπάνω προσεγγίσεις διευκολύνουν την συνεργατική ανάπτυξη συστημάτων. Επειδή τα SANs θα μπορούσε να πει κανείς ότι τεκμηριώνουν επαρκώς τον εαυτό τους (self-documented) μέσω των περιγραφών των κόμβων, είναι πολύ πιο εύκολο για μια ομάδα σχεδιαστών να αναπτύξει συνεργατικά ένα σύστημα και να δημιουργήσει και εκμεταλλευτεί επαναχρησιμοποιούμενα SANs και βιβλιοθήκες από SANs. Και τούτο επειδή η δουλειά ενός μέλους της ομάδας ερμηνεύεται σωστά και κατανοείται εύκολα από τα υπόλοιπα μέλη. Αντίθετα, μια μεσαίου μεγέθους βάση κανόνων (rule-base) είναι σημαντικά πιο δύσκολο να την κατανοήσει, συντηρήσει και επεκτείνει κανείς, ιδίως όταν αυτό πρέπει να γίνει από κάποιους που δεν την ανέπτυξαν οι ίδιοι.

Συμπερασματικά, τα SANs θα μπορούσε να πει κανείς ότι είναι επέκταση και βελτίωση της μοντελοποίησης συστημάτων με κανόνες ECA. Οι βελτιώσεις αυτές είναι τόσο σε επίπεδο μοντελοποίησης (κόμβοι-στόχοι ως τεκμηρίωση, σύνθετες ενέργειες για δημιουργία ιεραρχιών), όσο και σε επίπεδο εκτέλεσης (ενεργοποίηση / απενεργοποίηση κανόνων, μειωμένο κόστος χρήσης γεγονότων ή επιβάρυνσης του δικτύου). Ως εκ τούτων θεωρούμε ότι συστήματα μοντελοποιημένα με SANs είναι ευκολότερο να συντηρηθούν και επεκταθούν και είναι φθηνότερα κατά τη λειτουργία τους. Επίσης αυτά καθαυτά τα SANs αποτελούν έναν σημασιολογικά ανώτερο τρόπο μοντελοποίησης των συμπεριφορών και των στόχων των συστημάτων σε σχέση με τους κανόνες ECA, ειδικά όταν απαιτείται ικανότητα επίγνωσης της κατάστασης του περιβάλλοντος (situation awareness) ή επεξεργασία σύνθετων γεγονότων.

Πίνακας 5-2. Συνοπτική σύγκριση κανόνων ECA και SANs

Κανόνες ECA	SANs
Οι κανόνες ECA δεν ορίζουν κάποια δομή ή ιεραρχία για το σύστημα ή τη συμπεριφορά που μοντελοποιούν.	Τα SANs εξορισμού απαιτούν τον καθορισμό μιας ιεραρχικής δομής για την επίτευξη των επιδιωκόμενων στόχων και συμπεριφορών.
Οι κανόνες ECA μπορεί να προκαλέσουν τη δευτερογενή ενεργοποίηση άλλων κανόνων δημιουργώντας ροές εκτέλεσης που είναι δύσκολο να παρακολουθήσει κανείς. Έτσι η συντήρηση μεγάλων βάσεων κανόνων καθίσταται αρκετά δύσκολη.	Το σχέδιο εκτέλεσης των SANs είναι εύκολα κατανοητό και προβλέψιμο. Έτσι είναι σχετικά εύκολη η συντήρηση και επέκταση συστημάτων μοντελοποιημένων με SANs.
Συστήματα που στηρίζονται σε κανόνες ECA συνήθως στερούνται κάποιου σαφούς σχεδίου εκτέλεσης των εργασιών. Απλά αντιδρούν σε γεγονότα.	Τα SANs μοντελοποιούν σχέδια εκτέλεσης που περιλαμβάνουν τόσο σειριακές όσο και παράλληλες ενέργειες.

Κανόνες ECA	SANS
Η συνεργατική διαχείριση και συντήρηση μεσαίου ή μεγάλου μεγέθους συστημάτων, τα οποία βασίζονται σε κανόνες ECA, είναι δύσκολη (Whitestein Technologies AG, 2010).	Η συνεργατική διαχείριση και συντήρηση συστημάτων με SANS είναι ευκολότερη διότι η ιεραρχική τους δομή και η τεκμηρίωσή τους τα καθιστά εύκολα στην κατανόηση, ερμηνεία και φυσικά επεξεργασία, ακόμη κι από άτομα που δεν τα δημιούργησαν.
Η σημειολογία εξαρτάται από το εκάστοτε εργαλείο.	Διαθέτουν γραφική σημειολογία.

5.5.2 Σύγκριση με τη γλώσσα BPMN 2.0

Η γλώσσα Business Process Model and Notation (BPMN) είναι ένα πρότυπο του διεθνούς κινσώρτσουμ Object Management Group⁴⁵ (OMG), για τη μοντελοποίηση επιχειρησιακών διαδικασιών (business process modeling) και την καταγραφή τους σε διαγράμματα διαδικασιών (R. C. Simpson, 2004) με χρήση μιας γραφικής σημειογραφίας. Στηρίζονται δε στην απεικόνιση των διαδικασιών ως ροές εργασιών (workflows), παρόμοιων με τα διαγράμματα δραστηριότητας (activity diagrams) της Unified Modeling Language (UML) (White, 2006). Ο σκοπός της BPMN είναι να υποστηρίξει τη διαχείριση των επιχειρησιακών διαδικασιών (Business Process Management ή BPM) παρέχοντας μια προτυποποιημένη κοινή γλώσσα, κατανοητή απ' όλα τα ενδιαφερόμενα μέρη (lingua franca) όπως είναι οι αναλυτές που σχεδιάζουν τις διαδικασίες, οι τεχνικοί που τις υλοποιούν και οι υπεύθυνοι που παρακολουθούν τη λειτουργία τους. Επίσης παρέχει μια αντιστοίχιση μεταξύ των συμβόλων της γραφικής σημειογραφίας και δομών που απαιτούνται για την εκτέλεση των επιχειρησιακών διαδικασιών με χρήση εξειδικευμένων γλωσσών εκτέλεσης όπως η Business Process Execution Language (BPEL) (White, 2004). Η τρέχουσα έκδοση της BPMN είναι η BPMN 2.0 η οποία είναι επίσης και γλώσσα εκτέλεσης.

Οι δύο προσεγγίσεις (SANS και BPMN) έχουν διαφορετικές οπτικές γωνίες για το ίδιο (περίπου) θέμα, ήτοι τη μοντελοποίηση διαδικασιών. Δίνουν όμως έμφαση σε διαφορετικές πλευρές του και έρχονται να καλύψουν διαφορετικές ανάγκες. Συγκεκριμένα, η BPMN 2.0 εστιάζει στην απεικόνιση των διαδικασιών ως ροές εργασιών, με σημεία αποφάσεων και διακλαδώσεις ανάμεσά τους, καθώς και στην εκτέλεση αυτών.

Απ' την άλλη, τα SANS παρέχουν έναν πιο πρόσφορο τρόπο μοντελοποίησης όταν υπάρχει ανάγκη για δυναμικές προσαρμογές στη διαδικασία λόγω αλλαγών των καταστάσεων του περιβάλλοντος. Τούτο επιτυγχάνεται με την μοντελοποίηση των στόχων και υποστόχων και στο συσχετισμό αυτών με καταστάσεις που τους ενεργοποιούν αλλά και τις συμπεριφορές που τους επιδιώκουν και επιτυγχάνουν. Το γεγονός αυτό καθιστά τα SANS πολυπλοκότερα σε σχέση με τη BPMN 2.0, όσον αφορά τη σειρά των εκτελούμενων εργασιών. Ταυτόχρονα

⁴⁵ <http://www.omg.org/>

όμως παρέχει δυνατότητες για ευκολότερη προσαρμογή των διαδικασιών, κατά τρόπο που να διατηρεί τους στόχους τους (κάτι που η BPMN δεν αντιμετωπίζει επαρκώς). Εξάλλου όπως έχει προαναφερθεί, τα SANs τεκμηριώνουν επαρκώς τον εαυτό τους διευκολύνοντας σημαντικά τη συντήρησή τους. Στα διαγράμματα BPMN οι στόχοι και οι αναλύσεις τους δεν καταγράφονται πάντα με σαφήνεια, χάνοντας έτσι πολύτιμη πληροφορία για τη διαδικασία, ενώ καταστάσεις και συμπεριφορές που σχετίζονται με διαφορετικούς στόχους μπορεί να αλληλοπλέκονται τόσο μεταξύ τους όσο και με τις λεπτομέρειες υλοποίησης των ροών εργασιών. Κατά συνέπεια καθίσταται δυσκολότερη η αναγνώριση των επιμέρους στόχων, συμπεριφορών και καταστάσεων με επακόλουθο τη δυσκολότερη προσαρμογή των μοντέλων των διαδικασιών.

Όπως έχει αναφερθεί προηγούμενα, η BPMN μοντελοποιεί τις διαδικασίες με χρήση διαγραμμάτων ροών εργασιών, τα οποία είναι γενικευμένοι κατευθυντικοί γράφοι (επιτρέπουν κύκλους), χωρίς σαφή ιεραρχία στόχων. Αντίθετα τα SANs είναι ακυκλικοί κατευθυντικοί γράφοι (DAGs) και γι' αυτό υποστηρίζουν εγγενώς μια ιεραρχία. Στην BPMN η ιεραρχία μπορεί να απεικονιστεί εμμέσως, με χρήση υποδιαδικασιών, χωρίζοντας έτσι τη μοντελοποιούμενη διαδικασία σε πολλά διαγράμματα.

Στα SANs η έννοια του γεγονότος είναι κομβικής σημασίας. Με τη λήψη ενός γεγονότος ξεκινά (σχεδόν πάντα) η επιδίωξη των στόχων και η εκτέλεση των συμπεριφορών τους. Είναι τα γεγονότα που αναδεικνύουν την αποκρισμότητα των SANs (και κατ' επέκταση των διαδικασιών που μοντελοποιούν) στα ερεθίσματα του περιβάλλοντος. Αντιθέτως στη BPMN η λήψη ενός γεγονότος είναι απλά ένα βήμα της διαδικασίας. Αυτό κάνει λιγότερο προφανή την αποκρισμότητα της διαδικασίας στα ερεθίσματα του περιβάλλοντος.

Κάθε μία από τις δύο προσεγγίσεις έχει τα πλεονεκτήματα και τα μειονεκτήματα της. Αναλόγως με τις εκάστοτε ανάγκες μοντελοποίησης μπορεί να χρησιμοποιηθεί είτε η μία προσέγγιση είτε η άλλη, επιτυγχάνοντας όμως και οι δύο παρόμοια αποτελέσματα κατά την εκτέλεση. Η BPMN παρέχει πλούσια σημειογραφία για τη μοντελοποίηση επιχειρησιακών διαδικασιών. Τα SANs από την άλλη τεκμηριώνουν πολύ καλά τη συμπεριφορά των διαδικασιών, για κάθε (προβλέψιμη) κατάσταση του περιβάλλοντος, επιτρέποντας έτσι την ευκολότερη προσαρμογή τους με ταυτόχρονη διατήρηση των στόχων τους.

Πίνακας 5-3. Συνοπτική σύγκριση BPMN2.0 και SANs

BPMN 2.0	SANs
Μοντελοποιεί διαδικασίες συστημάτων δίνοντας έμφαση στην αλληλουχία των δραστηριοτήτων και στα σημεία λήψης αποφάσεων	Μοντελοποιεί τους στόχους συστημάτων, τις συνθήκες που «επιβάλλουν» την επίτευξή τους καθώς και τις λειτουργίες που τους επιδιώκουν
Χρησιμοποιεί κυρίως ροές εργασιών (workflows)	Χρησιμοποιεί ιεραρχικά δίκτυα (DAGs) που είναι κατάλληλα για την ανάλυση στόχων
Η λήψη ενός γεγονότος είναι απλά μια δραστηριότητα	Η λήψη γεγονότων κατάστασης ξεκινά την επιδίωξη στόχων

BPMN 2.0	SANs
Παρέχει πλούσια σημειογραφία για τη μοντελοποίηση διαδικασιών	Παρέχει περιορισμένη σημειογραφία για περιγραφή διαδικασιών ως ροές εργασιών
Παρέχει βασικές δυνατότητες προσαρμογής των διαδικασιών (π.χ. ως underspecified υποδιαδικασίες)	Υποστηρίζει προσαρμογές συμπεριφορών οδηγούμενες από τις εκάστοτε συνθήκες του περιβάλλοντος
Περιγράφει αναλυτικά τις διαδικασίες ως ροές εργασιών (workflows) και αναδεικνύει τα σημεία αποφάσεων	Τα SANs περιέχουν αναλυτική τεκμηρίωση των διαδικασιών ως προς τους στόχους, τις συμπεριφορές τους και τις καταστάσεις που τα επηρεάζουν

5.5.3 Σύγκριση με Ιεραρχικά Δίκτυα Εργασιών και Αλγόριθμους Σχεδιασμού

Οι αλγόριθμοι σχεδιασμού ιεραρχικών δικτύων εργασιών (HTNs), στη βασική τουλάχιστο μορφή τους, παράγουν σχέδια εκτέλεσης με διαδοχικές αναλύσεις των στόχων σε υποστόχους και ενέργειες. Ένα πρόβλημα αυτής της προσέγγισης είναι ότι οι αναλύσεις μπορεί να είναι αναδρομικές (recursive) με αποτέλεσμα να καταλήγουν σε σχέδια απρόβλεπτου μήκους. Για παράδειγμα ο στόχος «περπάτημα» μπορεί να αναλυθεί σε «ένα βήμα και (πάλι) περπάτημα». Γι' αυτό το λόγο ο ιεραρχικός σχεδιασμός (στη βασική του μορφή) θεωρείται αφύσικος και υψηλού κόστους επιλογή για συγκεκριμένα είδη εργασιών και ειδικά όταν υπάρχουν συνδυαστικοί στόχοι. Τα SANs από την άλλη διαθέτουν κόμβους προσάρτησης που επιτρέπουν την επαναχρησιμοποίηση (reuse), και κόμβους βρόχων που επιτρέπουν την επανάληψη μοντέλων, στόχων και συμπεριφορών κατά την εκτέλεση. Έτσι επιτρέπουν τη δημιουργία μοντέλων μικρότερου μεγέθους αποσοβώντας την επανάληψη μερών τους.

Επιπλέον οι αλγόριθμοι σχεδιασμού HTNs (αλλά και οι αλγόριθμοι σχεδιασμού γενικότερα) μοντελοποιούν τις απαραίτητες αλληλεπιδράσεις με το περιβάλλον καθώς αναζητούν μια αλληλουχία ενεργειών που οδηγεί στην επίτευξη του επιθυμητού στόχου. Όμως δεν υπάρχει πραγματική αλληλεπίδραση με το περιβάλλον αφού όλα αυτά γίνονται κατά τον σχεδιασμό κι όχι κατά την εκτέλεση του σχεδίου. Συνεπώς οι αλγόριθμοι σχεδιασμού στηρίζονται στην υπόθεση ότι το περιβάλλον είναι σχετικά σταθερό, πράγμα που συχνά δε συμβαίνει στην πράξη. Αντιθέτως, τα SANs όταν περιλαμβάνουν κόμβους αφηρημένων ενεργειών (abstract action nodes) παράγουν το σχέδιο κατά τη διάρκεια της εκτέλεσης, ακριβώς τη στιγμή που χρειάζεται. Κατ' αυτόν τον τρόπο χρησιμοποιούν πάντα τις πιο πρόσφατες διαθέσιμες πληροφορίες για το περιβάλλον ή το πλαίσιο εκτέλεσης και έτσι μπορούν να λειτουργήσουν σε συγκριτικά πιο ευμετάβλητα περιβάλλοντα. Επίσης τα τελικώς παραγόμενα σχέδια μπορεί να διαφέρουν από εκτέλεση σε εκτέλεση.

Ο δυναμικός σχεδιασμός (dynamic planning) και ο επανασχεδιασμός (re-planning), απ' την άλλη, παρέχουν πιο ρεαλιστικές λύσεις για δυναμικά περιβάλλοντα σε σχέση με τους απλούς αλγόριθμους σχεδιασμού, διότι χρησιμοποιούν μεθόδους που προβλέπουν την παρακολούθηση της εκτέλεσης των σχεδίων. Σε πολλές όμως περιπτώσεις (για παράδειγμα

(Warfield et al., 2007)) εστιάζουν μόνο στη διόρθωση των σχεδίων όταν κάτι δε λειτουργεί με τον αναμενόμενο τρόπο. Δηλαδή δεν «αντιδρούν» στις αλλαγές του περιβάλλοντος καθώς ο χρόνος τα σχέδια δεν αποτυγχάνουν. Αυτό έχει ως συνέπεια να μην λαμβάνουν υπόψη νέες ευκαιρίες ή πόρους που προστίθενται κατά τη διάρκεια της εκτέλεσης και έτσι να μην πραγματοποιούν βελτιώσεις στα σχέδια που παράγουν. Τα SANs όμως, επειδή σχεδιάζουν τις επόμενες ενέργειες ακριβώς τη στιγμή που αυτό απαιτείται, λαμβάνουν υπόψη όλες τις ευκαιρίες και πόρους που διατίθενται κι έτσι επιτυγχάνουν να δημιουργούν βελτιωμένα σχέδια εκτέλεσης.

Όταν το περιβάλλον είναι σχετικά σταθερό και τα σχέδια εκτελούνται ικανοποιητικά τότε οι αλγόριθμοι σχεδιασμού και επανασχεδιασμού λειτουργούν οικονομικότερα σε σχέση με τα SANs, υπό την έννοια ότι δε χρειάζεται να παράγουν σε κάθε εκτέλεση τα ίδια σχέδια. Τα SANs πάσχουν στη συγκεκριμένο σημείο αλλά θα μπορούσαν να εφαρμοστούν τεχνικές caching των σχεδίων προκειμένου να βελτιωθεί η επίδοσή τους και επιτευχθεί οικονομία πόρων. Η μελέτη όμως αυτών των τεχνικών είναι πέραν των ενδιαφερόντων της παρούσας διατριβής.

Πίνακας 5-4. Συνοπτική σύγκριση HTNs και SANs

Αλγόριθμοι HTNs και επανασχεδιασμού (HTN and Re-planning algorithms)	SANs
Οι κλασικοί αλγόριθμοι σχεδιασμού HTNs (HTN planners) χρησιμοποιούν αναδρομικές αναλύσεις των στόχων που μπορεί να οδηγήσουν σε μη προβλέψιμου μεγέθους σχέδια	Τα SANs παρέχουν κόμβους προσάρτησης με τους οποίους μπορεί να επιτευχθεί επαναχρησιμοποίηση μοντέλων, στόχων και συμπεριφορών κι έτσι να αποσοβηθεί το πρόβλημα
Οι HTN planners χρησιμοποιούν τις πληροφορίες του περιβάλλοντος ή του εκάστοτε πλαισίου (context) που είναι διαθέσιμες κατά τον σχεδιασμό. Όμως το σχέδιο που παράγεται μπορεί να χρησιμοποιηθεί αρκετά αργότερα όταν πιθανώς το περιβάλλον ή το πλαίσιο έχουν αλλάξει σημαντικά	Τα SANs «σχεδιάζουν» την εξέλιξή τους δυναμικά κατά το χρόνο εκτέλεσης (δείτε κεφάλαιο 9). Έτσι χρησιμοποιούν πάντα τις πιο πρόσφατες διαθέσιμες πληροφορίες για το περιβάλλον ή το πλαίσιο. Συνεπώς τα SANs μπορούν να λειτουργήσουν σε σχετικά ευμετάβλητα περιβάλλοντα. Επίσης το σχέδιο που παράγεται τελικά μπορεί να διαφέρει από εκτέλεση σε εκτέλεση
Οι αλγόριθμοι επανασχεδιασμού επιχειρούν να διορθώσουν ή συμπληρώσουν σχέδια που ήδη εκτελούνται, ούτως ώστε να εξαλείψουν τις όποιες αποκλίσεις που παρουσιάζονται. Έτσι όμως αγνοούν νέες ευκαιρίες και πόρους που εμφανίζονται κατά το χρόνο εκτέλεσης	Τα SANs σχεδιάζουν ακριβώς τη στιγμή που αυτό είναι απαραίτητο, δηλαδή όταν η εκτέλεση φτάσει σε ένα στόχο (goal) που δεν είναι αναλυμένος (abstract). Κατ' αυτό τον τρόπο σχεδιάζουν λαμβάνοντας υπόψη όλους τους διαθέσιμους πόρους και πληροφορίες, ακόμη και αυτούς που εμφανίζονται κατά την εκτέλεση

Αλγόριθμοι HTNs και επανασχεδιασμού (HTN and Re-planning algorithms)	SANs
Οι αλγόριθμοι HTNs και επανασχεδιασμού δεν «αντιδρούν» στις όποιες αλλαγές του περιβάλλοντος, καθόσον χρόνο τα σχέδια δεν αποτυγχάνουν	Τα SANs είναι πιο ευαίσθητα στις αλλαγές του περιβάλλοντος επειδή σχεδιάζουν κατά την εκτέλεση
Οι αλγόριθμοι επανασχεδιασμού και τα HTNs δεν πραγματοποιούν βελτιστοποίηση των επιδόσεων των σχεδίων τους	Τα SANs δημιουργούν νέο σχέδιο σε κάθε εκτέλεσή τους και έτσι μπορούν να έχουν πάντα βελτιωμένα σχέδια εκτέλεσης, ακόμη και όταν δεν έχει προκύψει κάποιο πρόβλημα
Οι αλγόριθμοι HTNs σχεδιάζουν μόνο μία φορά στην αρχή (εκτός κι αν προκύψει πρόβλημα). Έτσι επιτυγχάνουν οικονομία στη χρήση πόρων και ενδεχομένως καλύτερες επιδόσεις	Τα SANs σχεδιάζουν κατά τη διάρκεια της εκτέλεσης. Αν όμως το περιβάλλον δεν μεταβάλλεται σημαντικά τότε πιθανώς σε κάθε εκτέλεση να παράγεται το ίδιο σχέδιο με αποτέλεσμα τη μεγαλύτερη χρήση πόρων και τη συγκριτικά μικρότερη επίδοση. Μπορεί να χρησιμοποιηθούν τεχνικές για caching των σχεδίων

Ολοκληρώνοντας σημειώνουμε και πάλι ότι τα SANs σχεδιάζουν τις επόμενες ενέργειες κατά την εκτέλεση, ακριβώς τη στιγμή που αυτό είναι απαραίτητο και όχι ενωρίτερα. Κατ' αυτόν τον τρόπο χρησιμοποιούν πάντα τις πιο πρόσφατες και επίκαιρες πληροφορίες.

5.5.4 Σύγκριση με Δένδρα Συμπεριφορών στην Τεχνητή Νοημοσύνη Παιχνιδιών Η/Υ

Τα δένδρα συμπεριφορών που χρησιμοποιούνται στην τεχνητή νοημοσύνη παιχνιδιών Η/Υ (behavior trees in game AI) είναι αρκετά σχετικά με τους στόχους της διατριβής αφού παρέχουν τη δυνατότητα αντίδρασης σε αλλαγές του περιβάλλοντος και τη λήψη σχετικών αποφάσεων. Το κύριο πλεονέκτημά τους είναι ότι αποτελούν ένα ξεκάθαρο «λεξιλόγιο» που οδηγεί τον τρόπο εργασίας του αναλυτή. Το μειονέκτημά τους όμως είναι ότι πρέπει να προδιαγραφεί με αυτά κάθε πιθανή συμπεριφορά της εφαρμογής που μοντελοποιείται, γεγονός που συνεπάγεται υψηλό σχεδιαστικό φόρτο. Συνεπώς η χρήση τους είναι μεν πρακτική σε εφαρμογές με ντετερμινιστικά περιβάλλοντα (όπως είναι τα παιχνίδια) αλλά γίνεται πιο δυσχερής σε δυναμικά και εξελισσόμενα περιβάλλοντα, όπως συμβαίνει πολλές φορές στην πράξη. Αντιθέτως τα SANs επιτρέπουν τη σχετικά ευκολότερη μοντελοποίηση διαδικασιών που λειτουργούν σε ευμετάβλητα περιβάλλοντα χάρις τη δυνατότητα επιλογής ενεργειών (και υποστόχων) κατά την εκτέλεση που παρέχει η χρήση κόμβων αφηρημένων ενεργειών (δείτε σχετικά στο κεφάλαιο 9). Με την τεχνική αυτή μετριάζεται η ανάγκη για τον λεπτομερή προσδιορισμό των ασαφών μερών των διαδικασιών, τα οποία όμως προσδιορίζονται πλήρως κατά την εκτέλεση με βάση μεθόδους που καθορίζει ο σχεδιαστής

των SANs, ανάλογα με τη διαμόρφωση του περιβάλλοντος. Έτσι διευκολύνεται η μοντελοποίηση των διαδικασιών και σε ένα βαθμό αυτοματοποιείται.

Οι αλγόριθμοι διάσχισης των δένδρων συμπεριφορών εξειδικεύονται στις εκάστοτε εφαρμογές τεχνητής νοημοσύνης παιχνιδιών Η/Υ (Palma Durán, 2010; C. Simpson, 2014). Αυτό βέβαια είναι απαραίτητο προκειμένου να βελτιστοποιηθούν στο έπακρο οι επιδόσεις των μηχανισμών των παιχνιδιών. Τυπικά η εκτέλεση των δένδρων συμπεριφορών σε παιχνίδια Η/Υ μπορεί να διακόπτεται και να συνεχίζεται (λόγω περιορισμών χρόνου και αποκρισιμότητας των μηχανισμών των παιχνιδιών) με αποτέλεσμα να απαιτούνται πιο σύνθετοι αλγόριθμοι διάσχισης και δομές δεδομένων για την καταγραφή της κατάστασής τους. Τα SANs από την άλλη εκτελούνται συνεχόμενα (αυτό εξάλλου είναι απαραίτητο για να μη χάνονται γεγονότα) και δεν απαιτούν σύνθετους αλγορίθμους διάσχισης, με καταγραφή των σημείων διακοπής και της τρέχουσας κατάστασης.

Πίνακας 5-5. Συνοπτική σύγκριση Δένδρων Συμπεριφορών και SANs

Δένδρα Συμπεριφορών στην Τεχνητή Νοημοσύνη παιχνιδιών Η/Υ	SANs
Στην πράξη χρησιμοποιούνται στην τεχνητή νοημοσύνη παιχνιδιών Η/Υ	Είναι ανεξάρτητα πεδίου εφαρμογής
Η εκτέλεση των δένδρων συμπεριφορών σε παιχνίδια Η/Υ μπορεί να διακόπτεται και να επανεκκινείται (λόγω περιορισμών χρόνου και αποκρισιμότητας των μηχανισμών των παιχνιδιών) με αποτέλεσμα να απαιτούνται πιο σύνθετοι αλγόριθμοι διάσχισης και δομές δεδομένων για την καταγραφή της κατάστασής τους	Τα SANs εκτελούνται συνεχόμενα. Έτσι δεν απαιτείται καταγραφή των προηγούμενων καταστάσεών τους, των σημείων διακοπής ή σύνθετων μεθόδων επανα-διάσχισης (re-traverse)
Τα δένδρα συμπεριφορών δεν παρέχουν δυνατότητες εξέλιξης και έτσι απαιτούν αρκετή προσπάθεια κατά το σχεδιασμό διότι θα πρέπει να μοντελοποιούνται τα πάντα. Διαφορετικά, κατά την εκτέλεση μπορεί να εμφανιστούν απροσδόκητες συμπεριφορές	Τα SANs παρέχουν δυνατότητες εξέλιξης κατά την εκτέλεση, επιτρέποντας μέρη των γράφων τους να παραμείνουν αφηρημένα (δηλαδή να μην αναλύονται). Αυτό σημαίνει ότι κατά το σχεδιασμό δεν απαιτείται ο προσδιορισμός κάθε λεπτομέρειας και κάθε πιθανής ροής εκτέλεσης. Τα αφηρημένα μέρη του γράφου προσδιορίζονται κατά την εκτέλεση, αξιοποιώντας διάφορες μεθόδους (δείτε σχετικά στο κεφάλαιο 9)

Τα SANs προσομοιάζουν αρκετά με τα δένδρα συμπεριφορών, αφού αυτά αποτέλεσαν την έμπνευση για τη δημιουργία των SANs. Συνεπώς μοιράζονται αρκετά κοινά χαρακτηριστικά και έννοιες. Ωστόσο τα SANs διαφοροποιούνται σε ορισμένα σημεία κι επίσης έχουν

εμπλουτιστεί με ιδιότητες που τυπικά δεν διαθέτουν τα δένδρα συμπεριφορών (όπως για παράδειγμα η δυνατότητα αφαίρεσης ενεργειών).

Σημειώνουμε ότι τα δένδρα συμπεριφορών στην τεχνολογία συμπεριφορών (behavior trees in behavior engineering) χρησιμοποιούνται για την καταγραφή των συμπεριφορών και των λειτουργικών απαιτήσεων συστημάτων μεγάλης κλίμακας. Στοχεύουν στην τυποποίηση της μετάβασης από τις απαιτήσεις στο σχεδιασμό λογισμικού. Ωστόσο δεν παρέχουν κάποια λύση σε προβλήματα του χρόνου εκτέλεσης ούτε και παρέχουν τη δυνατότητα σχεδιασμού.

5.5.5 Ικανοποίηση των ερευνητικών ερωτημάτων

Ο Πίνακας 5-6 που ακολουθεί δίνει μια σύνοψη των περιορισμών των προσεγγίσεων που εξετάστηκαν σε σχέση με το πρώτο ερευνητικό ερώτημα που τέθηκε στο κεφάλαιο 4, ήτοι «Αντίληψη της ανάγκης προσαρμογής με χρήση γεγονότων». Το ερώτημα αυτό σχετίζεται περισσότερο με τα SANs, σύμφωνα με το προτεινόμενο πλαίσιο της διατριβής (Σχήμα 4-10 και Πίνακας 4-2).

Πίνακας 5-6. Περιορισμοί των προσεγγίσεων σε σχέση με το πρώτο ερευνητικό ερώτημα

Παράμετροι ερ. ερωτήματος Προσεγγίσεις	Αξιοποίηση γεγονότων	Επίγνωση Κατάστασης	Αποτύπωση τρόπου ανίχνευσης	Ανίχνευση ανάγκης προσαρμογής
Κανόνες ECA	Ναι. Εγγραφή σε όλες τις σχετικές πηγές γεγονότων απ' την αρχή	Ναι. Με ενεργοποίηση των αντίστοιχων κανόνων	Ως σύνολα κανόνων (πολύπλοκη προσέγγιση)	Διαπιστώνεται από τους κανόνες (πολύπλοκη προσέγγιση)
BPMN 2.0	Ναι. Ως μηνύματα μιας διαδικασίας	Όχι. Δεν υποστηρίζεται εγγενώς απ' τη σημειογραφία	Ως διαδικασία	Διαπιστώνεται από τη διαδικασία
Ιεραρχικά Δίκτυα Εργασιών (HTNs)	Όχι.	Όχι. Σχεδιάζονται πριν την εκτέλεση	Όχι	Όχι
Δυναμικός σχεδιασμός και Επανασχεδιασμός	Μερικώς. Μόνο για την ανίχνευση αποτυχίας του σχεδίου	Μόνο όταν εκτελείται ο αλγόριθμος για επανασχεδιασμό	Όχι	Διαπιστώνεται μόνο όταν αποτύχει το σχέδιο
Δένδρα Συμπεριφορών στην τεχνητή νοημοσύνη παιχνιδιών Η/Υ	Ναι. Με γεγονότα του παιχνιδιού (όχι CEP)	Ναι. Με γεγονότα και ιεραρχική ανάλυση στόχων	Ως δένδρο συμπεριφοράς	Συνήθως δεν απαιτείται προσαρμογή στα παιχνίδια Η/Υ
Δίκτυα Κατάστασης – Ενεργειας (SANs)	Ναι. Με χρήση CEP. Επίσης αυτόματη εγγραφή – διαγραφή από πηγές γεγονότων όταν απαιτείται	Ναι. Με συνδυασμό CEP και ιεραρχικής ανάλυσης στόχων	Ως SAN	Διαπιστώνεται με CEP και έλεγχο συνθηκών του πλαισίου

Όπως προκύπτει και από τον παραπάνω πίνακα (Πίνακας 5-6), οι προσεγγίσεις που εξετάστηκαν απαντούν μόνο μερικώς στο πρώτο ερευνητικό ερώτημα. Συνεπώς καμιά τους δεν είναι κατάλληλη για το σκοπό της διατριβής. Ωστόσο αποτελούν τη βάση επάνω στην οποία αναπτύχθηκε η προτεινόμενη λύση, δηλαδή τα SANs, η οποία δανείστηκε ιδέες από αυτές έτσι ώστε να διαθέτει τα πλεονεκτήματά τους και παράλληλα να απαντά και στο πρώτο ερευνητικό ερώτημα.

Με βάση τις παραπάνω συγκρίσεις διαπιστώνουμε ότι τα SANs απαντούν σε επαρκή βαθμό σε όλες τις παραμέτρους του πρώτου ερευνητικού ερωτήματος. Επίσης υπερέχουν των σχετικών προσεγγίσεων, οι οποίες είτε δεν απαντούν σε κάποιες παραμέτρους είτε δεν τις απαντούν πλήρως. Συνεπώς τα SANs είναι κατάλληλα για χρήση στο προτεινόμενο πλαίσιο εργασίας της διατριβής.

Στο επόμενο κεφάλαιο παρουσιάζεται μια πλατφόρμα εργαλείων λογισμικού για τη δημιουργία, επεξεργασία και εκτέλεση SANs.

6 Υλοποίηση των Δικτύων Κατάστασης – Ενέργειας

Στο παρόν κεφάλαιο παρουσιάζεται το Σύστημα Επίγνωσης Κατάστασης (*Situation Awareness System, SAS*) που παρέχει μια υλοποίηση των SANs. Το SAS περιλαμβάνει εργαλεία λογισμικού που απαιτούνται για τη δημιουργία, επεξεργασία και εκτέλεση των SANs. Συνεργάζεται δε με τον δίαυλο γεγονότων (event bus) καθώς και το μηχανισμό επεξεργασίας σύνθετων γεγονότων (CEP engine).

Πρακτικά το SAS απαρτίζεται από ξεχωριστά πακέτα και βιβλιοθήκες λογισμικού, τα οποία συνδυάζονται σε ένα αρμονικό σύνολο. Μπορεί να χρησιμοποιηθεί είτε μόνο του (stand-alone) ή να αποτελείσει μέρος ενός ευρύτερου έργου (project). Ακολουθώς δίνεται η λογική αναπαράσταση της πλατφόρμας των SANs.



Σχήμα 6-1. Η λογική αρχιτεκτονική του SAS

Στο λογικό διάγραμμα του SAS διακρίνουμε τις ακόλουθες μονάδες, οι οποίες ανάλογα με την υλοποίηση ενδέχεται να είναι ξεχωριστά πακέτα λογισμικού ή μέρη του ίδιου πακέτου.

- **Επεξεργαστής SANs (SAN Editor).** Είναι το περιβάλλον το οποίο χρησιμοποιεί ένας αναλυτής προκειμένου να δημιουργήσει και τροποποιήσει τα SANs. Το εργαλείο αυτό παρέχει ένα γραφικό, εύκολα κατανοητό και διαισθητικό τρόπο επεξεργασίας των SANs, έτσι ώστε να διευκολύνει τον αναλυτή στην εργασία του. Ο επεξεργαστής

των SANs μπορεί να αποθηκεύει τα SANs που δημιουργεί στο αποθετήριο SANs, καθώς και σε αρχεία.

- **Αποθετήριο SANs (SAN Repository).** Είναι το μέσο/χώρος όπου αποθηκεύονται τα SANs που δημιουργεί ο επεξεργαστής SANs και απ' όπου τα φορτώνει και εκτελεί ο μηχανισμός εκτέλεσης των SANs. Το αποθετήριο SANs μπορεί να είναι κάτι απλό όπως ένα αρχείο ή σύνολο αρχείων, ή μπορεί να είναι κάτι πιο σύνθετο όπως μια βάση δεδομένων ή ένα RDF repository. Η περαιτέρω μελέτη του αποθετηρίου SANs θεωρείται τετριμμένη και πέραν των ενδιαφερόντων της διατριβής.
- **Μηχανισμός (εκτέλεσης των) SANs (SAN Engine).** Είναι η καρδιά του SAS αφού αυτός φορτώνει από το αποθετήριο κι έπειτα εκτελεί τα SANs. Επιτελεί μια σειρά από λειτουργίες όπως η διάσχιση των γράφων των SANs, η διαχείριση των στιγμιότυπων των SANs και των οντοτήτων, η διαχείριση των πλαισίων εκτέλεσης, η αποστολή, λήψη και contextualization των γεγονότων, η δημιουργία (όταν χρειάζεται) προτύπων σύνθετων γεγονότων (CEPats) και η ενεργοποίησή τους σε έναν CEP engine, η εγγραφή σε ροές γεγονότων και η διαγραφή από αυτές, καθώς και η προώθηση για εκτέλεση των απλών ενεργειών σε εξειδικευμένη μονάδα λογισμικού. Ο μηχανισμός των SANs παρέχει κατάλληλο περιβάλλον διαχείρισης απ' όπου θα φορτώνονται και θα εκτελούνται τα SANs και θα παρακολουθείται η λειτουργία τους. Είναι επιθυμητό ο μηχανισμός SANs να μπορεί να λειτουργήσει κι ως μέρος ευρύτερου λογισμικού, π.χ. ως ενσωματωμένη μονάδα (embedded unit).
- **Μηχανισμός εκτέλεσης απλών ενεργειών (Action execution engine).** Πρόκειται για μονάδα λογισμικού ή βιβλιοθήκη που δέχεται τις περιγραφές συγκεκριμένων απλών ενεργειών και τις φέρει εις πέρας. Ανάλογα με την υλοποίηση του SAS ο μηχανισμός εκτέλεσης απλών ενεργειών μπορεί να διαφέρει. Η προδιαγραφή και η περαιτέρω μελέτη αυτής της μονάδας του SAS ξεφεύγει από τα ενδιαφέροντα της παρούσας διατριβής.
- **Βιβλιοθήκη λογισμικού SANs (SAN Library).** Είναι επαναχρησιμοποιήσιμες μονάδες κώδικα που επιτελούν συγκεκριμένες απλές ή συνθετότερες λειτουργίες, σχετικές με την εκτέλεση και τον κύκλο ζωής των SANs. Τις μονάδες αυτές μπορούν να τις μοιράζονται τα εργαλεία που αποτελούν το SAS αλλά και τρίτες εφαρμογές. Μερικά παραδείγματα απλών λειτουργιών είναι η φόρτωση ενός αρχείου και η εγγραφή σε μια ροή γεγονότων, ενώ παραδείγματα πιο σύνθετων είναι η φόρτωση και εκτέλεση ενός SAN ή η δημιουργία και ενεργοποίηση ενός CEPat. Ακόμη, σχεδόν ολόκληρος ο μηχανισμός εκτέλεσης SANs μπορεί να είναι μέρος αυτής της βιβλιοθήκης, έτσι ώστε να μπορεί να επαναχρησιμοποιείται από τρίτες εφαρμογές.

Τα επόμενα δύο στοιχεία της λογικής αρχιτεκτονικής είναι αυτόνομες εφαρμογές, εκτός του SAS. Γι' αυτό απαιτούνται κατάλληλες προγραμματιστικές διεπαφές μεταξύ αυτών και του μηχανισμού των SANs που θα αλληλεπιδρά μαζί τους.

- **Δίαυλος Γεγονότων ή Event Bus.** Είναι εξειδικευμένο λογισμικό με κύριο σκοπό του τη διαβίβαση γεγονότων από τις πηγές τους προς όλους τους ενδιαφερόμενους. Ο δίαυλος γεγονότων που θα χρησιμοποιηθεί, θα πρέπει να υποστηρίζει τη λογική των ροών γεγονότων (τουλάχιστο σε λογικό επίπεδο) καθώς επίσης και λειτουργίες

εγγραφής/διαγραφής (subscribe/unsubscribe) και αποστολής σε αυτές (publish). Ο δίαυλος γεγονότων μπορεί ακόμη να παρέχει λειτουργίες όπως η αποθήκευση γεγονότων και η αναζήτηση αυτών με χρήση κατάλληλων ερωτημάτων (queries). Τα διακινούμενα γεγονότα μπορεί να κωδικοποιούνται με οποιαδήποτε γλώσσα μπορεί να υποστηρίξει ο δίαυλος γεγονότων.

Η αναλυτικότερη προδιαγραφή και η περαιτέρω μελέτη των διαύλων γεγονότων ξεφεύγει από τα ενδιαφέροντα της διατριβής. Μπορεί να χρησιμοποιηθεί οποιαδήποτε υλοποίηση διαύλου γεγονότων, αρκεί αυτή να καλύπτει τις προαναφερθείσες προδιαγραφές και φυσικά να έχει υλοποιηθεί μια κατάλληλη προγραμματιστική διεπαφή μεταξύ αυτού και του μηχανισμού των SANs. Αυτή μεταξύ άλλων θα αναλαμβάνει και τη μετατροπή του περιεχομένου των γεγονότων σε μορφή καταληπτή από τον μηχανισμό των SANs.

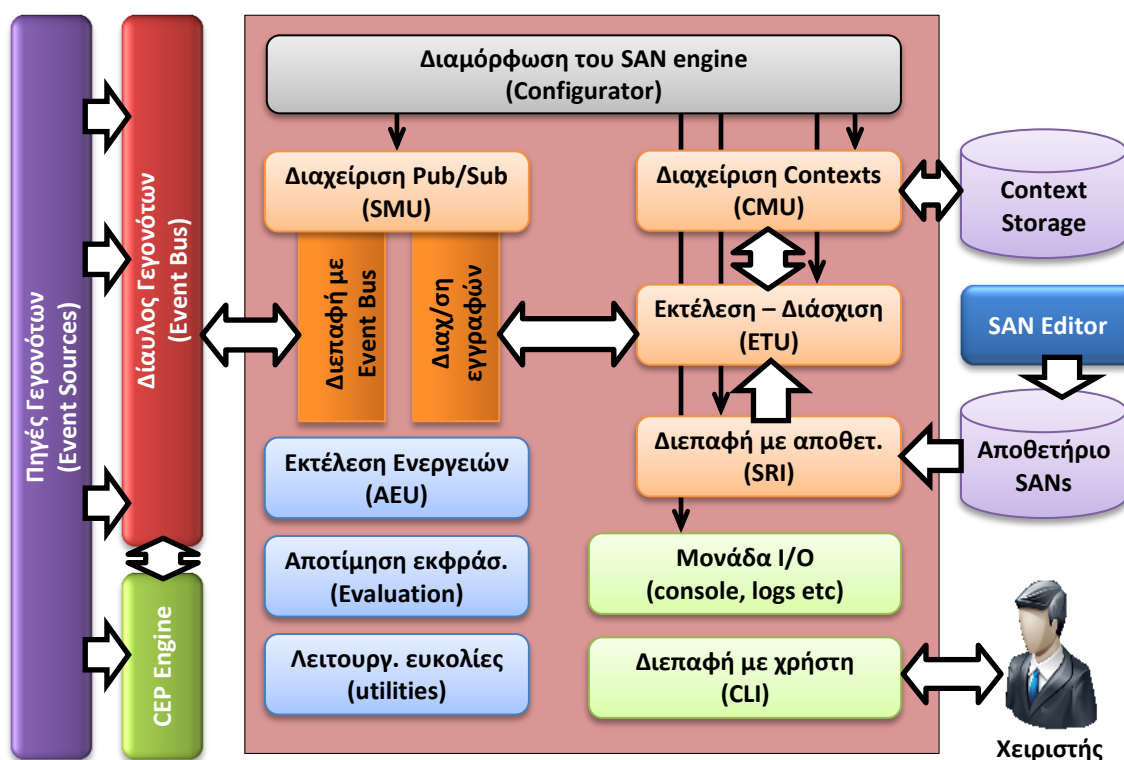
- **Μηχανισμός Επεξεργασίας Σύνθετων Γεγονότων (CEP Engine).** Είναι εξειδικευμένο λογισμικό που αποσκοπεί στη δημιουργία σύνθετων γεγονότων από ροές απλούστερων γεγονότων, με χρήση κατάλληλων *προτύπων σύνθετων γεγονότων* (CEPats). Ο CEP engine που θα χρησιμοποιηθεί από το SAS, θα πρέπει να είναι σε θέση να δέχεται περιγραφές CEPats από τον SAN engine (π.χ. ως ειδικά γεγονότα ή μηνύματα) και κατόπιν να τις ενσωματώνει στους κανόνες επεξεργασίας (π.χ. στη rule base που χρησιμοποιεί). Επιπλέον θα πρέπει να είναι σε θέση να απενεργοποιεί και διαγράφει όσα CEPats δεν χρειάζονται πλέον. Τέλος, είναι θεμιτό να δέχεται CEPats που χρησιμοποιούν τρέχοντα αλλά και παρελθόντα (ιστορικά) γεγονότα.

Η αναλυτικότερη προδιαγραφή και η περαιτέρω μελέτη των CEP engines ξεφεύγει από τα ενδιαφέροντα της διατριβής. Με το SAS μπορεί να χρησιμοποιηθεί οποιαδήποτε υλοποίηση CEP engine αρκεί να καλύπτει τις προαναφερθείσες προδιαγραφές και φυσικά να έχει υλοποιηθεί μια κατάλληλη προγραμματιστική διεπαφή μεταξύ αυτού και του μηχανισμού των SANs.

Στη συνέχεια θα παρουσιαστούν σε μεγαλύτερη ανάλυση τα δύο σημαντικότερα στοιχεία του SAS, ήτοι ο μηχανισμός εκτέλεσης των SANs και ο επεξεργαστής των SANs. Φυσικά και τα υπόλοιπα μέρη της πλατφόρμας είναι απαραίτητα αλλά τα δύο προαναφερθέντα είναι εκείνα που εμπíπτουν στα ενδιαφέροντα της διατριβής.

6.1 Αρχιτεκτονική του Μηχανισμού Εκτέλεσης των SANs (SAN engine)

Ο μηχανισμός εκτέλεσης των SANs είναι ένα σύνθετο σύστημα λογισμικού αποτελούμενο από διαφορετικές αλλά στενά συνεργαζόμενες υπομονάδες. Προκειμένου να επιτευχθεί αυτό έχει οριστεί μία προγραμματιστική διεπαφή (API) που χρησιμοποιείται απ' όλα τα μέρη του. Χάρης το API καθίσταται εύκολη η αντικατάσταση των διαφόρων μερών του SAN engine με εναλλακτικές υλοποιήσεις. Έτσι παρέχεται ένα ευέλικτο και προσαρμόσιμο εργαλείο. Το πλήρες API δίνεται στο παράρτημα III. Ακολούθως παρατίθεται η λογική αρχιτεκτονική (conceptual architecture) του μηχανισμού εκτέλεσης των SANs.



Σχήμα 6-2. Λογική αρχιτεκτονική του Μηχανισμού εκτέλεσης των SANs

Εν συνεχεία περιγράφονται συνοπτικά οι διάφορες μονάδες που απαρτίζουν τον μηχανισμό εκτέλεσης των SANs.

- **Μονάδα διαμόρφωσης του SAN engine ή SAN engine Configurator.** Είναι μια σημαντική μονάδα αφού αποτελεί το σημείο εκκίνησης και ελέγχου του SAN engine. Σκοπός της είναι να διαβάζει ένα κατάλληλο αρχείο διαμόρφωσης (configuration file) κι έπειτα να αρχικοποιεί όλες τις υπόλοιπες μονάδες. Χάρης αυτήν είναι δυνατή η εύκολη αντικατάσταση των διαφόρων μονάδων με εναλλακτικές υλοποιήσεις, απλά τροποποιώντας το αρχείο διαμόρφωσης. Η μονάδα διαμόρφωσης παραμένει ενεργή καθόσον χρόνο εκτελείται ο SAN engine και διατηρεί αναφορές (δείκτες) προς όλες τις μονάδες που έχει αρχικοποιήσει. Επίσης διαχειρίζεται τις παραμέτρους διαμόρφωσης αλλά και τις παραμέτρους της εφαρμογής που τυχόν περιέχονται στο αρχείο διαμόρφωσης.
- **Μονάδα εκτέλεσης και διάσχισης των γράφων ή Execution and Traversal Unit (ETU).** Είναι η καρδιά του SAN engine αφού αυτή συντονίζει όλη τη βασική του λειτουργία. Είναι για τον SAN engine ότι η CPU για τον υπολογιστή. Η ETU υλοποιεί τον αλγόριθμο διάσχισης των γράφων των SANs, όπως περιγράφηκε στην ενότητα 5.3.4 «Αλγόριθμος διάσχισης», κι επίσης εκτελεί όλες τις εργασίες που σχετίζονται με το κάθε είδος κόμβου. Όταν απαιτείται η συνδρομή άλλων μονάδων αυτές καλούνται μέσω του API του SAN engine. Η ETU φορτώνει για εκτέλεση (από το SAN repository) τα μοντέλα των SANs και διαχειρίζεται τον κύκλο ζωής τους. Επίσης δημιουργεί και διαχειρίζεται όλα τα στιγμιότυπα SANs καθώς και τις αντίστοιχες οντότητες (entities).

- **Μονάδα διαχείρισης του πλαισίου εκτέλεσης ή Context Management Unit (CMU).** Αυτή η μονάδα έχει ως αποστολή της τη δημιουργία και διαχείριση πλαισίων εκτέλεσης (execution contexts) για όλα τα στιγμιότυπα των SANs. Επιπλέον παρέχει μια σειρά από βοηθητικές λειτουργίες για ευκολότερη προσπέλαση, αναζήτηση και τροποποίηση των τιμών των πληροφοριών πλαισίου (context elements). Προφανώς η ETU (δείτε πριν) μέσω των κατάλληλων μεθόδων του API χρησιμοποιεί την CMU για να δημιουργήσει και προσπελάσει τα εκάστοτε πλαίσια εκτέλεσης. Η CMU μπορεί να υποστηρίζεται κατά τη λειτουργία της από κατάλληλο Context repository όπου αποθηκεύονται τα δεδομένα των πλαισίων.
- **Διεπαφή με Αποθετήριο SANs ή SAN Repository Interface (SRI).** Σκοπός του είναι η εύρυθμη επικοινωνία μεταξύ της ETU και του αποθετηρίου SANs. Επειδή το αποθετήριο των SANs δεν αποτελεί μέρος του SAN engine (αλλά της αρχιτεκτονικής του SAS), απαιτείται μια προγραμματιστική διεπαφή που θα εναρμονίζει το εσωτερικό API του SAN engine με το API που παρέχει η εκάστοτε υλοποίηση του αποθετηρίου των SANs. Για παράδειγμα αν το αποθετήριο είναι μια βάση δεδομένων τότε η SRI από τη μια θα εκτελεί όλες τις ενέργειες που απαιτούνται για σύνδεση στη βάση δεδομένων και την τήρηση των δεδομένων σε πίνακες, ενώ από την άλλη θα παρέχει την προδιαγεγραμμένη διεπαφή (API) προς τις υπόλοιπες μονάδες του SAN engine.
- **Μονάδα διαχείρισης εγγραφών σε ροές γεγονότων ή Subscription Management Unit (SMU).** Στην πραγματικότητα αυτή η μονάδα κάνει πολύ περισσότερα απ' ό τι υπονοεί ο τίτλος της. Κατ' αρχήν η SMU αποτελείται από δύο μέρη (υπομονάδες). Η πρώτη παρέχει το κατάλληλο API στα άλλα μέρη του SAN engine ενώ η δεύτερη αποτελεί την προγραμματιστική διεπαφή (ή τον προσαρμογέα) με τον εκάστοτε χρησιμοποιούμενο δίαυλο γεγονότων και τον CEP engine.

Η πρώτη υπομονάδα είναι υπεύθυνη για τη διαχείριση των εγγραφών (subscriptions) των διαφόρων στιγμιότυπων SANs (των κόμβων-καταστάσεών τους) στις διάφορες ροές γεγονότων, ενώ αναλαμβάνει και την αποστολή γεγονότων (publish) σε αυτές. Επίσης, είναι υπεύθυνη για την παράδοση των λαμβανομένων (από τον δίαυλο γεγονότων) γεγονότων στους κόμβους-κατάστασης που τα αναμένουν. Τέλος υλοποιεί και κάποιες βασικές βελτιστοποιήσεις όπως η κοινή χρήση ροών γεγονότων που χρησιμοποιούνται από πολλούς κόμβους-κατάστασης ταυτόχρονα (event stream sharing).

Η δεύτερη υπομονάδα είναι υπεύθυνη για τη διεπαφή με τον εκάστοτε χρησιμοποιούμενο δίαυλο γεγονότων (event bus) και τον CEP engine. Προφανώς η προηγούμενη υπομονάδα αναθέτει (delegates) την πραγματική εκτέλεση των λειτουργιών publish / subscribe σε τούτη, η οποία με τη σειρά της επικοινωνεί με τον δίαυλο γεγονότων ή τον CEP engine για την υλοποίηση των αιτούμενων λειτουργιών. Επίσης παραλαμβάνει τα εισερχόμενα γεγονότα και τα παραδίνει στην προηγούμενη υπομονάδα για περαιτέρω επεξεργασία. Τέλος αναλαμβάνει την αποστολή και ενεργοποίηση (και αντίστροφα απενεργοποίηση) στον CEP engine των CEPats που ζητούνται από τους κόμβους-κατάστασης.

Οι υπόλοιπες μονάδες του SAN engine είναι βοηθητικές ωστόσο του προσθέτουν αρκετές προηγμένες δυνατότητες και ευκολίες.

- **Μονάδα Εισόδου – Εξόδους ή Input – Output Unit (IOU).** Αναλαμβάνει την επικοινωνία του SAN engine με το λειτουργικό σύστημα και τον χρήστη. Εξορισμού, ο SAN engine γράφει τα μηνύματά του στα standard output ή error streams και διαβάζει τις εντολές του από standard input stream του λειτουργικού συστήματος. Με χρήση όμως μιας μονάδας I/O αυτό μπορεί να αλλάξει έτσι ώστε όλα τα μηνύματα να αποθηκεύονται π.χ. σε ένα αρχείο και οι εντολές να διαβάζονται από άλλο αρχείο. Εναλλακτικές υλοποιήσεις μπορεί να χρησιμοποιηθούν για logging ή για καταγραφή σε άλλο μέσο ή με διαφορετικό τρόπο.
- **Διεπαφή με τον χρήστη ή Command Line Interface (CLI).** Είναι ένα αλληλεπιδραστικό περιβάλλον χρήσης του SAN engine. Παρέχει τη δυνατότητα στον χειριστή του SAN engine να φορτώνει και εκκινεί εφαρμογές βασισμένες σε SANs, να παρακολουθεί την εκτέλεσή τους, να βλέπει τα περιεχόμενα των πλαισίων εκτέλεσης και να εκτελεί κάποιες βοηθητικές ή debug λειτουργίες, όπως χειρωνακτική αποστολή γεγονότων (manual publish), επανεκκίνηση του SAN engine, εκτέλεση προσομοιώσεων (simulations) με φόρτωση και αποστολή γεγονότων από αρχείο καταγραφής (παλαιότερων) γεγονότων και άλλα.

Η μονάδα διεπαφής με τον χρήστη μπορεί να έχει διάφορες μορφές ανάλογα με το που θα χρειαστεί να χρησιμοποιηθεί ο SAN engine. Για παράδειγμα μπορεί να είναι ένα γραφικό περιβάλλον διαχείρισης του SAN engine, ή μια προγραμματιστική διεπαφή με έναν Application Server (π.χ. Tomcat) έτσι ώστε να είναι δυνατός ο έλεγχος του SAN engine από το Web. Η CLI μπορεί ακόμη να υλοποιηθεί για να λειτουργεί από τη γραμμή εντολών του λειτουργικού συστήματος του υπολογιστή. Εξάλλου το όνομα CLI έχει παραμείνει απ' την πρώτη υλοποίηση της διεπαφής που ήταν για τη γραμμή εντολών.

- **Μονάδα αποτίμησης εκφράσεων ή Expression Evaluation Unit (EEU).** Σκοπός της είναι να αποτιμά, δηλαδή να υπολογίζει τις τιμές των εκφράσεων που της δίνονται, χρησιμοποιώντας δεδομένα από διάφορες πηγές, όπως τα πλαίσια εκτέλεσης, το αρχείο διαμόρφωσης, αλλά και εξωτερικές εφαρμογές και υπηρεσίες.

Αναλυτικότερα, κατά την εκτέλεση των στιγμιότυπων των SANs συναντώνται κόμβοι που απαιτούν την αποτίμηση κάποιων εκφράσεων προκειμένου να επιτελέσουν τη λειτουργία τους. Για παράδειγμα, οι κόμβοι συνθήκης πλαισίου απαιτούν την αποτίμηση μίας λογικής (Boolean, true/false) συνθήκης ελέγχου προκειμένου να επιτρέψουν την εκτέλεση του επόμενων κόμβων ενέργειας. Επίσης, οι κόμβοι-κατάστασης μπορεί να περιέχουν παραμετρικούς ορισμούς των καταστάσεων που τους ενδιαφέρουν καθώς και των αντίστοιχων CEPats, οι οποίοι ορισμοί συγκεκριμενοποιούνται κατά τη στιγμή της εκτέλεσης των κόμβων. Επίσης και αρκετοί άλλοι κόμβοι απαιτούν την αποτίμηση εκφράσεων (π.χ. loop και condition decorators, οι κόμβοι σύνθετων ενεργειών κλπ).

Η EEU δεν υποστηρίζει κάποια συγκεκριμένη γλώσσα έκφρασης και αποτίμησης των εκφράσεων. Αντιθέτως, αρχικοποιεί και διαχειρίζεται μια συλλογή από

Αποτιμητές (Evaluators), οι οποίοι είναι οι υπομονάδες που υλοποιούν και παρέχουν συγκεκριμένες γλώσσες εκφράσεων ο καθένας τους. Στην υλοποίησή μας παρέχονται δύο τέτοιοι αποτιμητές. Ο ένας αποτιμά εκφράσεις εκτελώντας μια στατική (static) μέθοδο μιας κλάσης Java, ενώ ο δεύτερος αποτιμά εκφράσεις και προγράμματα γραμμένα σε JavaScript. Ο ρόλος της EEU είναι να βρίσκει τον κατάλληλο αποτιμητή για κάθε έκφραση που της δίνεται και κατόπιν να του αναθέτει (delegate) την εργασία αυτή. Η EEU και οι αποτιμητές αποτελούν μέρος της βιβλιοθήκης των SANs (SAN Library).

- **Μονάδα εκτέλεσης απλών ενεργειών ή Action Execution Unit (AEU).** Σκοπός της είναι να εκτελεί τις απλές ενέργειες των SANs, τις οποίες της δίνει η μονάδα εκτέλεσης (ETU). Μπορεί να αποτελεί μέρος του SAN engine (SAN Library) ή να είναι κάποιο ξεχωριστό λογισμικό. Η λειτουργία αυτής της μονάδας ξεφεύγει από τα ενδιαφέροντα αυτής της διατριβής.
- **Διάφορες λειτουργικές ευκολίες ή Utilities.** Πρόκειται για μια βιβλιοθήκη με επαναχρησιμοποιήσιμα κομμάτια κώδικα για διάφορες συχνά χρησιμοποιούμενες εργασίες. Για παράδειγμα ανάγνωση και εγγραφή αρχείων, επικοινωνία μέσω πρωτοκόλλου HTTP, δημιουργία μοναδικών αναγνωριστικών (GUID), εμφάνιση πλαισίων διαλόγων (message, confirm, prompt), ευκολότερη πρόσβαση στο context, ευκολότερη διαχείριση SANs και οντοτήτων, μορφοποίηση ημερομηνιών (π.χ. με το W3C format), μορφοποίηση δεδομένων σε XML, καταγραφή στατιστικών δεδομένων, επεξεργασία προτύπων (templates) και συμπλήρωση των placeholders με συγκεκριμένα δεδομένα και άλλα.

6.2 Υλοποίηση του Συστήματος Επίγνωσης Κατάστασης (SAS)

Στο πλαίσιο της παρούσας διατριβής αναπτύχθηκε μια υλοποίηση του SAS, βασισμένη στην δημοφιλή γλώσσα προγραμματισμού Java™. Η εκτέλεση των SANs γίνεται σε ξεχωριστό νήμα εκτέλεσης (dedicated thread) για κάθε στιγμιότυπο, ενώ τα πλαίσια εκτέλεσης αποθηκεύονται στη μνήμη (in-memory context management). Ως αποθετήριο SANs έχει χρησιμοποιηθεί το Sesame, ένα αρκετά δημοφιλές RDF repository που παρέχει τη δυνατότητα ερωτήσεων (queries) με τη γλώσσα SPARQL.

Στην υλοποίηση αυτή δεν έχουν συμπεριληφθεί ο δίαυλος γεγονότων και ο CEP engine. Ωστόσο έχουν υλοποιηθεί διεπαφές για δύο διαύλους γεγονότων τον “Distributed Service Bus (DSB)” και Gama System PubSub, και για έναν CEP engine τον “Distributed CEP (DCEP)” (Anicic et al., 2012a).

Έχουν ακόμη υλοποιηθεί δύο αποτιμητές εκφράσεων. Ο πρώτος εκτελεί μια στατική μέθοδο (static method) μιας κλάσης Java, προκειμένου αυτή να πραγματοποιήσει την αποτίμηση. Η κλάση και η μέθοδος, όπως και το πλαίσιο εκτέλεσης, δίνονται στον αποτιμητή. Ο δεύτερος εκτελεί κώδικα JavaScript που αποτιμά την έκφραση. Επίσης μπορεί να εκτελέσει και μικρά προγράμματα, τα οποία έχουν πρόσβαση σε όλες τις λειτουργικές ευκολίες (utilities) του SAN engine. Συνεπώς μπορούν να διαβάζουν και γράφουν αρχεία, έχουν πρόσβαση στο πλαίσιο εκτέλεσης, μπορούν να στέλνουν γεγονότα, να επικοινωνούν

με web services, να εμφανίζουν πλαίσια διαλόγου στον χρήστη (dialog boxes – alert, confirm, prompt) και άλλα. Τέλος αναφέρουμε ότι δεν έχει υλοποιηθεί κάποιος ιδιαίτερος μηχανισμός εκτέλεσης των απλών ενεργειών, αφού με κατάλληλη χρήση των αποτιμητών στους κόμβους απλών ενεργειών μπορεί να γραφεί σχεδόν οποιοδήποτε πρόγραμμα που εκτελεί οποιαδήποτε λειτουργία. Υπενθυμίζουμε ότι οι κόμβοι απλών ενεργειών μπορεί να συμπεριλαμβάνουν μια έκφραση προς αποτίμηση.

```

C:\windows\system32\cmd.exe - bin\unt.bat
:umpCapacity "" ;
:Grain "" ;
:Destination "QINGDAO" ;
:MMSI "355443000" ;
:satTelx "" ;
t:endTime "2012-02-19T13:28:43.786+0200"^^<http://www.w3.org/2001/XMLSchema#dateTime> ;
t:stream "http://streams.event-processing.org/ids/VesselStream#stream" .

-----
Using namespaces: prefix : <http://imu.ntua.gr/ais#> prefix t: <http://events.event-processing.org/types/> prefix s: <http://imu.ntua.gr/ais#> prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> prefix wsnt: <http://docs.oasis-open.org/wsn/b-2>
RDFContextualizer: addToContext: Adding to LOCAL context:
  latLon = [27.27225122,3632],
  event=CEPAT for situation: Vessel Event received#0,
  node=null
RDFContextualizer: addToContext: Adding to LOCAL context:
  tm = [2012-02-19T13:28:43.786+0200],
  event=CEPAT for situation: Vessel Event received#0,
  node=null
RDFContextualizer: addToContext: Adding to LOCAL context:
  name = [APL ZEEBRUGGE],
  event=CEPAT for situation: Vessel Event received#0,
  node=null
RDFContextualizer: addToContext: Adding to LOCAL context:
  type = [containership],
  event=CEPAT for situation: Vessel Event received#0,
  node=null
RDFContextualizer: addToContext: Adding to LOCAL context:
  uri = [http://events.event-processing.org/ids/VesselStream#stream],
  event=CEPAT for situation: Vessel Event received#0,
  node=null
RDFContextualizer: addToContext: Adding to LOCAL context:
  area = [China Sea],
  event=CEPAT for situation: Vessel Event received#0,
  node=null
RDFContextualizer: addToContext: Adding to LOCAL context:
  speed = [15],
  event=CEPAT for situation: Vessel Event received#0,
  node=null
RDFContextualizer: addToContext: Adding to LOCAL context:
  mmsi = [355443000],
  event=CEPAT for situation: Vessel Event received#0,
  node=null
Spawning new thread for 'Vessel Moved'
Root Goal 'Vessel Moved' starts iteration #121
Root Situation 'Vessel Event received' waits for Event
DsbCEPENGINE: waitForEvent: waiting on CEPAT 'CEPAT for situation: vessel Event received#0 ': SAN execution blocks.
Running Goal 'Vessel Moved'
Checking context-condition 'check vessel speed' :
js: cc-0: Vessel APL ZEEBRUGGE (355443000) in jurisdiction
js: cc-0: Vessel APL ZEEBRUGGE (355443000) is not monitored
js: cc-0: Vessel speed = 15 knots
false
Thread of Root Goal 'Vessel Moved' terminated
    
```

Εικόνα 6-1. Στιγμιότυπο από τη λειτουργία του SAN engine – CLI

Στη συνέχεια παρουσιάζεται συνοπτικά ο Επεξεργαστής SANs (SAN editor) με τον οποίο γίνεται η δημιουργία και επεξεργασία των SANs.

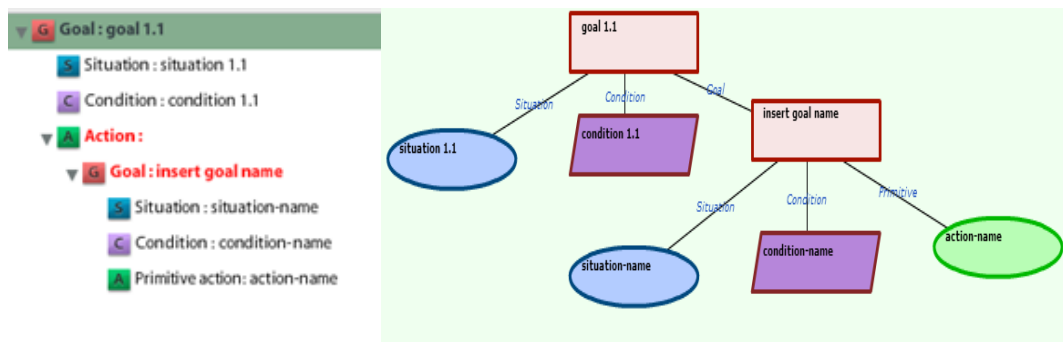
6.3 Επεξεργασία των SANs (SAN editor)

Η δημιουργία και επεξεργασία των SANs γίνεται με ένα εξειδικευμένο γραφικό περιβάλλον σχεδιασμού που ονομάζεται «Επεξεργαστής SANs» ή **SAN Editor**. Πρόκειται για μια web-based εφαρμογή που έχει αναπτυχθεί σε Adobe® Flash®. Ο SAN editor παρέχει στον σχεδιαστή των SANs όλα τα εκείνα τα εργαλεία που είναι αναγκαία για τη δημιουργία και επεξεργασία SANs, οντοτήτων (entities) αλλά και τον ορισμό των μετα-πληροφοριών των πλαισίων εκτέλεσης (execution contexts) κάθε SAN ή οντότητας ή του γενικού πλαισίου. Ο SAN editor παράγει αρχεία με τα SANs μιας εφαρμογής, τα οποία στη συνέχεια μπορούν να εκτελεστούν από τον SAN engine. Στην υλοποίηση του SAN editor, τα SANs κωδικοποιούνται με χρήση της γλώσσας RDF, συνήθως στη μορφή (format) RDF/N3⁴⁶. Σημειώνουμε τέλος ότι

⁴⁶ Η Notation-3 ή N3 πρόκειται για μια επέκταση του μοντέλου δεδομένων της RDF που προσθέτει τύπους, μεταβλητές, λογικά συμπεράσματα, ιδιότητες καθώς και ένα συντακτικό (RDF/N3) εναλλακτικό του συντακτικού της RDF (RDF/XML syntax). [Πηγή: <http://www.w3.org/TeamSubmission/n3/>]

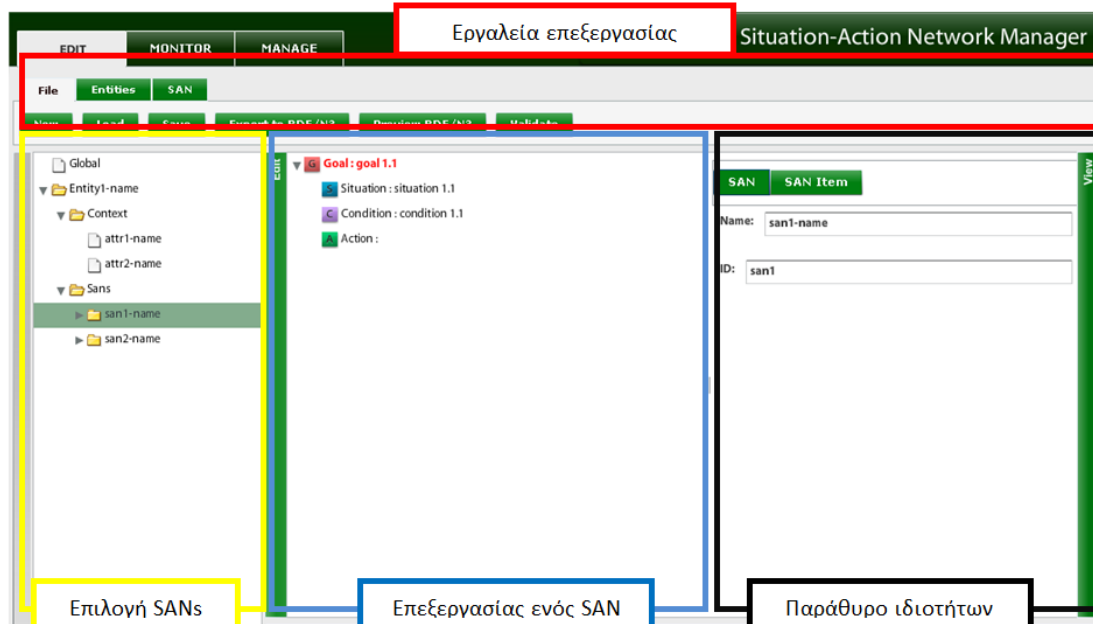
η δημιουργία και επεξεργασία των SANs μπορεί να γίνει και με χρήση απλού επεξεργαστή κειμένου, εργασία όμως γενικά δύσκολη και με μεγάλη πιθανότητα να συμβούν λάθη.

Η βασικότερη λειτουργία του SAN editor είναι η δημιουργία και επεξεργασία SANs μέσω ενός απλού στη χρήση γραφικού περιβάλλοντος. Τα SANs οπτικοποιούνται με δύο τρόπους, (α) ως μια δενδρική ιεραρχία από κόμβους ή (β) σύμφωνα με τον τρόπο αναπαράστασης των SANs που χρησιμοποιεί σχήματα και χρώματα για να υποδείξει τα διάφορα είδη κόμβων.

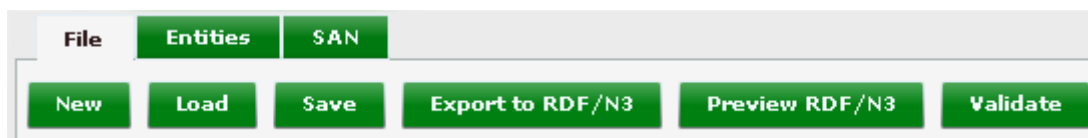


Εικόνα 6-2. Παραδείγματα των δύο οπτικοποιήσεων SANs που χρησιμοποιεί ο SAN editor

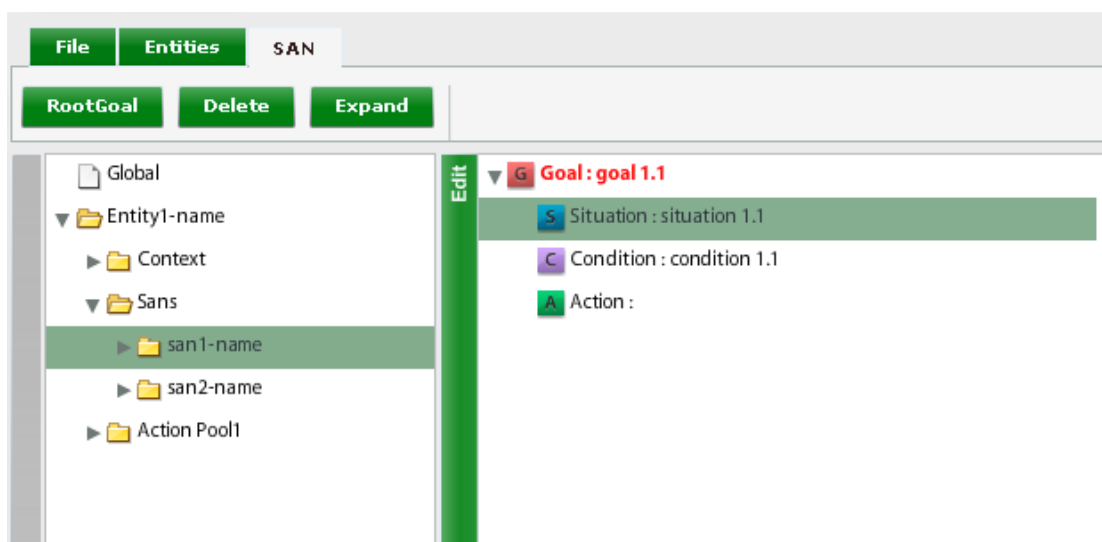
Με χρήση κατάλληλων εργαλείων (widgets) προστίθενται σε ένα SAN οι απαραίτητοι κόμβοι, ενώ στο παράθυρο επεξεργασίας των ιδιοτήτων καθορίζονται με λεπτομέρεια οι παράμετροι (ιδιότητες) του εκάστοτε επιλεγμένου κόμβου. Ιδιαίτερη μνεία αξίζει να γίνει στους κόμβους κατάσταση, όπου με διπλό κλικ επάνω τους ανοίγει το παράθυρο επεξεργασίας του contextualization. Εκεί μπορεί να προσδιοριστεί ο τύπος του contextualizer, το πλαίσιο εκτέλεσης (context) που θα δεχτεί τις πληροφορίες που θα εξαχθούν από το γεγονός καθώς επίσης και διάφορες παράμετροι της εξαγωγής. Στην υλοποίηση του SAN engine υποστηρίζονται δύο είδη contextualizers, ο RDF contextualizer που εξάγει μετα-πληροφορίες από γεγονότα κωδικοποιημένα σε RDF, και ο Name-Value Pairs (NVP) contextualizer που λαμβάνει από το περιεχόμενο (payload) των γεγονότων μια λίστα από ζεύγη ονομάτων – τιμών (name-value pairs), τα οποία και εισάγει αυτούσια στο πλαίσιο εκτέλεσης. Ο τρόπος λειτουργίας του δεύτερου είναι προφανής. Αντιθέτως, ο RDF contextualizer εκτελεί μια σειρά από πιο σύνθετες ενέργειες. Κατ' αρχήν φορτώνει το RDF περιεχόμενο του γεγονότος σε ένα προσωρινό RDF repository (και αυτό υλοποιείται με χρήση του Sesame). Έπειτα εκτελεί μια σειρά από ερωτήματα (queries) εκπεφρασμένα με τη γλώσσα SPARQL. Τα αποτελέσματα των ερωτημάτων έχουν τη μορφή πίνακα, όπου κάθε στήλη έχει ένα όνομα (name) και μία ή περισσότερες τιμές (values) που αντιστοιχούν στις γραμμές του πίνακα. Άρα στην ουσία λαμβάνεται μια σειρά από ζεύγη ονομάτων – τιμών (name-value pairs), τα οποία και εισάγονται στο πλαίσιο εκτέλεσης που προσδιορίζεται στον contextualizer. Τόσο το είδος του contextualizer και του πλαισίου εκτέλεσης, όσο και τα ερωτήματα σε SPARQL (για τον RDF contextualizer) ορίζονται στο παράθυρο επεξεργασίας του contextualization του SAN editor.



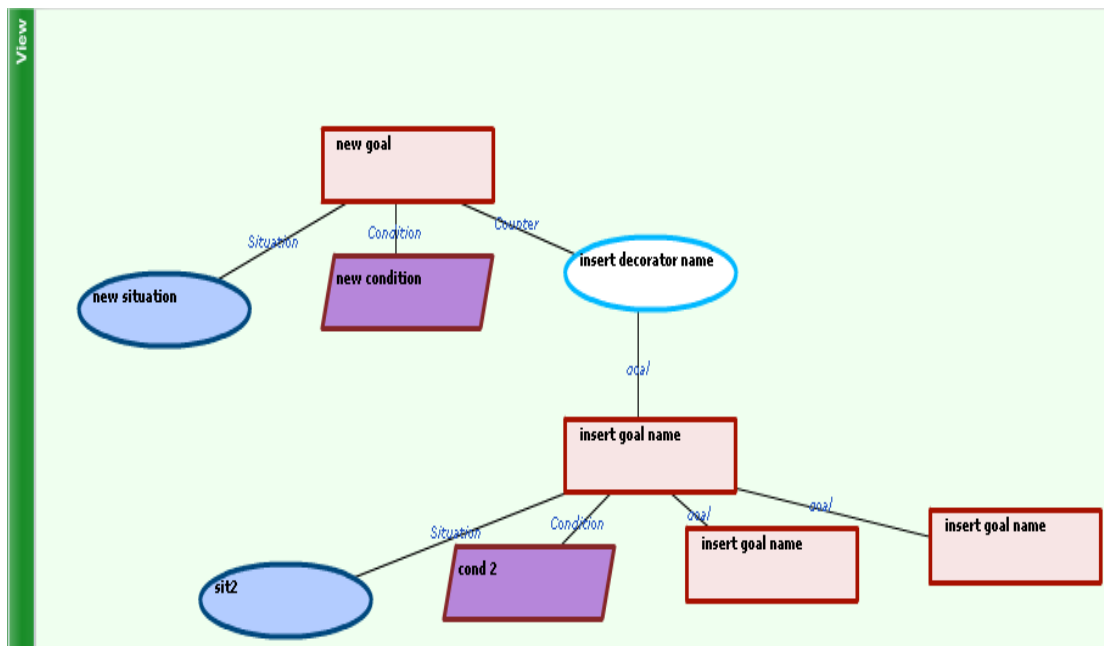
Εικόνα 6-3. Ο επεξεργαστής SANs (SAN editor)



Εικόνα 6-4. Τα εργαλεία επεξεργασίας των SANs



Εικόνα 6-5. Τα παράθυρα επιλογής (αριστερά) και επεξεργασίας των SANs (δεξιά)

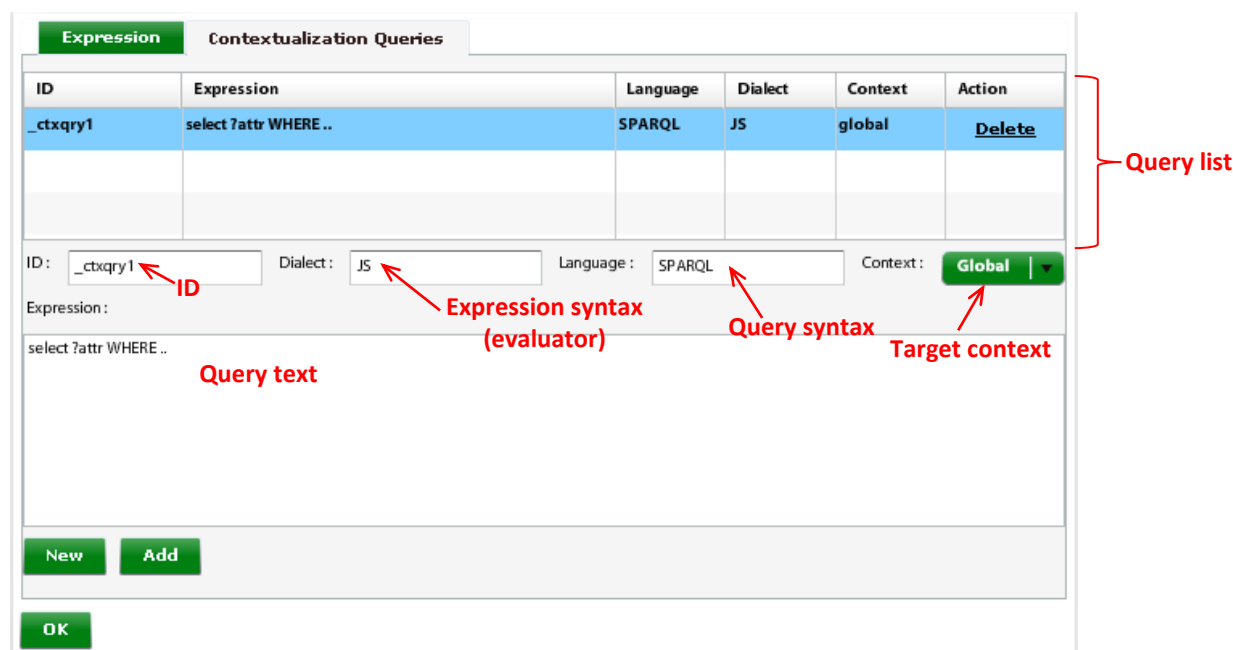


Εικόνα 6-6. Εναλλακτική απεικόνιση του επιλεγμένου SAN

SAN		SAN Item	
Parameter	Value	Parameter	Value
Id	_ng1	Id	_news1
Name	new goal	Name	new situation
		Dialect	EP-SPARQL
		Expression	

SAN		SAN Item	
Parameter	Value	Parameter	Value
Id	_decor1332592047531	Type	Select
Name	insert decorator name	Loop	
		Counter	
		Timer	
		Print	
		Break	

Εικόνα 6-7. Στιγμιότυπα του παραθύρου επεξεργασίας ιδιοτήτων

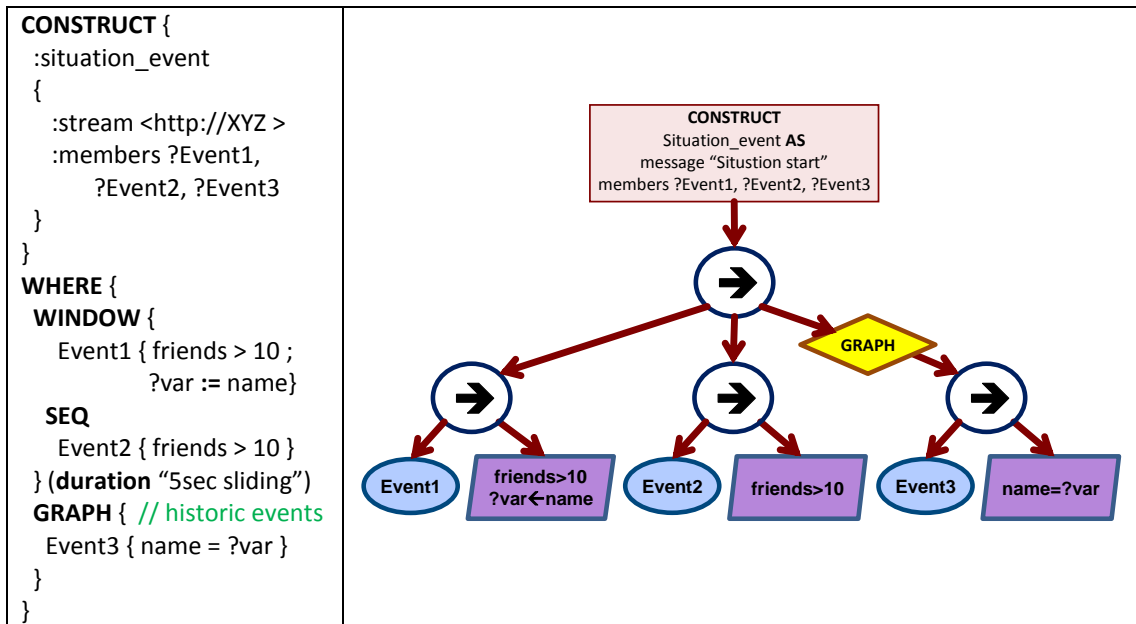


Εικόνα 6-8. Παράθυρο επεξεργασίας του contextualization του επιλεγμένου κόμβου-κατάστασης

6.4 Εναλλακτικές χρήσεις των SANs

Τα SANs έχουν αναπτυχθεί ως τρόπος μοντελοποίησης των συμπεριφορών συστημάτων. Ωστόσο θα μπορούσαν να έχουν και άλλες δευτερεύουσες χρήσεις μερικές απ' τις οποίες αναφέρουμε στην συνέχεια. Φυσικά δεν μπορούμε να ισχυριστούμε ότι τα SANs παρέχουν την βέλτιστη λύση για τις χρήσεις αυτές, αφού δεν σχεδιάστηκαν για να εξυπηρετήσουν τους σκοπούς τους.

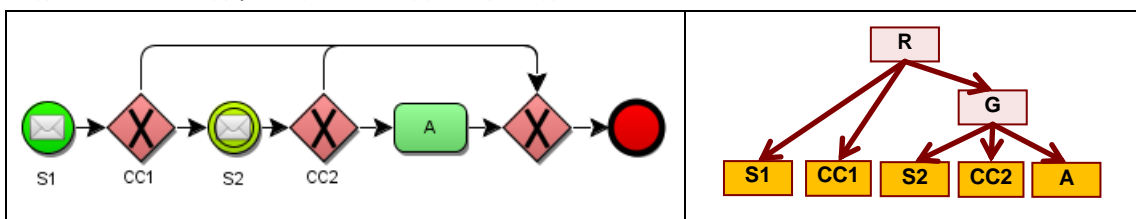
Η ιεραρχική δομή των SANs εκτός από την ανάλυση των στόχων σε υποστόχους και σε ενέργειες δίνει και την αλληλουχία, τη διαδοχή και την αλληλεξάρτηση των καταστάσεων που ενεργοποιούν τους κόμβους-στόχων (δείτε Σχήμα 5-4). Αν επεκτείνουμε λίγο την οπτική αυτή θα μπορούσαμε να χρησιμοποιήσουμε τα SANs ως μια τυπική γλώσσα ορισμού Προτύπων Σύνθετων Γεγονότων (CEPat). Η δομή του SAN αν «διαβαστεί» από το χαμηλότερο επίπεδο προς τα επάνω, θα μπορούσε να προδιαγράψει τα απλά γεγονότα που θα πρέπει να ληφθούν, τους ελέγχους που πρέπει να γίνουν σε αυτά, κι έπειτα τον τρόπο με τον οποίο αυτά θα συνδυαστούν σε σύνθετα γεγονότα (με χρήση κόμβων σύνθετων ενεργειών). Επίσης με τη χρήση decorators θα μπορούσαν να τεθούν χρονικοί, γεωγραφικοί ή άλλοι περιορισμοί, επαναληπτικές δομές και συνθήκες ελέγχου που είναι σημαντικές παράμετροι των CEPats. Η διαδικασία αυτή εφαρμοζόμενη σε όλα τα επίπεδα και για όλους τους ενδιάμεσους κόμβους, από το πιο κάτω επίπεδο (απλά γεγονότα) έως την κορυφή του SAN, θα μπορούσε να παράγει ολόένα και πιο σύνθετα γεγονότα. Φυσικά το σύνθετο γεγονός της κορυφής θα είναι το αποτέλεσμα του CEPat.



Εικόνα 6-9. Παράδειγμα CEPat σε EP-SPARQL και με σημειογραφία SANs

Μια παραλλαγή του παραπάνω θα μπορούσε να χρησιμοποιηθεί για την περιγραφή των καταστάσεων του περιβάλλοντος (situation modeling) όπου εκεί αντί για γεγονότα θα εξετάζονταν οι διάφορες παράμετροι του περιβάλλοντος.

Μια άλλη εναλλακτική χρήση των SANs θα μπορούσε να είναι ως γλώσσα μοντελοποίησης επιχειρησιακών διαδικασιών (business processes). Μάλιστα αυτή η χρήση θα μπορούσε να επεκταθεί και στην εκτέλεση αυτών. Εξάλλου έχουμε δει ότι τα SANs εμπεριέχουν ένα σχέδιο εκτέλεσης για την επίτευξη ενός στόχου.



Εικόνα 6-10. Παράδειγμα διαδικασίας σε BPMN και με σημειογραφία SANs

Φυσικά τα SANs δεν διαθέτουν τον πλούτο της σημειογραφίας και των εξειδικευμένων δομών που έχουν γλώσσες όπως η BPMN.

Στο επόμενο κεφάλαιο επιχειρείται μια αξιολόγηση των SANs με χρήση διαφόρων μέσων μελέτης αυτών (σενάριο χρήσης και συγκριτικά παραδείγματα). Επιπρόσθετα, εξετάζονται οι επιδόσεις της υλοποίησης του μηχανισμού των SANs μέσω μιας σειράς πειραμάτων.

7 Αξιολόγηση των Δικτύων Κατάστασης – Ενέργειας

Στο κεφάλαιο αυτό επιχειρείται μια αξιολόγηση των SANs μέσω ενός σεναρίου χρήσης. Το σενάριο αυτό εκτελέστηκε με πραγματικά δεδομένα προκειμένου να επιβεβαιωθεί η αποτελεσματικότητα των SANs ως μεθοδολογίας μοντελοποίησης και η καταλληλότητα της υλοποίησης του SAS. Στη συνέχεια του κεφαλαίου, δίνονται τα ευρήματα μιας σειράς πειραμάτων και μετρήσεων που αφορούν τη σύγκριση των SANs και της υλοποίησής τους, με το Drools, ένα γνωστό σύστημα κανόνων ECA.

7.1 Σενάριο Χρήσης με SANs

Στην παρούσα ενότητα δίνεται ένα πλήρες παράδειγμα μοντελοποίησης με χρήση των SANs. Το σενάριο του παραδείγματος αφορά την παρακολούθηση της κίνησης των σκαφών μιας θαλάσσιας περιοχής (Σινική θάλασσα) για την τήρηση των διεθνών κανόνων ασφαλούς ναυσιπλοΐας. Πιο συγκεκριμένα επιτρέπει στις λιμενικές αρχές της περιοχής να ελέγχουν αν εφαρμόζονται οι Διεθνείς Κανονισμοί Αποφυγής Συγκρούσεων στη Θάλασσα, στην περιοχή ευθύνης τους. Το σενάριο εστιάζει στον εντοπισμό δύο σοβαρών καταστάσεων που αφορούν ταχύπλοα σκάφη (high-speed boats) και είναι (α) η πλεύση με μεγάλη ταχύτητα ενώ επικρατούν θυελλώδεις συνθήκες στην περιοχή, και (β) η μη ελάττωση της ταχύτητας όταν παραπλέουν σε κοντινή απόσταση μικρότερα πλοία ή άλλα ταχύπλοα σκάφη.

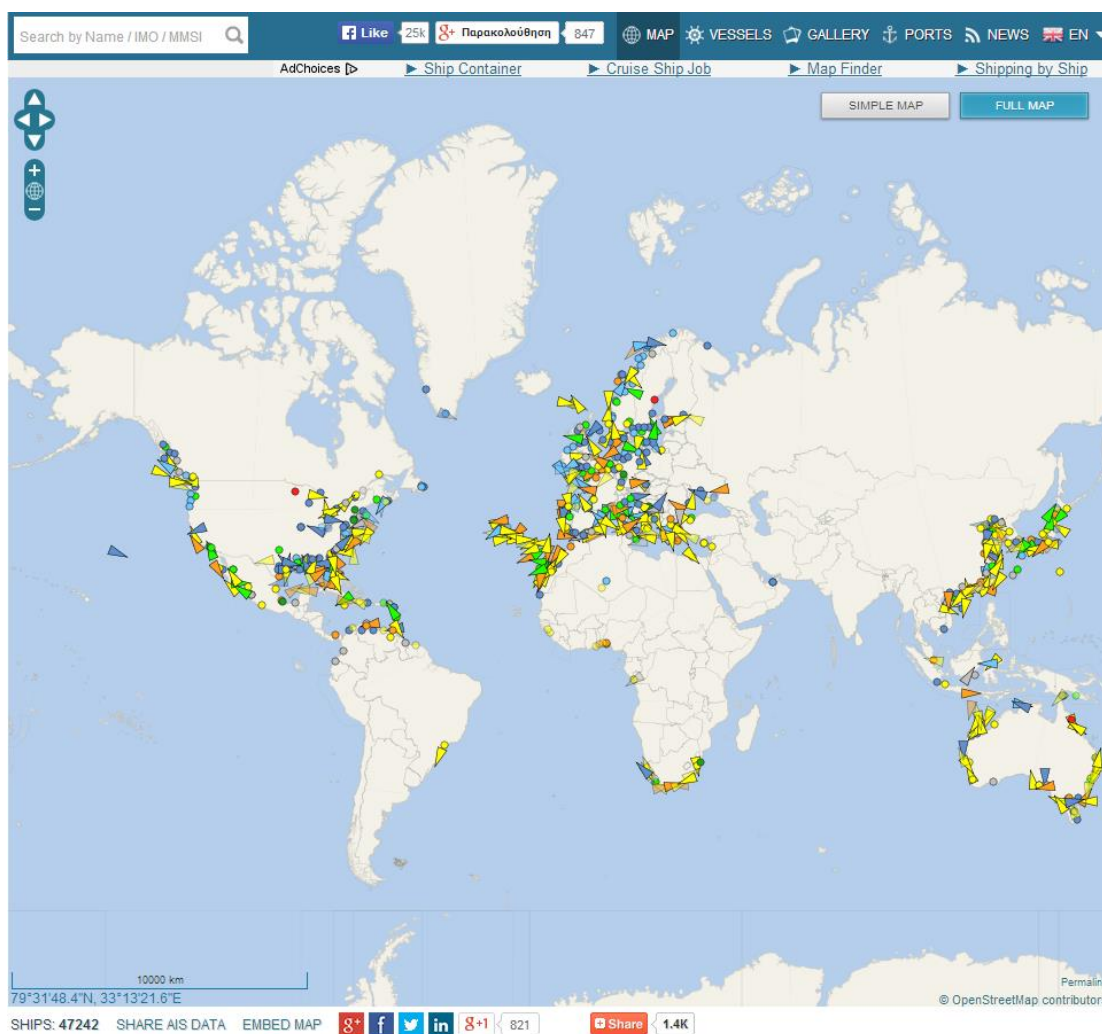
Το συγκεκριμένο παράδειγμα είναι κατάλληλο για την αξιολόγηση των SANs τόσο ως γλώσσα μοντελοποίησης όσο και ως υλοποίηση καθότι περιλαμβάνει ένα ρεαλιστικό πρόβλημα προς μοντελοποίηση αλλά και ένα μεγάλο όγκο πραγματικών δεδομένων προς επεξεργασία (ως γεγονότα). Επίσης απαιτεί τον συνδυασμό των γεγονότων αυτών προκειμένου να επιτευχθεί ο σκοπός του παραδείγματος, ο οποίος είναι ο εντοπισμός των δύο τύπων παραβάσεων που προαναφέρθηκαν.

Το σενάριο εκτελέστηκε με πραγματικά δεδομένα πλοίων που ληφθήκαν από το AISHub portal⁴⁷, μια δικτυακή πύλη που συλλέγει και παρέχει μέσω του Διαδικτύου πληροφορίες για την ταυτότητα, θέση και πορεία των πλοίων καθώς και μετεωρολογικά δεδομένα. Τα

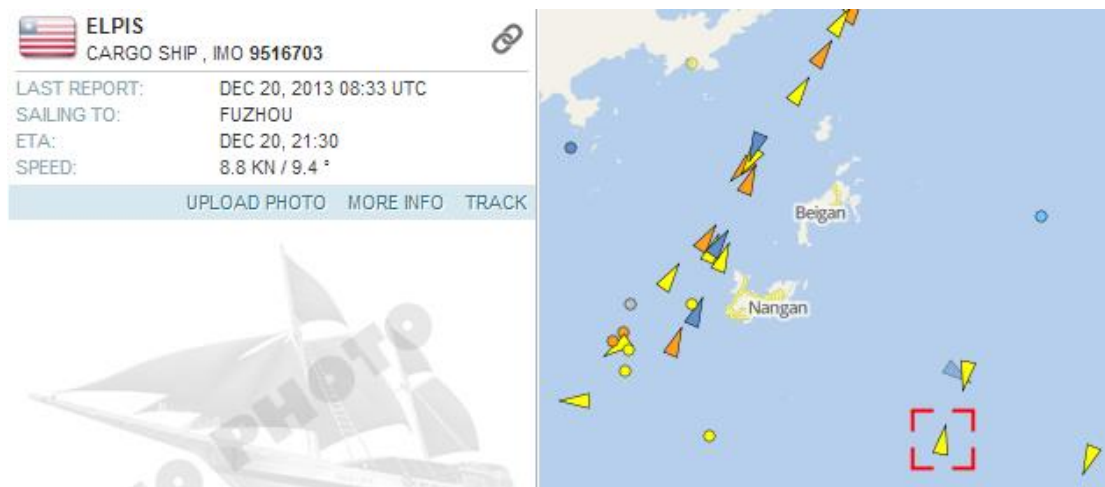
⁴⁷ <http://www.aishub.net/>

στοιχεία που συλλέγει το AISHub μεταδίδονται από τις συσκευές AIS που υποχρεωτικά φέρουν όλα τα πλοία σήμερα και λαμβάνονται μέσω δεκτών VHF.

Στις επόμενες εικόνες ο αναγνώστης μπορεί να δει με ποιον τρόπο παρουσιάζονται στο AISHub portal οι διάφορες πληροφορίες. Στην Εικόνα 7-1 δίνεται μια γενική άποψη της κίνησης των πλοίων σε παγκόσμιο επίπεδο. Στην Εικόνα 7-3 φαίνεται η αυξημένη κυκλοφορία σε μέρος της Σινικής θάλασσας. Επιλέγοντας ένα οποιοδήποτε πλοίο επάνω στο χάρτη, εμφανίζονται πληροφορίες για την τρέχουσα θέση, την ταχύτητα και την πορεία του (Εικόνα 7-2).



Εικόνα 7-1. Οθόνη του AISHub.net



Εικόνα 7-2. Τρέχουσες πληροφορίες ενός πλοίου



Εικόνα 7-3. Κυκλοφορία σε περιοχή της Σινικής θάλασσας

Ακόμη μπορεί να ληφθούν πληροφορίες για το γεωγραφικό μήκος και γεωγραφικό πλάτος του πλοίου καθώς και σε ορισμένες περιπτώσεις μετεωρολογικά δεδομένα της τρέχουσας θέσης του πλοίου, όπως η ταχύτητα και η κατεύθυνση του ανέμου και η θερμοκρασία. Παρέχονται ακόμη και ορισμένες πληροφορίες προστιθέμενης αξίας όπως ιστορικά δεδομένα (π.χ. πρόσφατες αναχωρήσεις ή αφίξεις σε λιμάνια, ιστορικό δρομολογίων και προηγούμενες θέσεις) (Εικόνα 7-4) καθώς και στοιχεία που αφορούν παραπλέοντα πλοία (Εικόνα 7-5).

PUMA (IMO: 9078048)
✕

FLAG	🇩🇰 DENMARK		
SHIP TYPE	GENERAL CARGO SHIP	BUILT	1994
GT	1398	DWT	2120

MMSI	219574000	CALLSIGN	OZDV2
SIZE	73 / 11 M	DRAUGHT	3.5 M


DESTINATION	BAR	ETA	OCT 01, 11:00
-------------	-----	-----	---------------

LATITUDE	22.20799 N	COURSE	118.3 °
LONGITUDE	120.39046 E	SPEED	0.9 KN

LAST FIVE PORT CALLS (DETECTED BY AIS)

OCT 10, 2013, 01:03 UTC	PORT SAID	🇪🇬 EGYPT
SEP 20, 2013, 16:02 UTC	SANTANDER	🇪🇸 SPAIN
SEP 18, 2013, 07:04 UTC	CHERBOURG	🇫🇷 FRANCE
SEP 13, 2013, 13:04 UTC	PAPENBURG	🇩🇪 GERMANY
AUG 29, 2013, 15:01 UTC	VARBERG	🇸🇪 SWEDEN

SHIP DETAILS



COPYRIGHT HOLDER:
DATE:
PLACE:
UPLOADED:

UPLOAD PHOTO

Εικόνα 7-4. Λεπτομερείς πληροφορίες πλοίου με προηγούμενους προορισμούς

'CASTELLA SQUARE' - Nearby Vessels: 50 Records Found

Vessel's Name		Vessel's Type	Speed	Course	Current Port	Distance to...	Bearing	Course Intersection in...
🚢 CASTELLA SQUARE	Show on Map	High Speed Craft	39.8	76		0	On Board	
🚢 GUANG HUI 19	Show on Map	Tanker	5.6	127		0.94	188	
🚢 900004362	Show on Map		7.3	80		1.52	72	2 minutes (2.68 n.miles) Subject vessel passes ahead
🚢 SHEN ZHOU 17	Show on Map	Cargo	5.7	74		1.53	69	
🚢 HAI YANG SHI YOU 623	Show on Map	Tug	6.9	128		1.66	155	
🚢 NANHAI211	Show on Map	Tug	8.6	127		1.73	145	
🚢 NEW FERRY 83	Show on Map	High Speed Craft	40.2	93		1.93	277	-2 minutes (0.45 n.miles) Possible collision
🚢 XING LONG 3	Show on Map	Cargo	9.6	342		2.14	332	
🚢 ONWA93	Show on Map	Pleasure Craft	0.1	0		2.97	212	
🚢 ONWA1087	Show on Map	Unspecified	0	0		2.98	212	
🚢 SPACEON_GRACE	Show on Map	Unspecified	0	0		3.03	213	
🚢 800002480	Show on Map		0	295		3.03	213	
🚢 ONWA740	Show on Map	Cargo	0	0		3.03	212	
🚢 ONWA53	Show on Map	Pleasure Craft	0	0		3.04	213	
🚢 ONWA81	Show on Map	Unspecified	0	0		3.04	213	
🚢 ONWA1071	Show on Map	Unspecified	0	0		3.04	213	

Εικόνα 7-5. Παραπλέοντα πλοία

7.1.1 Σενάριο του παραδείγματος

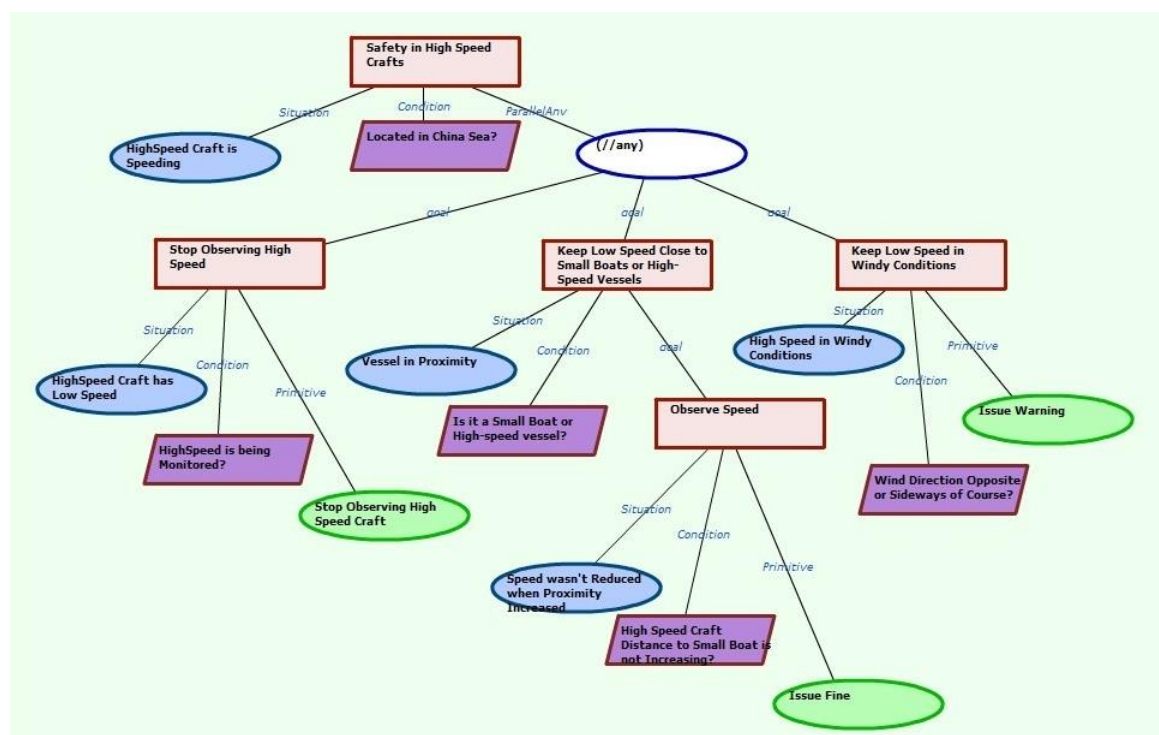
Για τους σκοπούς του παραδείγματος συλλέχθηκαν από το AISHub portal, σε τακτά χρονικά διαστήματα, δεδομένα 3 ημερών για όλα τα πλοία που έπλεαν στη Σινική θάλασσα. Στη συνέχεια μορφοποιήθηκαν σε κατάλληλα γεγονότα και προωθήθηκαν υπό τη μορφή δύο ροών γεγονότων.

- Ροή γεγονότων πλοίων. Τα γεγονότα αυτής της ροής φέρουν πληροφορίες όπως το όνομα, ο τύπος, η ταυτότητα (αριθμός MMSI), η γεωγραφική θέση, η ταχύτητα, η πορεία του πλοίου, καθώς και η ταχύτητα και η κατεύθυνση του ανέμου (όταν είναι διαθέσιμα).
- Ροή γεγονότων εγγύτητας. Τα γεγονότα αυτής της ροής δίνουν συνοπτικές πληροφορίες για παραπλέοντα πλοία, όπως η γεωγραφική θέση, η κατεύθυνσή τους και η μεταξύ τους απόσταση.

Parameter	Value
Id	_Wind_Direction_Opposi
Name	Wind Direction Opposite
Dialect	default
Expression	function check_wind() {

Εικόνα 7-6. Το SAN για την ασφαλή πλεύση με μεγάλες ταχύτητες (1/2)

Για τον έλεγχο εφαρμογής των Διεθνών Κανονισμών Αποφυγής Συγκρούσεων στη Θάλασσα, και συγκεκριμένα για εντοπισμό των δύο τύπων παραβάσεων που προαναφέρθηκαν, χρησιμοποιήθηκε το SAN που φαίνεται στην Εικόνα 7-6 και στην Εικόνα 7-7. Για την αποτύπωσή του χρησιμοποιήθηκε ο επεξεργαστής των SANs.



Εικόνα 7-7. Το SAN για την ασφαλή πλεύση με μεγάλες ταχύτητες (2/2)

Προκειμένου να επιτύχει τον αρχικό του στόχο «Ασφαλής πλεύση ταχύπλων σκαφών» (Safety in High-Speed Crafts), το SAN ενεργοποιεί στον CEP engine ένα πρότυπο σύνθετων γεγονότων (CEPat) που αντιστοιχεί στην κατάσταση όταν ένα «ταχύπλοο πλοίο επιταχύνει» (High-Speed Craft is Speeding), για παράδειγμα πάνω από 15 κόμβους. Ο CEP engine, σύμφωνα με το CEPat, παρακολουθεί τη ροή γεγονότων πλοίων έτσι ώστε να εντοπίσει όσα ταχύπλοα κινούνται πάνω από το όριο ταχύτητας που έχει τεθεί. Σε μια τέτοια περίπτωση δημιουργεί και στέλνει ένα σύνθετο γεγονός που αφορά το συγκεκριμένο πλοίο, το οποίο γεγονός λαμβάνει ο SAN engine. Αυτός με τη σειρά του, σύμφωνα με τον αλγόριθμο εκτέλεσης των SANs, δημιουργεί ένα νέο στιγμιότυπο του SAN για να παρακολουθήσει το συγκεκριμένο πλοίο. Αν φυσικά υπάρχουν περισσότερα από ένα ταχύπλοα πλοία που πλέουν πάνω από το όριο ταχύτητας τότε δημιουργούνται ισάριθμα στιγμιότυπα του SAN. Επίσης, τα γεγονότα κατάστασης που εκκινούν τη διαδικασία, γίνονται contextualize στα τοπικά πλαίσια εκτέλεσης (local contexts) των αντίστοιχων στιγμιότυπων. Οι πληροφορίες πλαισίου (context elements) που εξάγονται από τα γεγονότα αφορούν το όνομα του πλοίου, τον αριθμό MMSI, τη θέση, την ταχύτητα, την κατεύθυνση, τον τύπο του καθώς και στοιχεία για τον άνεμο που πνέει εκείνη τη στιγμή.

Στη συνέχεια κάθε στιγμιότυπο συνεχίζει τη διάσχιση του SAN από τον κόμβο-κατάστασης το «ταχύπλοο πλοίο επιταχύνει». Έπειτα, καθότι ο συγκεκριμένος κόμβος ανήκει στον αρχικό στόχο, το αντίστοιχο CEPat δεν απενεργοποιείται, αλλά συνεχίζει η διάσχιση στον επόμενο κόμβο-συνθήκης πλαισίου, δηλαδή τον «Πλέει στη Σινική Θάλασσα» (Located in China Sea). Εκεί ελέγχεται αν το συγκεκριμένο πλοίο ήδη παρακολουθείται από κάποιο άλλο στιγμιότυπο κι επίσης αν βρίσκεται εντός της περιοχής ευθύνης της λιμενικής αρχής.

Για την πραγματοποίηση του δεύτερου ελέγχου μπορεί να χρησιμοποιηθεί ένα εξωτερικό Web Service⁴⁸ που παρέχεται από τη λιμενική αρχή. Αν το πλοίο δε βρίσκεται εντός δικαιοδοσίας ή αν ήδη παρακολουθείται τότε ο κόμβος-συνθήκης ολοκληρώνει τη λειτουργία του με αποτυχία και διαδοχικά αποτυγχάνει και ο στόχος κορυφής, οπότε το στιγμιότυπο ολοκληρώνει την εκτέλεσή του. Αν όμως το πλοίο βρίσκεται εντός δικαιοδοσίας και δεν παρακολουθείται ήδη τότε η εκτέλεση του SAN συνεχίζεται ως ακολούθως. Η πληροφορία ότι το τρέχον στιγμιότυπο άρχισε να παρακολουθεί το εν λόγω πλοίο, καταχωρείται στο πλαίσιο εκτέλεσης της οντότητας της λιμενικής αρχής. Έτσι τα επόμενα στιγμιότυπα θα μπορούν να το γνωρίζουν αυτό και να μη συνεχίζουν στην παρακολούθηση του ίδιου πλοίου.

Κατόπιν εκτελείται ο κόμβος σύνθετης ενέργειας που έπεται, ο οποίος είναι κόμβος παράλληλης εκτέλεσης ενεργειών τύπου ANY, δηλαδή αναμένει την επιτυχή εκτέλεση ενός από τους υποστόχους-παιδιά του προκειμένου να επιτύχει και αυτός. Κάτω από τον κόμβο παράλληλης εκτέλεσης υπάρχουν τρία κλαδιά (που εκτελούνται παράλληλα).

1. Ο στόχος «ελάττωση ταχύτητας σε θυελλώδεις συνθήκες» (Keep Low Speed in Windy Conditions) που αποσκοπεί στο να συστήσει στο πλοίο να ελαττώσει την ταχύτητά του λόγω των θυελλωδών καιρικών συνθηκών που επικρατούν στην περιοχή.
2. Ο στόχος «ελάττωση ταχύτητας λόγω εγγύτητας σε μικρότερο ή ταχύπλοο σκάφος» (Keep Low Speed Close to Small Boats or High-Speed Vessels) που αποσκοπεί στην πρόληψη συγκρούσεων.
3. Ο στόχος «τερματισμός παρακολούθησης» (Stop Observing High-Speed) που χρησιμοποιείται για να τερματίσει την παρακολούθηση του πλοίου όταν αυτό ελαττώσει την ταχύτητά του κάτω από το όριο ταχύτητας.

Αυτοί οι τρεις στόχοι ενεργοποιούνται αντιστοίχως από τρεις καταστάσεις ενδιαφέροντος. Κάθε μία από αυτές καταχωρεί στον CEP engine κι από ένα πρότυπο σύνθετου γεγονότος που την περιγράφει. Επίσης κάθε πρότυπο προσδιορίζει τα γεγονότα ποιου πλοίου πρέπει να χρησιμοποιηθούν.

1. Η κατάσταση «μεγάλη ταχύτητα σε θυελλώδεις συνθήκες» (High Speed in Windy Conditions) εξετάζει αν το πλοίο κινείται πάνω από το όριο ταχύτητας τα τελευταία 20 λεπτά (άρα χρήση και ιστορικών δεδομένων για την ταχύτητα) και ταυτόχρονα η ταχύτητα του ανέμου είναι πάνω από 18 κόμβους. Ο CEP engine εγγράφεται στη ροή γεγονότων πλοίων και ζητά να λάβει τόσο τρέχοντα όσο και ιστορικά γεγονότα, ενώ τα φιλτράρει με βάση την ταχύτητα του ανέμου.
2. Η κατάσταση «άλλο πλοίο σε κοντινή απόσταση» (Vessel in Proximity) εξετάζει αν υπάρχει κάποιο άλλο πλοίο στην περιοχή (π.χ. σε απόσταση μικρότερη των 5 ναυτικών μιλίων). Ο CEP engine εγγράφεται στη ροή γεγονότων εγγύτητας και τα φιλτράρει με βάση τη σχετική απόσταση των πλοίων.

⁴⁸ Το Web Service λαμβάνει τις ακριβείς συντεταγμένες του πλοίου και απαντά αν αυτές είναι εντός ή εκτός δικαιοδοσίας. Η χρήση του Web Service δεν είναι υποχρεωτική αλλά ενδεικτική των δυνατοτήτων που παρέχουν τα SANs.

3. Τέλος η κατάσταση το «πλοίο επιβράδυνε» (High-Speed Craft has Low Speed) ελέγχει αν το πλοίο ελάττωσε την ταχύτητά του. Ο CEP engine εγγράφεται στη ροή γεγονότων πλοίων και εξετάζει την ταχύτητα του πλοίου.

Οι καταστάσεις αυτές έχουν νόημα μόνο όταν ένα πλοίο κινείται πάνω από το όριο ταχύτητας. Εδώ δηλαδή φαίνεται και η ιεραρχία των καταστάσεων.



Σχήμα 7-1. Ιεραρχία καταστάσεων του SAN στην Εικόνα 7-7

Η συνέχεια της εκτέλεσης του SAN εξαρτάται από το ποιο σύνθετο γεγονός, ποιας κατάστασης θα έρθει πρώτο. Θα εξετάσουμε και τις τρεις περιπτώσεις. Σημειώνουμε ότι όταν ληφθεί ένα γεγονός σε έναν από τους κόμβους-καταστάσεων, τότε αυτός πρέπει να απενεργοποιήσει το αντίστοιχο πρότυπο στον CEP engine, καθώς δε χρειάζεται πλέον. Με τη σειρά του και ο CEP engine διαγράφεται από την αντίστοιχη ροή γεγονότων.

1. Αν το πλοίο πλέει με μεγάλη ταχύτητα σε θυελλώδεις συνθήκες τότε λαμβάνεται το γεγονός αρχής της κατάστασης «Μεγάλη ταχύτητα σε θυελλώδεις συνθήκες». Κατόπιν στη συνθήκη πλαισίου (Wind Direction Opposite or Sideways to Course) εξετάζεται αν επιπλέον η κατεύθυνση του ανέμου είναι αντίθετη, πλάγια ή πλαγιομετωπική με αυτήν του πλοίου (π.χ. $45^\circ - 315^\circ$). Τότε εκτελείται η ενέργεια «Αποστολή Ειδοποίησης» (Issue Warning) προς το πλοίο για να ελαττώσει ταχύτητα. Εκεί ολοκληρώνεται και η διάσχιση του συγκεκριμένου στιγμιότυπου του SAN με επιτυχία. Φυσικά αν το πλοίο δε συμμορφωθεί, τότε σε μετέπειτα χρόνο θα επαναληφθεί ο ίδιος κύκλος και θα σταλεί και νέα ειδοποίηση, συμπεριφορά που όμως είναι επιθυμητή.
2. Αν το πλοίο πλέει με μεγάλη ταχύτητα κοντά σε άλλο πλοίο (π.χ. μικρότερη των 5 ναυτικών μιλίων) τότε εξετάζεται στον κόμβο συνθήκης πλαισίου (Is it a Small Boat or High-Speed Vessel?) αν το παραπλέον πλοίο είναι μικρό σκάφος ή ταχύπλοο. Αν όντως είναι θα συνεχίσει η παρακολούθηση για ένα μικρό χρονικό διάστημα προκειμένου να διαπιστωθεί αν τα πλοία έχουν αποκλίνουσες πορείες ή αν το παρακολουθούμενο πλοίο ελαττώσει την ταχύτητά του. Αυτό εκφράζεται από τον υποστόχο «Παρακολούθηση ταχύτητας» (Observe Speed) και παρακολουθείται από την κατάσταση «Μη ελάττωση ταχύτητας με συγκλίνουσες πορείες» (Speed wasn't Reduced when Proximity Decreased). Αν εν τέλει δεν μειωθεί η ταχύτητα του πλοίου ή αυξηθεί η απόστασή των δύο πλοίων σε πάνω από 5 ναυτικά μίλια, τότε εκτελείται η ενέργεια του υποστόχου που είναι «Επιβολή ποινής» (Issue Fine) στον κυβερνήτη του πλοίου.
3. Τέλος, αν το πλοίο ελαττώσει την ταχύτητά του κάτω από το όριο ταχύτητας, τότε ο υποστόχος επιτυγχάνει και ομοίως και ο κόμβος παράλληλων ενεργειών ANY, καθώς και ο στόχος κορυφής. Έτσι ολοκληρώνεται η εκτέλεση του στιγμιότυπου.

Επισημαίνουμε ότι με την επιτυχή ολοκλήρωση ενός εκ των τριών παράλληλων υποστόχων, οι άλλοι δύο διακόπτονται. Τότε οι αντίστοιχοι κόμβοι-κατάστασης απενεργοποιούν τα πρότυπα σύνθετων γεγονότων που είχαν ενεργοποιήσει στον CEP engine, ο οποίος διαγράφεται από τις αντίστοιχες ροές γεγονότων. Ακόμη σημειώνουμε ότι αν μελλοντικά το ίδιο πλοίο αυξήσει εκ νέου την ταχύτητά τότε ολόκληρος ο κύκλος ξεκινά από την αρχή.

7.1.2 Εκτέλεση του σεναρίου

Το παραπάνω σενάριο ελέγχθηκε στην πράξη με χρήση πραγματικών δεδομένων AIS από το AISHub portal. Συγκεκριμένα χρησιμοποιήθηκαν περίπου 1,54 εκατομμύρια γεγονότα, τα οποία αντιστοιχούν σε τρεις ημέρες παρακολούθησης. Αναφέρονται ακολούθως μερικά στοιχεία για το δείγμα των γεγονότων.

- 1,54 εκατομμύρια γεγονότα απ' όλο τον κόσμο
- 2885 πλοία
- 64456 γεγονότα πλοίων (στη ροή γεγονότων πλοίων)
- 1470642 γεγονότα εγγύτητας (στη ροή γεγονότων εγγύτητας)
- 8498 γεγονότα στην περιοχή της Σινικής θάλασσας (περιοχή ευθύνης)
- 7488 γεγονότα για ταχύπλοα σκάφη στην εν λόγω περιοχή.

Κατά την εκτέλεσή του το SAN παρήγαγε 22 ειδοποιήσεις εκ των οποίων οι 20 αφορούσαν πλεύση με μεγάλη ταχύτητα σε θεωλλώδεις συνθήκες, ενώ οι υπόλοιπες 2 πλεύση με μεγάλη ταχύτητα κοντά σε μικρότερα ή άλλα ταχύπλοα πλοία. Η Εικόνα 7-8 δίνει ένα στιγμιότυπο από τη λειτουργία του SAN engine κατά τη διάρκεια του πειράματος, όταν εμφανίστηκε μία ειδοποίηση για πλεύση με μεγάλη ταχύτητα σε θεωλλώδεις συνθήκες.

```

C:\windows\system32\cmd.exe - bin/vmt.bat
:RumpCapacity "" ;
:Grain "" ;
:Destination "QINGDAO" ;
:MMSI "355443000" ;
:SatTelx "" ;
t:endTime "2012-02-19T13:28:43.786+0200" ;
t:stream "http://streams.event-processing.org/ids/vesselstream" ;

Using namespaces: prefix : <http://imu.ntua.gr/ais#> prefix t: <http://events.event-processing.org/types/> prefix s: <http://imu.ntua.gr/ais#> prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> prefix wsnt: <http://docs.oasis-open.org/wsnt/b-2>
RDFContextualizer: addtoContext: Adding to LOCAL context:
  location = [27.27259122,3832],
  event=CEPAT for Situation: Vessel Event received#0,
  node=null
RDFContextualizer: addtoContext: Adding to LOCAL context:
  tm = [2012-02-19T13:28:43.786+0200],
  event=CEPAT for Situation: Vessel Event received#0,
  node=null
RDFContextualizer: addtoContext: Adding to LOCAL context:
  name = [APL ZEEBRUGGE],
  event=CEPAT for Situation: Vessel Event received#0,
  node=null
RDFContextualizer: addtoContext: Adding to LOCAL context:
  type = [containership],
  event=CEPAT for Situation:
  node=null
RDFContextualizer: addtoContext: Adding to LOCAL context:
  url = [http://events.event-processing.org/ids/vesselstream],
  event=CEPAT for Situation:
  node=null
RDFContextualizer: addtoContext: Adding to LOCAL context:
  area = [China Sea],
  event=CEPAT for Situation: Vessel Event received#0,
  node=null
RDFContextualizer: addtoContext: Adding to LOCAL context:
  speed = [15],
  event=CEPAT for Situation: Vessel Event received#0,
  node=null
RDFContextualizer: addtoContext: Adding to LOCAL context:
  mmsi = [355443000],
  event=CEPAT for Situation: Vessel Event received#0,
  node=null
Spawning new thread for 'vessel Moved'
Root Goal 'vessel Moved' starts iteration #121
Root Situation 'vessel Event received' waits for Event
DsbCEPengine: waitforEvent: waiting on CEPAT 'CEPAT for Situation: Vessel Event received#0': SAN execution blocks.
Running Goal 'vessel Moved'
Checking context-condition 'check vessel speed':
JS: CC-0: Vessel APL ZEEBRUGGE (355443000) in jurisdiction
JS: CC-0: Vessel APL ZEEBRUGGE (355443000) is not monitored
JS: CC-0: Vessel speed = 15 knots
false
Thread of Root Goal 'vessel Moved' terminated
  
```

Εικόνα 7-8. Ειδοποίηση για πλεύση σε θεωλλώδεις συνθήκες

Είναι ακόμη ενδιαφέρον να επισημάνουμε ότι όλες οι ειδοποιήσεις που παρήγαγε το SAN απεστάλησαν (published) στο δίαυλο γεγονότων, στη ροή «Ειδοποιήσεις Λιμενικής Αρχής», (PortAlert topic) έτσι ώστε να είναι δυνατή η αξιοποίησή τους από άλλες εφαρμογές. Στην Εικόνα 7-9 φαίνεται ένα τέτοιο γεγονός κωδικοποιημένο με τη γλώσσα RDF.

The screenshot shows a web interface titled "Events" with a search bar containing "Topic: s:PortAlert". Below the search bar is a table with the following data:

ID	Timestamp	Topic	Event
150020	2012-03-03 10:37:30.153	s:PortAlert	<pre><?xml version="1.0" encoding="UTF-8"?> <event-body xmlns:xsd="http://www.w3.org/1999/XMLSchema" xmlns:wst="http://docs.oasis-open.org/wsn/b-2" xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"> <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#" xmlns:esr="http://play.eu/esr/v1/" xmlns:xsd="http://www.w3.org/2001/XMLSchema#"> <rdf:Description rdf:nodeID="10"> <esr:recommendation>JS: CC-2: Vessel APL DALLAS (636091448) sails with strong wind (26 knots) opposite or sideways </esr:recommendation> </rdf:Description> </rdf:RDF> </event-body></pre>

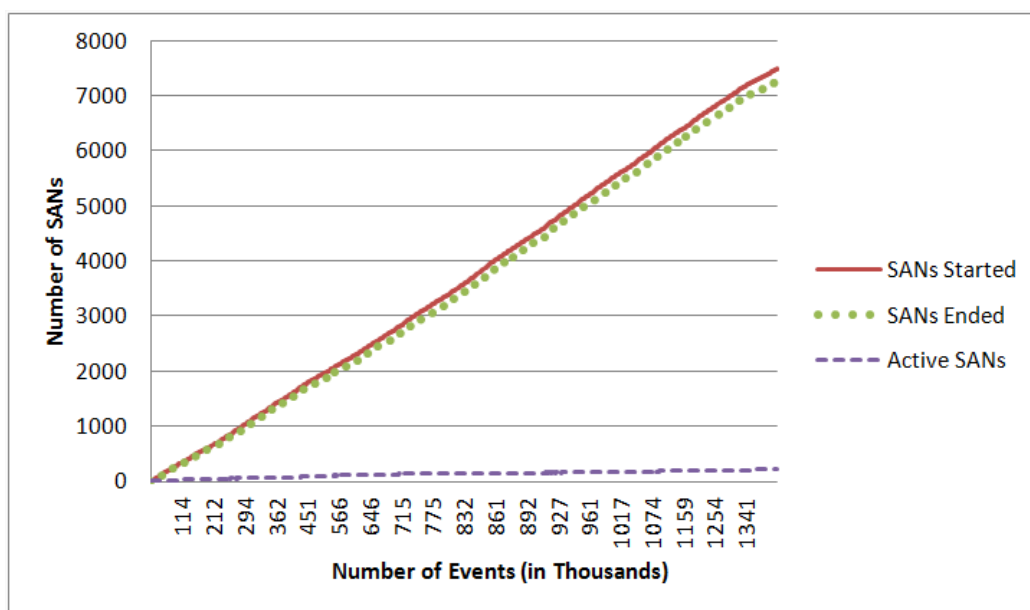
Εικόνα 7-9. Γεγονός στη Ροή γεγονότων «Ειδοποιήσεις Λιμενικής Αρχής»

Το πείραμα εκτελέστηκε σε υπολογιστή με επεξεργαστή Intel Core i5-750 στα 2.66 GHz, Quad Core CPU, με μνήμη 8 GB DDR3 RAM, σκληρό δίσκο 1TB SATA2 7200rpm και λειτουργικό σύστημα Windows 7 έκδοση 64-bit.

7.2 Αποδοτικότητα χρήσης των γεγονότων

Είναι σημαντικό σε συστήματα οδηγούμενα από γεγονότα να λαμβάνεται μέριμνα για τη διατήρηση του κόστους χρήσης των γεγονότων όσο το δυνατό χαμηλότερο ενώ παράλληλα να λαμβάνονται τα απαραίτητα γεγονότα από τις αντίστοιχες ροές. Το κόστος μπορεί να επιφέρεται είτε με την εγγραφή σε κάποια ροή (εφάπαξ ή ανά εγγραφή ή με χρονοχρέωση) είτε με τον όγκο και το είδος των γεγονότων που λαμβάνονται. Φυσικά είναι δυνατοί και συνδυασμοί αυτών των πολιτικών χρέωσης. Αν η χρέωση γίνεται ανά εγγραφή σε ροή γεγονότων, μπορεί τότε να επιτευχθεί ελάττωση του κόστους με την εγγραφή στις απαραίτητες ροές κατά την έναρξη λειτουργίας του συστήματος ή κατά την πρώτη χρήση της ροής. Μια τέτοια πολιτική χρέωσης (και κατά συνέπεια στρατηγική εγγραφής σε ροές) μπορεί να οδηγήσει σε πλεονάζουσες εγγραφές και πλεονάζον όγκο διακινούμενων γεγονότων μέσα από το δίκτυο, και κατά συνέπεια σε αυξημένο κόστος επεξεργασίας, αποθήκευσης των γεγονότων και λειτουργίας του δικτύου. Μια πιο ρεαλιστική πολιτική χρέωσης (τουλάχιστο όταν μιλάμε για μεγάλο όγκο γεγονότων) είναι η χρονοχρέωση ή η ογκοχρέωση. Τότε μια στρατηγική εγγραφής σε ροές γεγονότων, όταν απαιτείται η λήψη των αντίστοιχων γεγονότων, και η διαγραφή από αυτές όταν δεν χρειάζεται πλέον είναι η πιο ενδεδειγμένη για τη διατήρηση του κόστους χρήσης σε σχετικά χαμηλό επίπεδο.

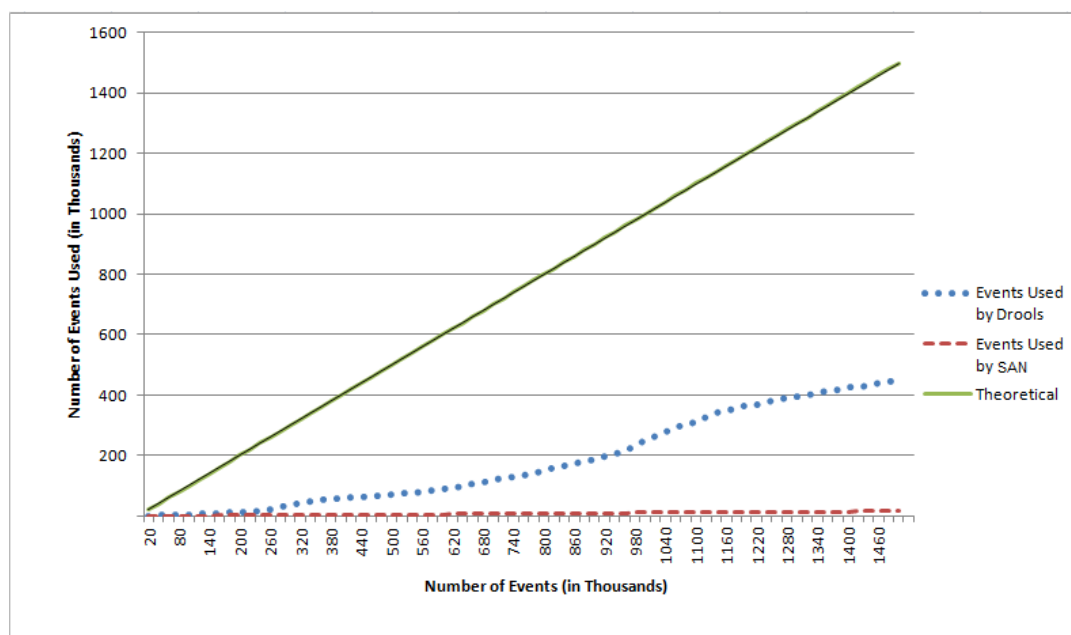
Θεωρούμε το θέμα του κόστους ως κομβικό σημείο για την αξιολόγηση των SANs. Χρησιμοποιώντας τη στρατηγική εγγραφής σε ροές μόνο όταν και για όσο χρόνο χρειάζεται, τα SANs στοχεύουν στην ελάττωση του χρόνου εγγραφής και τη λήψη/χρήση μόνο όσων γεγονότων απαιτούνται για την επίγνωση καταστάσεων ενδιαφέροντος. Στην Εικόνα 7-10 παρουσιάζεται ο αριθμός των στιγμιότυπων των SANs που ξεκίνησαν, τερμάτισαν ή παρέμειναν ενεργά, καθώς ο μηχανισμός SANs ελάμβανε γεγονότα από το AISHub, στο πλαίσιο του σεναρίου χρήσης που παρουσιάστηκε παραπάνω (ενότητα 7.1). Σύμφωνα με το μοντέλο του SAN του σεναρίου, κάθε ενεργό SAN αντιστοιχεί ουσιαστικά σε τρεις εγγραφές σε ροές γεγονότων. Η γραμμή των ενεργών SANs δείχνει το πλήθος των στιγμιότυπων SANs, που παρέμειναν ενεργά παρακολουθώντας τα αντίστοιχα πλοία, σε σχέση με τον συνολικό αριθμό των γεγονότων που ληφθήκαν.



Εικόνα 7-10. Αριθμός SANs σε σχέση με το συνολικό αριθμό γεγονότων

Για την αξιολόγηση του μηχανισμού SANs τον συγκρίναμε με το σύστημα Drools (JBoss community, 2014) χρησιμοποιώντας το ίδιο σύνολο γεγονότων (dataset). Για τη σύγκριση αυτή θα μπορούσε να έχει χρησιμοποιηθεί οποιοσδήποτε μηχανισμός κανόνων ECA αλλά επιλέξαμε το Drools επειδή είναι ελεύθερο λογισμικού ανοιχτού κώδικα. Επίσης δημιουργήσαμε ένα σύνολο κανόνων ECA που υλοποιούν την ίδια συμπεριφορά με το SAN του σεναρίου χρήσης και συγκρίναμε τον αριθμό των γεγονότων που έλαβε και επεξεργάστηκε το κάθε εργαλείο (Εικόνα 7-11). Η πράσινη γραμμή αναπαριστά τη θεωρητική συμπεριφορά ενός απλού μηχανισμού κανόνων ECA χωρίς δυνατότητα δυναμικής αλλαγής των εγγραφών σε ροές γεγονότων κατά το χρόνο εκτέλεσης. Ουσιαστικά αυτό είναι το χειρότερο σενάριο λήψης γεγονότων αφού λαμβάνονται όλα τα γεγονότα των ροών. Παρατηρούμε ότι η επίδοση του Drools είναι σημαντικά βελτιωμένη σε σχέση με τη θεωρητική (worst case). Ο λόγος εξηγείται στη συνέχεια. Ο μηχανισμός SANs πάλι παρουσιάζει ακόμη καλύτερη επίδοση. Αυτό οφείλεται στην ικανότητά του για

δυναμική εγγραφή και διαγραφή από τις ροές γεγονότων που οδηγεί στη σημαντική μείωση του αριθμού των γεγονότων που λαμβάνονται και χρησιμοποιούνται.



Εικόνα 7-11. Σύγκριση χρησιμοποίησης γεγονότων

Οι σημερινοί μηχανισμοί κανόνων ECA (για παράδειγμα RuleCore, Drools) δεν διαθέτουν εγγενώς κάποιον τρόπο δυναμικής μεταβολής των εγγραφών σε ροές γεγονότων, με βάση τις καταστάσεις που επικρατούν ή άλλες σχετικές πληροφορίες του πλαισίου εκτέλεσης. Ωστόσο ενδέχεται να παρέχουν μηχανισμούς για την εκτέλεση παραμετρικών εντολών, οι οποίες μπορεί να χρησιμοποιηθούν για τη δημιουργία νέων κανόνων, με δεδομένα που λαμβάνονται κατά το χρόνο εκτέλεσης. Κατ' αυτό τον τρόπο μπορούν να δημιουργηθούν δυναμικά νέοι κανόνες, όπως αυτός που παρουσιάζεται στη συνέχεια (Πίνακας 7-1), αλλά το συνολικό μοντέλο του συστήματος ή διαδικασίας καταλήγει να είναι δυσνόητο και πολύπλοκο, ιδίως όταν ο αριθμός των κανόνων αυξάνει.

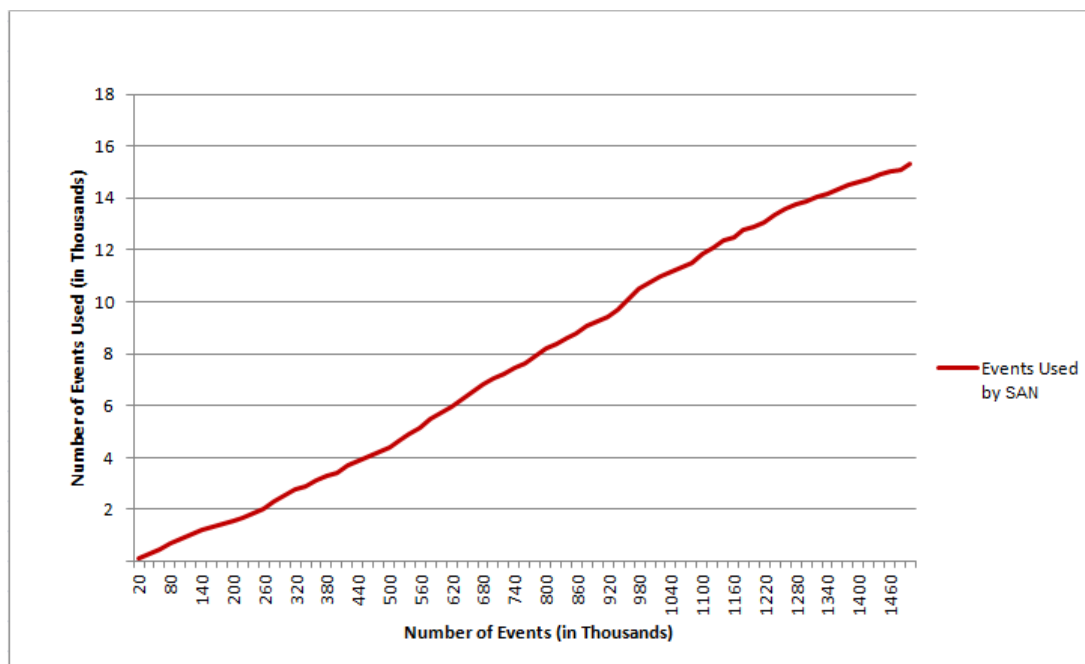
Με την προσέγγιση που παρουσιάζεται ακολούθως (Πίνακας 7-1) μπορεί ένας μηχανισμός κανόνων ECA να δημιουργήσει έναν νέο κανόνα (Rule2) κατά το χρόνο εκτέλεσης. Ουσιαστικά πρόκειται για εμφώλευση (nesting) του νέου κανόνα μέσα στον πρώτο. Το ίδιο μπορεί να γίνει αναδρομικά σε περισσότερα επίπεδα, δηλαδή να δημιουργηθεί μια ιεραρχία εμφωλευμένων κανόνων. Στο συγκεκριμένο παράδειγμα η δημιουργία του νέου κανόνα (Rule2) συμβαίνει όταν εκτελεστεί ο κανόνας Rule1 και προσδιοριστεί η τιμή του MMSI. Με αυτόν τον τρόπο το Drools εγγράφεται δυναμικά στις ροές γεγονότων Vessel και Proximity όπως κάνει και ο μηχανισμός SANs. Ωστόσο, η Εικόνα 7-11 δείχνει ότι υπάρχει σημαντική διαφορά μεταξύ του αριθμού γεγονότων που έλαβε το Drools και αυτών που έλαβε ο μηχανισμός SANs. Αυτή η διαφορά οφείλεται στο ότι δεν απενεργοποιείται ο κανόνας Rule2 όταν δεν τον χρειάζεται πλέον, με αποτέλεσμα να συνεχίσει να λαμβάνει γεγονότα που πρακτικά δεν χρησιμοποιεί. Αντιθέτως, ο μηχανισμός SANs καταγράφει αυτόματα και παρακολουθεί όλες τις ενεργές εγγραφές σε ροές γεγονότων, με τρόπο «διάφανο» (μη αντιληπτό), ενώ πραγματοποιεί διαγραφή από όσες ροές δε χρειάζονται

πλέον. Όλα αυτά υποστηρίζονται εγγενώς από τον μηχανισμό των SANs και γι' αυτό δεν απαιτείται κώδικας ή κάποια άλλη ρύθμιση στα μοντέλα των SANs.

Πίνακας 7-1. Παράδειγμα δυναμικού κανόνα ECA στο Drools

<p>Rule1: ON V , IF V.speed>15Knots AND V.location = China Sea THEN deploy { Rule2: ON V SEQUENCE P, IF V.mmsi = ?mmsi AND V.mmsi=P.mmsi THEN C }</p> <p>Όπου: V, P: είναι οι ροές γεγονότων Vessel και Proximity αντίστοιχα C: είναι η ροή των σύνθετων γεγονότων ?mmsi: Το αναγνωριστικό ενός πλοίου (παράμετρος) που προσδιορίζεται κατά το χρόνο εκτέλεσης.</p>

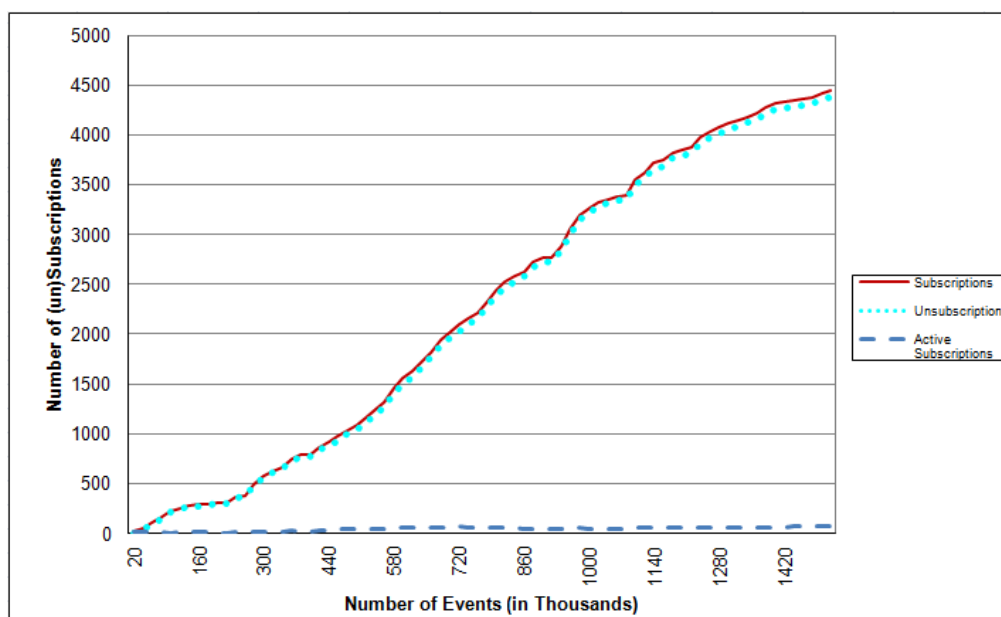
Στην Εικόνα 7-12 δίνεται με μεγαλύτερη λεπτομέρεια το διάγραμμα που συσχετίζει τον αριθμό των γεγονότων που χρησιμοποίησε ο μηχανισμός SANs σε σχέση με τον συνολικό αριθμό γεγονότων.



Εικόνα 7-12. Χρησιμοποίηση γεγονότων

Η Εικόνα 7-13 δίνει τον αριθμό των εγγραφών και διαγραφών σε ροές γεγονότων που πραγματοποιήθηκαν κατά τη διάρκεια του πειράματος. Επίσης δίνει και τον αριθμό των ενεργών εγγραφών κάθε στιγμή, καθώς ο συνολικός αριθμός γεγονότων αυξάνεται. Παρατηρούμε ότι χάρις την ικανότητα του μηχανισμού SANs για δυναμική εγγραφή σε και διαγραφή από ροές γεγονότων οδηγεί σε ένα χαμηλό αριθμό ενεργών εγγραφών κάθε στιγμή. Αυτό πρακτικά σημαίνει ότι το κόστος χρήσης των γεγονότων για το συγκεκριμένο

σενάριο θα ήταν σημαντικά χαμηλότερο σε σχέση με την περίπτωση που δε χρησιμοποιείτο μια στρατηγική δυναμικών εγγραφών/διαγραφών (όπως αυτή των SANs).

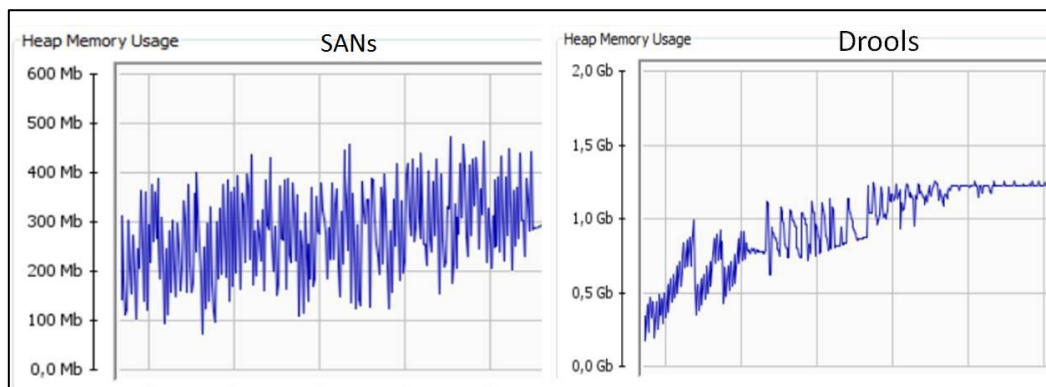


Εικόνα 7-13. Εγγραφές, Διαγραφές και Ενεργές εγγραφές σε ροές γεγονότων

7.3 Επίδοση και Κλιμάκωση

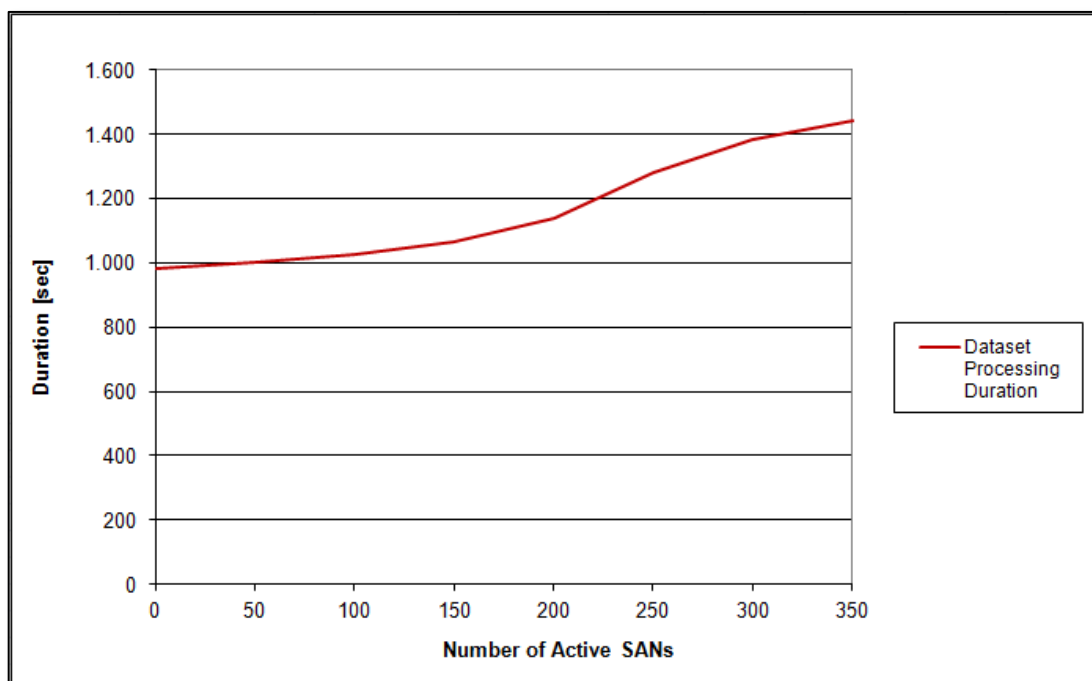
Στη συνέχεια παρουσιάζεται μια σύγκριση της υλοποίησης του SAS με το σύστημα Drools, το οποίο είναι ένα μηχανισμός ECA. Τα δύο συστήματα δεν είναι ομοειδή ωστόσο η σύγκριση επικεντρώνεται σε κοινά τους χαρακτηριστικά όπως ο χρόνος επεξεργασίας ενός γεγονότος και η χρήση μνήμης. Για τη σύγκριση χρησιμοποιήθηκε το εργαλείο JConsole (Oracle, 2011), το οποίο παρέχει γραφική αναπαράσταση της χρήσης της μνήμης (memory usage) σε πραγματικό χρόνο. Σημειώνουμε ότι και τα δύο συγκρινόμενα συστήματα έχουν υλοποιηθεί με Java™. Η σύγκριση έγινε με βάση μετρήσεις που ληφθήκαν κατά τη διάρκεια εκτέλεσης του σεναρίου χρήσης που παρουσιάστηκε παραπάνω (ενότητα 7.1), χρησιμοποιώντας το ίδιο σύνολο γεγονότων.

Στην Εικόνα 7-14 παρατηρείται αύξηση της χρήσης μνήμης και για τα δύο συστήματα καθώς ο αριθμός των εισερχόμενων γεγονότων αυξάνεται. Παρατηρούμε ότι το Drools απαιτεί τη διπλάσια περίπου μνήμη σε σχέση με τον μηχανισμό SANs.



Εικόνα 7-14. Σύγκριση της χρήσης μνήμης

Η Εικόνα 7-15 δίνει το μέσο χρόνο επεξεργασίας ενός γεγονότος που χρειάζεται ο SAN engine καθώς η χρήση του επεξεργαστή αυξάνεται λόγω της παράλληλης εκτέλεσης ολοένα και περισσότερων SANs.



Εικόνα 7-15. Κλιμάκωση του χρόνου επεξεργασίας σε σχέση με τον αριθμό των ενεργών SANs

Η μελέτη της συμπεριφοράς του μηχανισμού των SANs κατά την εκτέλεση, αποκάλυψε ότι αυτός επιδεικνύει μια εύλογη (και αναμενόμενη) αύξηση του χρόνου επεξεργασίας καθώς εκτελούνται παράλληλα ολοένα και περισσότερα στιγμιότυπα του SAN. Όλα τα στιγμιότυπα έχουν την ίδια πολυπλοκότητα και τις ίδιες ανάγκες για χρήση πόρων. Η Εικόνα 7-15 καταδεικνύει ότι ο μηχανισμός SANs έχει την ικανότητα για κλιμάκωση καθώς αυξάνεται ο αριθμός των ενεργών SANs (παράλληλα εκτελούμενων στιγμιότυπων του SAN).

7.4 Πλεονεκτήματα και Μειονεκτήματα

Η χρήση των SANs για τη μοντελοποίηση των συμπεριφορών συστημάτων παρέχει ορισμένα πλεονεκτήματα σε σχέση με άλλες μεθόδους, ιδίως αν πρόκειται για συστήματα των οποίων η συμπεριφορά επηρεάζεται από την εκάστοτε κατάσταση του περιβάλλοντος οπότε απαιτείται επίγνωση αυτής.

Τα SANs υιοθετούν μια προσέγγιση προσανατολισμένη στην επίτευξη στόχων (goal-oriented approach) που επιτρέπει στον σχεδιαστή των SANs να εστιάζει στα επιδιωκόμενα και αναμενόμενα αποτελέσματα του συστήματος. Έτσι με βάση αυτά προσδιορίζει τις καταστάσεις που επηρεάζουν τις συμπεριφορές που οδηγούν στην επίτευξή τους. Επιπλέον η ανάλυση των στόχων σε υποστόχους επιτρέπει την εμβάθυνση της ανάλυσης σε λεπτομέρειες της σχεδιαζόμενης υλοποίησης, παρέχοντας έτσι τη δυνατότητα στον σχεδιαστή των SANs να ελέγχει τις επιμέρους συμπεριφορές του συστήματος, σε οποιοδήποτε επίπεδο και βαθμό λεπτομέρειας απαιτείται.

Επιπλέον τα SANs υποστηρίζουν την επαναχρησιμοποίηση στόχων και συμπεριφορών, με χρήση των κόμβων προσάρτησης (mount actions). Μέσω αυτών, SANs που βρίσκονται σε βιβλιοθήκες μπορούν να προσαρτούνται και να επεκτείνουν άλλα SANs, προσθέτοντάς τους έτσι τις συμπεριφορές που αυτά μοντελοποιούν. Κατ' αυτό τον τρόπο αποφεύγεται η επανάληψη των ίδιων δομών και συμπεριφορών (αποφυγή πλεονασμού), αποφεύγεται η σπατάλη πόρων για την ανάπτυξη του ίδιου μοντέλου πολλές φορές (re-inventing the wheel) και βέβαια γίνεται πιο εύκολη η συντήρησή τους. Επίσης καθίσταται ευκολότερη και η συνεργατική ανάπτυξη μοντέλων συστημάτων.

Τα SANs παρέχουν μια μεθοδολογία μοντελοποίησης που ουσιαστικά «καθοδηγεί» τη σκέψη του σχεδιαστή. Η top-down (ιεραρχική) ανάλυση σε υποστόχους, καταστάσεις και ενέργειες οδηγεί στη λήψη σημαντικών αποφάσεων για το πώς θα μοντελοποιηθεί το σύστημα (ή επιμέρους συμπεριφορές του), ήδη από την αρχή της φάσης του σχεδιασμού. Οι διαδοχικές διαιρέσεις του αρχικού προβλήματος μοντελοποίησης σε επιμέρους υπο-προβλήματα, μέχρις ότου αυτά γίνουν αρκετά μικρά και επιλύσιμα, παρέχει μια συστηματική μεθοδολογία επίλυσης. Έτσι η εργασία του σχεδιαστή μπαίνει από νωρίς σε ένα πλαίσιο, γεγονός που καθιστά ευκολότερο τον προγραμματισμό της και την κατανομή της σε μια ομάδα σχεδιασμού.

Τέλος τα SANs υιοθετούν μια προσέγγιση «αντίδρασης» (reactive approach) ως προς τα ερεθίσματα του περιβάλλοντος. Αυτή άλλωστε είναι η προσέγγιση στην οποία στηρίζεται η πλειονότητα των οδηγούμενων από γεγονότα συστημάτων. Συνεπώς τα SANs αποτελούν μια λογική επιλογή για μοντελοποίηση τέτοιων συστημάτων.

Από την ενασχόληση με τα SANs έχουμε όμως διαπιστώσει και ορισμένα μειονεκτήματα. Κατ' αρχήν τα SANs, ως μια νέα προσέγγιση, δεν έχουν δοκιμαστεί επαρκώς σε πραγματικές εφαρμογές ή σε συνθήκες παραγωγής. Έτσι στερούνται πρακτικής επιβεβαίωσης της καταλληλότητάς τους για εφαρμογή σε ρεαλιστικά περιβάλλοντα. Αυτό είναι κάτι που θα πρέπει να γίνει στο μέλλον, διαδικασία που θα συμβάλλει και στην ωρίμανσή τους ως τεχνολογία. Εξάλλου η παρούσα υλοποίηση του SAS δεν παρέχει βελτιστοποιήσεις και

πρόσθετες λειτουργίες που ενδεχομένως θα απαιτούνταν στην πράξη. Επίσης δεν είναι κατάλληλη για χρήση σε κρίσιμες εφαρμογές.

Ένα σημαντικό μειονέκτημα των SANs που διαπιστώσαμε στην πράξη, είναι το γεγονός ότι δεν είναι εύκολο διαισθητικά να αντιληφθεί κανείς τη ροή και τη διαδοχή των εργασιών της μοντελοποιούμενης διαδικασίας ή συστήματος μέσα από τους γράφους των SANs. Πολλοί αναλυτές έχουν συνηθίσει να σκέφτονται με όρους διαδικασιακού προγραμματισμού ή με ροές εργασιών, όσον τουλάχιστο αφορά τις εκτελούμενες ενέργειες. Τα SANs όμως δίνουν έμφαση στην ιεραρχική ανάλυση στόχων και δεν αναδεικνύουν το διαδικασιακό μέρος του σχεδίου εκτέλεσης. Συνεπώς απαιτείται ένας χρόνος εκμάθησης της χρήσης των SANs που εστιάζει κυρίως στην αλλαγή του τρόπου σκέψης του αναλυτή.

Ένα άλλο, πιο δευτερεύον, μειονέκτημα που εντοπίσαμε αφορά τη χρήση μη γραφικών τρόπων επεξεργασίας των SANs (χωρίς τον SAN editor). Ένα τέτοιο εγχείρημα αποδείχθηκε αρκετά επίπονο ενώ ήταν αρκετά εύκολο να γίνουν σφάλματα. Επιπλέον απαιτείται βαθειά γνώση και κατανόηση του εκάστοτε συντακτικού που χρησιμοποιείται για την αναπαράσταση των SANs. Τέλος, η μίξη των ορισμών και προδιαγραφών των γράφων των SANs με τις εντολές (και γλώσσες) που ορίζουν τα CEPats, τις συνθήκες πλαισίου, τις ενέργειες ή άλλες σημασιολογικές παραμέτρους των κόμβων των SANs, δημιουργεί ένα σχετικά δυσανάγνωστο αποτέλεσμα. Είναι δε αντίστοιχο με παρόμοια προβλήματα που έχουν διαπιστωθεί στις PHP, Java Server Pages (JSP), Active Server Pages (ASP) ή άλλες τεχνολογίες για δυναμική παραγωγή ιστοσελίδων, όπου αναμιγνύεται κώδικας HTML με κάποια γλώσσα προγραμματισμού.

Στο κεφάλαιο που ακολουθεί παρουσιάζεται η δεύτερη καινοτομία του προτεινόμενου πλαισίου εργασίας της διατριβής. Αυτή αφορά στη δημιουργία, την πρόταση και την εφαρμογή προσαρμογών σε επιχειρησιακές διαδικασίες, από κατάλληλα SANs.

8 Θεματοστρεφής προσαρμογή – Η μέθοδος FlexiBPMN2.0

Στο παρόν κεφάλαιο παρουσιάζεται και συζητείται μία νέα μέθοδος για οδηγούμενη από γεγονότα προσαρμογή των επιχειρησιακών διαδικασιών (event-driven business process adaptation). Ο σκοπός των προσαρμογών αυτών είναι τόσο η επιδιόρθωση προβλημάτων που εμφανίζονται κατά την εκτέλεση των διαδικασιών. Το βασικό γνώρισμα της μεθόδου είναι η χρήση και η αξιοποίηση των αρχών και τεχνικών του Θεματοστρεφούς Προγραμματισμού⁴⁹ (Aspect-Oriented Programming ή AOP) για διαχωρισμό των ανεξάρτητων λειτουργιών και απαιτήσεων συστημάτων λογισμικού. Η μέθοδος αυτή καλείται FlexiBPMN2.0 και επεκτείνει καινοτόμες προσεγγίσεις που έχουν προταθεί στη σχετική βιβλιογραφία.

Στη συνέχεια περιγράφεται η «Πλατφόρμα Προτάσεως Προσαρμογών» (Adaptation Recommendation Platform – ARP), η οποία ουσιαστικά υλοποιεί τη μέθοδο FlexiBPMN2.0 για διαδικασίες αποτυπωμένες στη γλώσσα BPMN2.0. Η πλατφόρμα αποτελείται από έναν μηχανισμό SANs και από το «Σύστημα Προσαρμοστικής Εκτέλεσης Διαδικασιών» (Adaptive Process Execution System – APES). Τα SANs χρησιμεύουν για την παρακολούθηση της εξέλιξης και της καλής εκτέλεσης των διαδικασιών καθώς και τη δημιουργία προτάσεων προσαρμογών όταν αυτό κρίνεται απαραίτητο. Το APES από την άλλη πρόκειται για ένα σύστημα εκτέλεσης ροών εργασιών (workflow execution system) που επιτρέπει την προσαρμογή αυτών κατά το χρόνο εκτέλεσης. Συνεργάζεται με τα SANs προκειμένου να λαμβάνει προτάσεις προσαρμογών και τις εφαρμόζει αν αυτές γίνονται δεκτές. Για λόγους συντομίας εφεξής θα χρησιμοποιούμε τους όρους APES για να αναφερόμαστε στο σύστημα προσαρμοστικής εκτέλεσης διαδικασιών και ARP για να αναφερόμαστε στην πλατφόρμα προτάσεως προσαρμογών.

Στην εισαγωγή που ακολουθεί θα δοθεί μια σύντομη περιγραφή των αρχών του θεματοστρεφούς προγραμματισμού και ειδικότερα της προσέγγισης AO4BPMN (Charfi et al., 2010) που σχετίζεται περισσότερο με την πρόταση της παρούσας διατριβής. Αυτό είναι

⁴⁹ Στην ελληνική έκδοση του βιβλίου του Ian Sommerville “Software Engineering”, ο όρος Aspect αποδίδεται ως Θέμα κι ο όρος Aspect-Oriented ως Θεματοστρεφής. (Sommerville, 2009)

απαραίτητο διότι αποτελούν το υπόβαθρο για την κατανόηση της λειτουργίας του APES, το οποίο και θα παρουσιαστεί στην συνέχεια. Ακόμη θα δοθεί μια σύντομη βιβλιογραφική επισκόπηση διαφόρων προσεγγίσεων για θεματοστρεφή προσαρμογή και τέλος θα παρουσιαστεί βήμα προς βήμα ένα παράδειγμα χρήσης της πλατφόρμας ARP για την προσαρμογή μιας διαδικασίας.

8.1 Εισαγωγή

Τα σύγχρονα επιχειρησιακά πληροφοριακά συστήματα αντιμετωπίζουν πολλές προκλήσεις που οφείλονται στην ανάγκη για διαρκή αλλαγή και προσαρμογή των επιχειρησιακών διαδικασιών που υποστηρίζουν. Αυτό είναι απαραίτητο προκειμένου να επιτυγχάνουν τις απαιτούμενες επιδόσεις, ποιότητα και αποτελέσματα, με απώτερο στόχο να παρέχουν συγκριτικό πλεονέκτημα στις επιχειρήσεις και στους οργανισμούς που τα αξιοποιούν. Οι υφιστάμενες προσεγγίσεις για αυτοματοποίηση των διαδικασιών που στηρίζονται σε υπηρεσίες ιστού (web services), όπως η BPEL, συχνά απαιτούν την εκ νέου μοντελοποίηση και επανεκκίνηση (redeployment) των διαδικασιών προκειμένου αυτές να προσαρμοστούν στα νέα δεδομένα και συνθήκες. Ωστόσο κάτι τέτοιο επισύρει σημαντικό κόστος (άμεσο ή έμμεσο) και προκαλεί διακοπές στην λειτουργία των συστημάτων καθώς και απώλεια πληροφοριών, ιδίως σε συστήματα συναλλαγών πραγματικού χρόνου (Hermosillo et al., 2010). Στο πλαίσιο των βασισμένων σε υπηρεσίες εφαρμογών (SBAs) μπορεί να εμφανιστούν διάφορες ανάγκες για αλλαγή ή να ανακύψουν διάφορα προβλήματα. Για παράδειγμα μια υπηρεσία ιστού μπορεί να πάψει να λειτουργεί ή να μεταβληθεί η πολιτική παροχής της κατά τη διάρκεια χρήσης της.

Η προσαρμοστικότητα και η ευελιξία των διαδικασιών, καθώς και των ροών εργασιών (workflows) που τις απαρτίζουν, αποτελούν το αντικείμενο πολλών ερευνητικών εργασιών που διαφέρουν τόσο στον βαθμό μοντελοποίησης (fully designed vs. underspecified), όσο και στον χρόνο μοντελοποίησης των προσαρμογών (design vs. run time) (Schonenberg, Mans, et al., 2008). Ένα πρόβλημα με πολλές από αυτές είναι ότι δεν παρέχουν αυτοματοποιημένη υποστήριξη για την ανίχνευση και αντίδραση στην αλλαγή των καταστάσεων (situation change), εκμεταλλευόμενες τα πλεονεκτήματα μεθόδων και τεχνολογιών όπως της οδηγούμενης από γεγονότα αρχιτεκτονικής (EDA). Επιπρόσθετα, απαιτείται συχνά ανθρώπινη παρέμβαση για την υλοποίηση προσαρμογών. Τέτοιες προσεγγίσεις κρίνονται ακατάλληλες για συστήματα και υπηρεσίες πραγματικού χρόνου (real-time) αφού κατά την προσαρμογή διακόπτεται η λειτουργία των διαδικασιών, ακόμη και κατά τη διάρκεια συναλλαγών, ενώ συνεπάγονται και σημαντικό διαχειριστικό κόστος.

8.2 Σχετικές προσεγγίσεις στη βιβλιογραφία

Αρκετές ερευνητικές προσπάθειες για προσαρμογή εφαρμογών βασισμένων σε υπηρεσίες (SBAs), χρησιμοποιούν τον Θεματοστρεφή Προγραμματισμό (AOP) ως έναν καινοτόμο τρόπο για την ενσωμάτωση εναλλακτικών ενεργειών ή υπηρεσιών μέσα στις επιχειρησιακές διαδικασίες. Αρχικά προτάθηκε ως μια μέθοδος για τον καλύτερο διαχωρισμό των

αρμοδιοτήτων⁵⁰ (concerns) σε συστήματα λογισμικού και για την προσθήκη ανεξάρτητων μεν αλλά «τεμνόμενων» (cross-cutting) λειτουργιών χωρίς να χρειάζονται αλλαγές στην επιχειρησιακή λογική του βασικού λογισμικού. Ως πεδίο έρευνας εμφανίστηκε το 1997 (Kiczales et al., 1997) με πρωτεργάτες τον Gregor Kiczales και την ομάδα του (Kiczales et al., 2001) στο Palo Alto Research Center της Xerox. Παρότι πρωτότυπος και καινοτόμος, ο Θεματοστρεφής Προγραμματισμός κληρονομεί και μεταχειρίζεται ιδέες και έννοιες από άλλα είδη προγραμματισμού, όπως είναι ο ανακλαστικός προγραμματισμός (reflective programming), τα πρωτόκολλα μετα-αντικειμένων (meta-object protocols), τα συστήματα ανοικτής υλοποίησης (open implementations), ο γενικευμένος προγραμματισμός (generative programming) και άλλα. Ως ένα νέο προγραμματιστικό υπόδειγμα, ο Θεματοστρεφής Προγραμματισμός εισάγει νέες έννοιες όπως το *Θέμα* (Aspect), το *Σημείο Τομής* (Pointcut) και η *Συμβουλή* (Advice). Οι έννοιες αυτές δεν υποκαθιστούν αλλά προστίθενται σε ήδη προϋπάρχουσες όπως είναι για παράδειγμα οι έννοιες της κλάσης, του στιγμιότυπου, της διαδικασίας και της μεθόδου. Επιπλέον, οι παραπάνω έννοιες δεν εστιάζουν σε ένα συγκεκριμένο στυλ προγραμματισμού (π.χ. αντικειμενοστραφής ή διαδικασιακός) ή σε ένα συγκεκριμένο συντακτικό (Java, C#, Ada, COBOL κλπ). Αντιθέτως, υπάρχουν επεκτάσεις για Θέματα (aspects) για αρκετές γλώσσες, τόσο αντικειμενοστραφείς όσο και διαδικασιακές.

Πέρα από τις γλώσσες προγραμματισμού έχουν προταθεί θεματοστρεφείς προσεγγίσεις και γλώσσες μοντελοποίησης ροών εργασιών (workflow languages) (Charfi and Mezini, 2006). Αυτές εισάγουν έννοιες που διευκολύνουν τον διαχωρισμό των ανεξάρτητων αρμοδιοτήτων από το κυρίως μοντέλο ροής εργασιών μιας διαδικασίας. Περισσότερες πληροφορίες γι' αυτά θα δοθούν στη συνέχεια.

8.2.1 Βασικές έννοιες του Θεματοστρεφούς Προγραμματισμού

Ακολούθως παρουσιάζονται συνοπτικά οι βασικές έννοιες (δομές) του θεματοστρεφούς προγραμματισμού.

- **Θέματα (Aspects):** είναι δομές που αφορούν και εμπλέκονται σε πολλαπλά σημεία των διαδικασιών, αλλά δε σχετίζονται άμεσα με τους κύριους σκοπούς τους. Τα θέματα χρησιμοποιούνται για να μοντελοποιούν διακριτές και δευτερεύουσες λειτουργίες (αρμοδιότητες), ξεχωριστά από τις κύριες λειτουργίες των συστημάτων και διαδικασιών. Για παράδειγμα η καταγραφή των ενεργειών και των μηνυμάτων (logging) αφορά μεν πολλά σημεία ενός συστήματος, ωστόσο ως θέμα είναι διακριτό και ανεξάρτητο από τους σκοπούς και τις κύριες λειτουργίες του, τις οποίες και «τέμνει» (cross-cuts) θα μπορούσαμε να πούμε.
- **Σημεία Τομών (Pointcuts):** είναι σύνολα από σημεία σύνδεσης (joint points) των εργασιών ή ολόκληρα μέρη των ροών εργασιών μιας διαδικασίας ή συστήματος, όπου είναι δυνατό να γίνουν προσαρμογές. Όταν η εκτέλεση φτάνει σε ένα ενεργό σημείο τομής τότε εξετάζεται αν απαιτείται κάποια προσαρμογή και εφόσον

⁵⁰ Οι αρμοδιότητες (concerns) είναι διαφορετικές όψεις της λειτουργικότητας ενός συστήματος με διακριτούς στόχους μεταξύ τους. Για παράδειγμα, η λογική ενός συστήματος (business logic) και ο τρόπος αλληλεπίδρασης του με τον χρήστη (user interface) είναι δύο αρμοδιότητες.

συμβαίνει κάτι τέτοιο τότε αυτή εκτελείται. Ο έλεγχος αυτός γίνεται με λογικές εκφράσεις, συνθήκες ή κανόνες που επίσης περιλαμβάνονται στα σημεία τομών.

- Τα σημεία σύνδεσης (joint points) είναι οι εισοδοι και οι έξοδοι (inputs & outputs) των εργασιών (ή άλλων δομών) που απαρτίζουν τις διαδικασίες, μέσω των οποίων μπορούν να αλληλεπιδράσουν με το περιβάλλον του ή μεταξύ τους. Εκεί το αποτέλεσμα της εκτέλεσης (output) μιας εργασίας τροφοδοτείται ως είσοδος (input) στην επόμενη της.
- **Συμβουλές (Advices):** είναι λειτουργίες ή (υπο-)διαδικασίες που συμπληρώνουν, αντικαθιστούν και τροποποιούν μέρη μιας διαδικασίας, επεμβαίνοντας στα σημεία σύνδεσης των εργασιών τους και αντικαθιστώντας τα αντίστοιχα σημεία τομής. Εκτελούνται όταν η ροή εκτέλεσης φτάνει σε ενεργά σημεία τομών (pointcuts) που σχετίζονται με αυτές και κριθεί ότι είναι απαραίτητες οι προσαρμογές που παρέχουν. Οι συμβουλές χρησιμοποιούνται για να διαχωρίζουν δομικά (σε ξεχωριστές μονάδες λογισμικού) τις δευτερεύουσες από τις κύριες λειτουργίες ενός συστήματος. Σημειώνουμε ότι οι μονάδες ενός συστήματος, αν και διακριτές μεταξύ τους, μπορεί να συνεργάζονται στενά, να αλληλεπιδρούν και να παρεμβάλλονται οι λειτουργίες τις μιας ανάμεσα στις λειτουργίες των άλλων, δηλαδή να αλληλοπλέκονται. Βασικό χαρακτηριστικό του θεματοστρεφούς προγραμματισμού είναι και η «πλέξη⁵¹» (weaving) των διακριτών αυτών λειτουργιών σε ένα ενιαίο και συνεκτικό πρόγραμμα. Οι προσαρμογές πραγματοποιούνται με την προσθήκη και ενσωμάτωση των αντίστοιχων συμβουλών (ήτοι λειτουργιών ή υπο-διαδικασιών) στον κορμό ενός συστήματος ή διαδικασίας. Η ενσωμάτωση μπορεί να γίνει είτε στατικά, κατά τη δημιουργία ενός συστήματος (compile/build time), είτε δυναμικά κατά την εκτέλεσή του. Το ενδιαφέρον της παρούσας διατριβής εστιάζεται στη δεύτερη περίπτωση.

8.2.2 AO4BPEL

Μια από τις πλέον γνωστές προσεγγίσεις για προσαρμογή υπηρεσιών (service adaptation) με χρήση θεματοστρεφούς προγραμματισμού είναι η εργασία των (Charfi and Mezini, 2007) όπου παρουσιάζεται η AO4BPEL. Πρόκειται για μια γλώσσα βασισμένη στην XML, η οποία παρέχει ένα επίπεδο λειτουργικότητας που πλαισιώνει τη γλώσσα BPEL, προκειμένου να καταστεί δυνατή η χρήση θεμάτων κατά την εκτέλεση επιχειρησιακών διαδικασιών σε BPEL. Τα θέματα απαρτίζονται από ένα ή περισσότερα «σημεία τομών» (pointcuts) και αντίστοιχες «συμβουλές» (advices). Η AO4BPEL βασίζεται στη γλώσσα ερωτημάτων XPath⁵² για την επιλογή των σημείων σύνδεσης (joint points), δηλαδή των σημείων εκείνων της διαδικασίας που αντιστοιχούν στην αρχή ή στο τέλος της εκτέλεσης ενεργειών ή εσωτερικών σημείων (internal joint points) που βρίσκονται εντός των εκτελούμενων ενεργειών. Μια Συμβουλή (advise) είναι μια καινούρια λειτουργία που θα πρέπει να εισαχθεί σε ένα σημείο σύνδεσης και περιέχει νέο κώδικα που θα πρέπει να εκτελεστεί.

⁵¹ Εμείς προτιμούμε τους όρους «ενσωμάτωση» ή «εφαρμογή» μιας προσαρμογής, αντί του όρου «πλέξη» που είναι η ακριβής μετάφραση του weave / weaving.

⁵² <http://www.w3.org/TR/xpath/>

Προβλέπονται ειδικές δομές που μπορούν να χρησιμοποιηθούν από τον κώδικα μιας Συμβουλής για να αποκτήσει πρόσβαση στα δεδομένα εισόδου/εξόδου των σημείων τομής, στα αντίστοιχα μηνύματα SOAP καθώς και σε άλλες πληροφορίες. Στη βασική της μορφή η AO4BPEL υποστηρίζει μόνο την χειροκίνητη (manual) ενεργοποίηση και απενεργοποίηση των θεμάτων διαμέσου μιας διαχειριστικής διεπαφής. Η ενεργοποίηση ενός θέματος παρέχει τη δυνατότητα χρήσης της αντίστοιχης προσαρμογής, αν φυσικά υπάρχει η αντίστοιχη ανάγκη.

Το πλεονέκτημα της χρήσης της AO4BPEL είναι το ότι οι περιγραφές των επιχειρησιακών διαδικασιών μπορούν να αλλάζουν κατά την εκτέλεσή τους χωρίς να χρειάζεται να «επανεκκινηθούν» (redeloy) με επακόλουθο την απώλεια των πληροφοριών και των συναλλαγών όλων των ενεργών στιγμιότυπων των διαδικασιών. Η AO4BPEL υποστηρίζει δύο τρόπους εφαρμογής των θεμάτων: την εφαρμογή σε επίπεδο διαδικασίας (process-level deployment) και την εφαρμογή σε επίπεδο στιγμιότυπου (instance-level deployment) (Charfi and Mezini, 2007). Ο πρώτος τρόπος χρησιμοποιείται όταν ένα θέμα πρέπει να εφαρμοστεί σε όλα τα στιγμιότυπα μιας επιχειρησιακής διαδικασίας, ενώ ο δεύτερος χρησιμοποιείται όταν επηρεάζονται μόνο συγκεκριμένα στιγμιότυπα.

Μια διαδικασία μπορεί να προσαρμοστεί με τη χρήση τριών τύπων συμβουλών: τη συμβουλή «Πριν» (Before advice), τη συμβουλή «Μετά» (After advice) και τη συμβουλή «Πέριξ» (Around advice). Ο πρώτος τύπος ενεργοποιείται κατά την έναρξη εκτέλεσης (κλήσης) μιας υπηρεσίας που ανήκει σε ένα ενεργό σημείο τομής, οπότε ο κώδικας της συμβουλής εκτελείται πριν από αυτό. Μετά την ολοκλήρωση της εκτέλεσης της συμβουλής συνεχίζει η εκτέλεση της κανονικής ροής εργασιών της διαδικασίας, συμπεριλαμβανομένης και της εργασίας του σημείου τομής (pointcut activity) πριν από την οποία παρεμβλήθηκε η συμβουλή. Η συμβουλή «Μετά» εισάγεται ακριβώς μετά την εκτέλεση ενός ενεργού σημείου τομής και μετά την ολοκλήρωσή της η εκτέλεση της διαδικασίας συνεχίζει κανονικά. Τέλος, η συμβουλή «Πέριξ» παρεμβάλλεται τόσο πριν την έναρξη εκτέλεσης της εργασίας ενός σημείου τομής, όπως και η συμβουλή «Πριν», όσο και μετά το σημείο τομής, όπως η συμβουλή «Μετά». Επίσης η εκτέλεση του σημείου τομής είναι δυνατό να παραλειφθεί.

Οι Charfi et al. (2009) στηριζόμενοι στην AO4BPEL παρουσίασαν μια αρχιτεκτονική βασισμένη σε δυναμικά πρόσθετα (dynamic plug-ins) για αυτό-προσαρμοζόμενες διαδικασίες. Συγκεκριμένα, επέκτειναν την AO4BPEL ούτως ώστε να υποστηρίζει δυναμικά πρόσθετα που δημιουργούν, ενεργοποιούν και απενεργοποιούν θέματα κατά την εκτέλεση των διαδικασιών. Επίσης εισήγαγαν νέες εντολές για ενεργοποίηση και απενεργοποίηση των θεμάτων μέσα από τον κώδικα συμβουλών άλλων θεμάτων. Υπενθυμίζουμε ότι αρχικά η AO4BPEL υποστήριζε μόνο χειροκίνητη ενεργοποίηση και απενεργοποίηση των θεμάτων διαμέσου μιας διαχειριστικής διεπαφής.

Η αρχιτεκτονική που πρότειναν συμπεριλαμβάνει έναν μηχανισμό ενορχήστρωσης υπηρεσιών (AO4BPEL orchestration engine) που υποστηρίζει τη χρήση ειδικών πρόσθετων για υλοποίηση αυτο-προσαρμογών (self-adaptation), τα οποία εισάγουν δύο είδη επεκτάσεων στις διαδικασίες: τις έμμεσες (implicit) και τις άμεσες (explicit). Κάθε

δραστηριότητα μιας διαδικασίας αποτελεί εν δυνάμει ένα σημείο έμμεσης επέκτασης όπου τα πρόσθετα μπορούν να εκτελέσουν τη λογική τους για προσαρμογή της διαδικασίας. Στα σημεία τώρα άμεσης επέκτασης, η αρχιτεκτονική μπορεί να επεκταθεί με web services που παρέχονται από τα πρόσθετα, παραδείγματος χάριν για παρακολούθηση ή διάγνωση μιας διαδικασίας, έτσι ώστε να μπορούν να αποφασίσουν αν απαιτείται προσαρμογή. Κάθε πρόσθετο επιδιώκει κάποιους συγκεκριμένους και καλά καθορισμένους στόχους (π.χ. αυτό-ίαση, self-healing). Αποτελούνται δε από θέματα και υπηρεσίες που αναπτύσσονται από ειδικούς στον εκάστοτε τομέα (domain experts) και εισάγονται για εκτέλεση στον μηχανισμό ενορχήστρωσης μέσω κατάλληλης διαχειριστικής διεπαφής. Κάθε πρόσθετο περιλαμβάνει δύο τύπους θεμάτων: τα θέματα παρακολούθησης (monitoring aspects), τα οποία συλλέγουν πληροφορίες και αποφασίζουν αν απαιτείται προσαρμογή στη διαδικασία, και τα θέματα προσαρμογής (adaptation aspects), τα οποία αντιμετωπίζουν τις προβληματικές καταστάσεις που ανιχνεύθηκαν από τα θέματα παρακολούθησης. Τα θέματα παρακολούθησης μπορούν να ενεργοποιούν και να απενεργοποιούν τα θέματα προσαρμογής κατά την εκτέλεση.

Η ανάπτυξη των πρόσθετων για αυτό-προσαρμογή γίνεται σύμφωνα με το υπόδειγμα Sense–Think–Act (Ανίχνευση–Σκέψη–Δράση) που χρησιμοποιείται σε εφαρμογές Τεχνητής Νοημοσύνης (AI), και αποτελείται από τρεις φάσεις. Αναλυτικότερα, ένας προγραμματιστής εξετάζει συγκεκριμένα (concrete) σενάρια προβλημάτων και υλοποιεί την **Ανίχνευση** (Sense) χρησιμοποιώντας σημεία τομών. Έπειτα υλοποιεί τη **Σκέψη** (Think) με τη χρήση συμβουλών που μπορούν να αναγνωρίζουν τις κρίσιμες καταστάσεις που ενδεχομένως να προκύψουν, ακολουθώντας μία από τις εξής τρεις στρατηγικές: (1) εξέταση πριν την εκτέλεση (pre-examination), (2) εξέταση μετά την εκτέλεση (post-examination) και (3) πλαισίωση (wrapping). Η πρώτη στρατηγική (pre-examination) διασφαλίζει προληπτικά (proactively) ότι οι συγκεκριμένες ενέργειες που πρόκειται να εκτελεστούν δεν θα αποτύχουν ή καθορίζει εναλλακτικές ενέργειες για την αντικατάσταση τυχόν προβληματικών. Η δεύτερη στρατηγική (post-examination) εκτελείται μετά την ολοκλήρωση της εκτέλεσης των ελεγχόμενων ενεργειών (της διαδικασίας) και λαμβάνει θεραπευτικά μέτρα στην περίπτωση κάποιας αποτυχίας ή προβλήματος. Τέλος η στρατηγική της πλαισίωσης (wrapping) διασφαλίζει ότι μια ενέργεια δεν θα αποτύχει ενθυλακώνοντάς την (encapsulation) σε μια «περιβάλλουσα» ενέργεια (wrapping activity), όπως είναι ένας χειριστής σφαλμάτων (fault handler), η οποία «συλλαμβάνει» τις τυχόν προβληματικές καταστάσεις που θα ανακύψουν κατά την εκτέλεση της κύριας ενέργειας. Η πλαισίωση χρησιμοποιείται όταν δεν μπορεί να ελεγχθεί προκαταβολικά αν θα αποτύχει η εκτέλεση μιας ενέργειας κι έτσι δεν μπορεί να εφαρμοστεί η στρατηγική εξέτασης πριν την εκτέλεση (pre-examination). Τέλος, ο προγραμματιστής υλοποιεί τη **Δράση** (Act) με τη μορφή ενός θέματος προσαρμογής (adaptation aspect), το οποίο χρησιμοποιεί τα δικά του σημεία τομών και συμβουλές για να τροποποιήσει την εκτελούμενη διαδικασία.

Η έρευνα των ιδίων έχει εστιαστεί τόσο στο επίπεδο εκτέλεσης των διαδικασιών (Charfi and Mezini, 2007) όσο και στο επίπεδο της μοντελοποίησης αυτών (Charfi et al., 2009). Σε πιο πρόσφατη εργασία τους εισήγαγαν την AO4BPMN (Charfi et al., 2010), η οποία είναι μια επέκταση της γλώσσας BPMN ούτως ώστε να υποστηρίζει τη χρήση θεμάτων και τον

διαχωρισμό των ανεξάρτητων απαιτήσεων. Περισσότερα στοιχεία γι' αυτήν θα δοθούν στη συνέχεια του κεφαλαίου, αφού η AO4BPMN αποτέλεσε τη βάση για την ανάπτυξη της πρότασης της διατριβής όσον αφορά τη θεματοστρεφή προσαρμογή επιχειρησιακών διαδικασιών.

Οι Witteborg et al. (2014) παρουσίασαν μια νεώτερη έκδοση της AO4BPMN, τη μέθοδο AO4BPMN 2.0, η οποία αντιμετωπίζει μια σειρά από μειονεκτήματα της πρώτης. Συγκεκριμένα προδιαγράφει ένα συγκεκριμένο μηχανισμό εφαρμογής (πλέξης) των θεμάτων στα μοντέλα επιχειρησιακών διαδικασιών, προτείνει και χρησιμοποιεί την OCL ως γλώσσα ελέγχου των σημείων τομών (για επιλογή των σημείων σύνδεσης) και τέλος υποστηρίζει πλέον και διαδικασίες σε BPMN 2.0. Η μέθοδος αυτή εστιάζει στον μετασχηματισμό του μοντέλου μιας διαδικασίας σε ένα νέο, το οποίο περιλαμβάνει τις αλλαγές και τις νέες ροές εργασιών των σχετικών θεμάτων, κατάλληλα συνδυασμένες (πλεγμένες) με εκείνες του αρχικού μοντέλου (model-to-model transformation). Προφανώς ο μετασχηματισμός αυτός λαμβάνει χώρα πριν την εκτέλεση των διαδικασιών. Τέλος η AO4BPMN 2.0 έχει υλοποιηθεί με χρήση εργαλείων λογισμικού ανοικτού κώδικα όπως το Eclipse BPMN2 Modeler⁵³ και το QVTo⁵⁴.

8.2.3 BPEL'n'Aspects

Μια άλλη προσέγγιση που επίσης εισάγει τη χρήση θεμάτων στη γλώσσα BPEL, παρουσιάζεται στην εργασία των Karastoyanova and Leymann (2009). Η προσέγγιση BPEL'n'Aspects επιτρέπει την υλοποίηση θεμάτων τόσο με περιγραφές BPEL όσο και με τη χρήση υπηρεσιών ιστού. Έχει αποφευχθεί η εισαγωγή επεκτάσεων στην ίδια τη γλώσσα προκειμένου η προσέγγιση να είναι συμβατή με υφιστάμενες (legacy) διαδικασίες. Επίσης συνδυάζει την (κλασική) BPEL με το μοντέλο αποστολής / εγγραφής (publish / subscribe paradigm) καθώς και με το πρότυπο WS-Policy⁵⁵, ούτως ώστε οι λειτουργίες (operations) των web services να παίζουν το ρόλο των θεμάτων για τις διαδικασίες BPEL. Συγκεκριμένα τα θέματα εκφράζονται με ένα συντακτικό που επεκτείνει το πρότυπο WS-Policy.

Στόχος της BPEL'n'Aspects είναι η παροχή ευελιξίας και προσαρμοστικότητας με έναν μη παρεμβατικό τρόπο που λαμβάνει υπόψη τις υφιστάμενες τεχνολογίες και υποδομές, έτσι ώστε να εξασφαλιστεί η μεγαλύτερη δυνατή αποδοχή της προτεινόμενης προσέγγισης. Η βασική ιδέα είναι η παροχή ενημέρωσης στις ενδιαφερόμενες υπηρεσίες για το γεγονός ότι η εκτέλεση μιας διαδικασίας BPEL έφτασε σε κάποιο σημείο ενδιαφέροντος (σημείο τομής). Οι ενδιαφερόμενες υπηρεσίες θα πρέπει εκ των προτέρων να έχουν εγγραφεί (subscribe) για τη λήψη των σχετικών ενημερώσεων. Κάθε φορά που η εκτέλεση φτάνει σε ένα σημείο τομής καλούνται συγκεκριμένες λειτουργίες (operations) των ενδιαφερόμενων υπηρεσιών (που προσδιορίζονται κατά την εγγραφή). Αυτό πρακτικά ισοδυναμεί με την εξέταση των συνθηκών ελέγχου των σημείων τομής και την εκτέλεση των αντίστοιχων συμβουλών.

⁵³ <http://eclipse.org/bpmn2-modeler/>

⁵⁴ <http://projects.eclipse.org/projects/modeling.mmt.qvt-oml>

⁵⁵ <http://www.w3.org/2002/ws/policy/> και <http://schemas.xmlsoap.org/ws/2004/09/policy/>

Η αρχιτεκτονική της BPEL's Aspects απαρτίζεται από τέσσερα κύρια στοιχεία: τον μηχανισμό εκτέλεσης BPEL, έναν δίαυλο υπηρεσιών (service bus) (Leymann, 2005) για την κλήση των υπηρεσιών που υλοποιούν τις εργασίες της διαδικασίας, έναν μεσάζοντα (broker) με αποστολή του την ενσωμάτωση (weaving) των θεμάτων στις διαδικασίες και ένα εργαλείο διαχείρισης των θεμάτων (Aspect Management Tool) για τη δημιουργία, επεξεργασία, ενεργοποίηση, απενεργοποίηση και διαγραφή των θεμάτων. Προκειμένου να επιτευχθεί η επικοινωνία μεταξύ του μεσάζοντα και του μηχανισμού εκτέλεσης BPEL, έχει προστεθεί ένα νέο υποσύστημα στον μηχανισμό ActiveBPEL⁵⁶ (Khalaf et al., 2007), έναν ανοικτού κώδικα μηχανισμό εκτέλεσης BPEL. Το υποσύστημα αυτό καλείται ελεγκτής (controller) και είναι υπεύθυνος για την αποστολή ειδοποιήσεων σχετικών με τις διάφορες φάσεις εκτέλεσης της κάθε εργασίας που περιλαμβάνεται σε μια διαδικασία BPEL. Επίσης χρησιμοποιήθηκε η AspectJ⁵⁷, μια θεματοστρεφής επέκταση της γλώσσας προγραμματισμού JavaTM καθώς και το ActiveMQ⁵⁸, ένα ενδιάμεσο λογισμικό (middleware) για την υλοποίηση της εγγραφής των ενδιαφερόμενων υπηρεσιών και της αποστολής ενημερώσεων σε αυτές.

8.2.4 CEVICHE

Πιο πρόσφατα, οι Hermosillo et al. (2010) πρότειναν μια άλλη προσέγγιση βασισμένη στον θεματοστρεφή προγραμματισμό, το CEVICHE ή Complex Event processing for Context-adaptive processes in pervasive and Heterogeneous Environments. Πρόκειται για ένα πλαίσιο εργασίας που αποβλέπει στην υποστήριξη επιχειρησιακών διαδικασιών με ικανότητα επίγνωσης του πλαισίου τους (context-aware business processes), προκειμένου αυτές να προσαρμόζονται δυναμικά για να ανταποκρίνονται σε διαφορετικά σενάρια εκτέλεσης. Ο κύριος στόχος του CEVICHE είναι να παράσχει ένα μηχανισμό για αυτόματες προσαρμογές ούτως ώστε να επιτυγχάνεται υψηλή ποιότητα υπηρεσίας (QoS), όσον αφορά τις επιδόσεις και τη διαθεσιμότητα των υπηρεσιών ιστού που χρησιμοποιούνται σε επιχειρησιακές διαδικασίες. Το CEVICHE στηρίζεται στην επέκταση της γλώσσας BPEL με την AO4BPEL (Charfi and Mezini, 2007) και στη χρήση μηχανισμών επεξεργασίας σύνθετων γεγονότων (CEP engines) για την ανίχνευση καταστάσεων που απαιτούν προσαρμογή. Για το σκοπό αυτό όρισαν μια γλώσσα που ονομάζεται Πρότυπη Γλώσσα Επιχειρησιακών Διαδικασιών (Standard Business Process Language ή SBPL), η οποία συγκεντρώνει σε ένα αρχείο XML όλες τις πληροφορίες για τις διαδικασίες, το πλαίσιο, τους κανόνες λειτουργίας και τις συνθήκες (που χρίζουν) προσαρμογής.

Το CEVICHE απαρτίζεται από τρία κύρια τμήματα: μια διεπαφή με τον χρήστη για τη δημιουργία των αρχείων SBPL, μια υποδομή μετάφρασης που διαχειρίζεται και τις επεκτάσεις (plug-ins) για τον CEP engine, και έναν διαχειριστή θεμάτων (aspect manager) για την εφαρμογή των προσαρμογών στις διαδικασίες. Οι πληροφορίες του SBPL αρχείου δίνονται στην υποδομή μετάφρασης, η οποία τις διαιρεί σε τρία μέρη: τη διαδικασία BPEL,

⁵⁶ <http://www.activebpel.org>

⁵⁷ <http://eclipse.org/aspectj/>

⁵⁸ <http://activemq.apache.org/>

τις καταστάσεις που χρίζουν προσαρμογών (adaptation situations) με τη μορφή κανόνων για τον CEP engine (CEP rules), και τα θέματα που προσαρμόζουν τη διαδικασία. Με αυτόν τον τρόπο καθορίζονται τα μέρη της διαδικασίας που μπορεί να απαιτήσουν προσαρμογή, τα σημεία τομής δηλαδή. Χρησιμοποιώντας τα σημεία τομής καθορίζεται και το ποια συγκεκριμένη συμβουλή πρέπει να εφαρμοστεί σε κάθε σημείο της διαδικασίας κατά την εκτέλεση. Όταν ο κώδικας μιας συμβουλής ενσωματωθεί στη διαδικασία τότε αυτή προσαρμόζεται αντίστοιχα. Τέλος, με τη χρήση επεξεργασίας σύνθετων γεγονότων (CEP) αναλύονται τα διάφορα γεγονότα καθώς και το πλαίσιο τους, έτσι ώστε να αποφασίζεται αυτόματα πότε και ποιες προσαρμογές θα γίνονται στη διαδικασία.

8.2.5 Λοιπές εργασίες

Οι Morin et al. (2008) πρότειναν μια προσέγγιση που εκμεταλλεύεται τη Θεματοστρεφή Μοντελοποίηση (Aspect-Oriented Modeling ή AOM) καθώς και τον Οδηγούμενο από Μοντέλα Σχεδιασμό (Model Driven Engineering ή MDE) προκειμένου να διαχειριστούν την μεταβλητότητα και την ανάγκη προσαρμογής συστημάτων που ήδη λειτουργούν. Η εργασία τους βασίζεται στην έννοια των θεμάτων μοντέλων (aspect models), τα οποία μπορούν να ενσωματωθούν (woven) στο μοντέλο διαμόρφωσης (configuration) του εκτελούμενου συστήματος. Το μοντέλο διαμόρφωσης λειτουργεί «επάνω» από το σύστημα εποπτεύοντας και ελέγχοντάς το.

Οι Morin et al. (2009) επεκτείνουν την αρχική τους εργασία προκειμένου να δείξουν τον τρόπο με τον οποίο τα θέματα μπορούν να βοηθήσουν τους σχεδιαστές να προσδιορίσουν τις αλληλεπιδράσεις μεταξύ διαφόρων δυναμικών παραλλαγών (variants) των διαμορφώσεων. Επίσης δείχνουν το πως τα μοντέλα μπορούν να χρησιμοποιηθούν για την επικύρωση της ορθότητας (validation) των νέων διαμορφώσεων κατά το χρόνο εκτέλεσης (on the fly), και πριν αυτά εφαρμοστούν στο εκτελούμενο σύστημα. Επικεντρώνονται στα «σημεία μεταβλητότητας» και στις «παραλλαγές» των συστημάτων (που αναπαριστώνται από τα θέματα) και δεν εστιάζουν σε ολόκληρες τις διαμορφώσεις των συστημάτων. Μια διάσταση «μεταβλητότητας» είναι μια συγκεκριμένη αρμοδιότητα (concern) που μπορεί να πραγματοποιηθεί με διάφορους τρόπους. Χρησιμοποιούν το SmartAdapters (Lahire et al., 2007), ένα εργαλείο AOM για την εφαρμογή θεμάτων σε επίπεδο μοντέλου διαδικασίας. Ωστόσο η προσέγγιση που παρουσιάζουν στις δημοσιεύσεις τους δεν εξαρτάται από το SmartAdapters αλλά μπορεί να χρησιμοποιηθούν και άλλα εναλλακτικά εργαλεία AOM, όπως το MATA (Jayaraman et al., 2007). Τα θέματα μπορεί προαιρετικά να επεκταθούν με πλαίσια (contexts) περιγραφής των αιτίων ενεργοποίησης καθώς και του τρόπου εφαρμογής τους. Ένα πλαίσιο είναι (εννοιολογικά) ένα μέρος του περιβάλλοντος που περιγράφει πότε ένα θέμα είναι χρήσιμο καθώς και τις επιδράσεις του στις παραμέτρους ποιότητας υπηρεσίας (QoS). Θέματα χωρίς πλαίσιο μπορούν να ενεργοποιηθούν χειροκίνητα από τον χρήστη. Όταν δημιουργείται και επικυρώνεται ένα μοντέλο «στόχος» (target model), το οποίο αναπαριστά το σύστημα που θέλουμε να έχουμε, αυτό συγκρίνεται με το αρχικό μοντέλο (source model) που αναπαριστά την πραγματική αρχιτεκτονική του υφιστάμενου συστήματος. Η σύγκριση γίνεται με χρήση του EMF Ecore Compare⁵⁹ και

⁵⁹ <http://www.eclipse.org/modeling/emft/> και http://wiki.eclipse.org/index.php/Ecore_Tools

λαμβάνονται οι αλλαγές που απαιτούνται για τη μετάβαση από το αρχικό μοντέλο στο μοντέλο στόχο, όπως προσθήκη ή αφαίρεση στοιχείων (components) ή συνδέσεων (bindings), αλλαγή τιμών των ρυθμίσεων και άλλα.

Στην ίδια εργασία εστιάζουν ακόμη και στην επικύρωση των τελικών διαμορφώσεων (target configurations), καθορίζοντας ορισμένες σταθερές (Invariants) στο μετα-μοντέλο που χρησιμοποιούν (για την αναπαράσταση της πραγματικής αρχιτεκτονικής) και τις οποίες ελέγχουν για κάθε τελική διαμόρφωση που δημιουργείται (μέσω της εφαρμογής θεμάτων) και που τίθεται προς εκτέλεση. Οι σταθερές αυτές εκφράζονται ως μετα-θέματα Kermeta⁶⁰ (Muller et al., 2005), τα οποία εφαρμόζονται στο μετα-μοντέλο. Τα μετα-θέματα Kermeta χρησιμοποιούνται για βελτιώσουν τις υφιστάμενες μετα-κλάσεις (meta-classes) εισάγοντας «συμβάσεις» (contracts) για τις προϋποθέσεις / επιπτώσεις (pre/post-conditions) των μετα-κλάσεων, τις σταθερές, τις ιδιότητες, τις αναφορές, τις λειτουργίες και τις υπερ-κλάσεις (super-classes) τους. Όλα τα μοντέλα (μετα-μοντέλο, διαμορφώσεις και θέματα) μπορούν να εκφραστούν σε XML και να μεταδοθούν σε άλλα συστήματα ή μπορεί να χρησιμοποιηθούν για προσομοιώσεις.

Οι Sánchez and Villalobos (2008) χρησιμοποίησαν μια θεματοστρεφή προσέγγιση που εισάγει τη διάκριση ανάμεσα σε αρμοδιότητες (concerns) και τα μέσα επίτευξής τους (instrumentation). Στόχος τους είναι η ανάπτυξη επεκτάσιμων εφαρμογών ροής εργασιών χρησιμοποιώντας συγχρονιζόμενα εκτελέσιμα μοντέλα. Επίσης χρησιμοποιούν ανοικτά αντικείμενα (open objects) που είναι αναπαραστάσεις των καταστάσεων των διαφόρων στοιχείων ενός μοντέλου, προκειμένου να παρακολουθούν τις κλήσεις (invocations) σε υπηρεσίες και να προσαρμόζουν τις διαδικασίες ενσωματώνοντας αλληλεπιδράσεις με άλλα μοντέλα (πριν ή/και μετά τις κλήσεις).

Μια άλλη προσέγγιση, επίσης προσανατολισμένη σε θέματα, παρουσιάζεται στην εργασία των Rahman et al. (2008), όπου προτείνει μια προσαρμοστική αρχιτεκτονική βασισμένη σε κανόνες ECA καθώς και έναν μηχανισμό που την υλοποιεί. Αυτή βασίζεται στις αρχές της προσανατολισμένης σε υπηρεσίες αρχιτεκτονικής (service-oriented architecture) και του θεματοστρεφούς προγραμματισμού, και αφορά επιχειρησιακά πληροφοριακά συστήματα βασισμένα σε κανόνες. Επίσης χρησιμοποιεί κατάλληλες «συμβάσεις» (contracts) για την αντιστοίχιση των υπηρεσιών ιστού σε συγκεκριμένες κλήσεις εκτέλεσης.

Οι Baresi, Guinea and Plebani (2007) πρότειναν ένα μοντέλο σχεδίασης για των ορισμό εποπτευόμενων (supervised) διαδικασιών BPEL, όπου οι κανόνες εποπτείας παράγονται αυτόματα από τις πολιτικές που περιγράφουν τις χρησιμοποιούμενες υπηρεσίες. Οι πολιτικές αυτές χρησιμοποιούν τη γλώσσα περιγραφής περιορισμών WSCoL για τα μηνύματα που ανταλλάσσονται με τη διαδικασία. Εκφράζονται δε με το πρότυπο WS-Policy και ορίζονται μέσα σε έναν κατάλληλο περιγραφέα εγκατάστασης (deployment descriptor). Οι πολιτικές μετασηματίζονται κατόπιν σε κανόνες εποπτείας, οι οποίοι και εφαρμόζονται αυτόματα από μια υποδομή βασισμένη στον θεματοστρεφή προγραμματισμό. Η υποδομή

⁶⁰ "Kermeta workbench is a powerful metaprogramming environment based on an object-oriented DSL (Domain Specific Language) optimized for metamodel engineering" [Πηγή: <http://www.kermeta.org/>]

είναι υλοποιημένη με την επέκταση AspectJ που έχει ενσωματωθεί στον μηχανισμό εκτέλεσης διαδικασιών ActiveBPEL. Δεν υποστηρίζεται ωστόσο προσαρμοστικότητα κατά το χρόνο εκτέλεσης καθώς η AspectJ παρέχει μόνο στατική ενσωμάτωση των προσαρμογών (static weaving).

Οι Moser et al. (2008) παρουσίασαν το περιβάλλον VieDAME⁶¹ που είναι μια επέκταση του μηχανισμού εκτέλεσης ActiveBPEL και το οποίο επιτρέπει την παρακολούθηση των διαδικασιών BPEL σύμφωνα με χαρακτηριστικά Ποιότητας Υπηρεσίας (QoS attributes) και την αντικατάσταση των υπηρεσιών σύμφωνα με ορισμένες (επεκτάσιμες) στρατηγικές αντικατάστασης. Οι νέες υπηρεσίες θα πρέπει να είναι συντακτικά και σημασιολογικά ισοδύναμες με τις υφιστάμενες που πρόκειται να αντικαταστήσουν και να διαθέτουν συμβατή διεπαφή κλήσης (BPEL interface). Η αντικατάσταση μπορεί να γίνεται αυτόματα κατά την εκτέλεση, χωρίς να χρειαστεί να διακοπεί η λειτουργία του συστήματος. Αυτό επιτυγχάνεται χρησιμοποιώντας μια θεματοστρεφή προσέγγιση που παρεμβαίνει στα μηνύματα SOAP (που ανταλλάσσονται μεταξύ του μηχανισμού εκτέλεσης BPEL και των καλούμενων υπηρεσιών) και αλλάζει, όταν απαιτείται, τις υπηρεσίες που τελικά καλούνται, με μικρή επιβάρυνση στην επίδοση του μηχανισμού εκτέλεσης. Η κεντρική ιδέα του VieDAME είναι να προσαρμόζει τις διαδικασίες BPEL έχοντας όμως μια μη παρεμβατική συμπεριφορά όσον αφορά τα μοντέλα των διαδικασιών. Αυτό επιτρέπει την εναλλαγή των υπηρεσιών μιας διαδικασίας BPEL, κατά το χρόνο εκτέλεσης, χωρίς να τροποποιείται ο ορισμός (specification) της διαδικασίας ή των καλούμενων υπηρεσιών. Το περιβάλλον VieDAME υποστηρίζει τους μηχανισμούς ActiveBPEL 3.0⁶² και Apache ODE⁶³.

Οι Kongdenfha et al. (2006) προτείνουν ένα πλαίσιο εργασίας (framework) για προσαρμογή υπηρεσιών, το οποίο αποτελείται από μια κατηγοριοποίηση (taxonomy) των διαφόρων τύπων αναντιστοιχιών ή ανακολουθιών (mismatches) μεταξύ των προδιαγραφών και των υλοποιήσεων των υπηρεσιών, ένα αποθετήριο προτύπων θεμάτων (aspect-based templates) που χειρίζονται αυτόματα τις αναντιστοιχίες αυτές, και ένα εργαλείο που υποστηρίζει την αρχικοποίηση και εκτέλεσή τους μαζί με τις υλοποιήσεις των υπηρεσιών. Αυτή η προσέγγιση απαιτεί την τροποποίηση των διαδικασιών BPEL όταν ενσωματώνονται τα θέματα και εισάγονται στον βασικό τους κώδικα.

Οι David and Ledoux (2006) παρουσίασαν την Self-Adaptive FRActal compoNents ή SAFRAN. Πρόκειται αφενός για μία θεματοστρεφή προσέγγιση και αφετέρου για μια επέκταση του Fractal component model (Bruneton et al., 2006) για την ανάπτυξη πολιτικών «εκ των υστέρων» προσαρμογών (reactive adaptation policies). Οι πολιτικές αυτές ανιχνεύουν την εξέλιξη του πλαισίου των διαδικασιών κατά την εκτέλεση αυτών, και προσαρμόζουν την κύρια διαδικασία αλλάζοντας τις ρυθμίσεις της.

Μια άλλη ενδιαφέρουσα προσέγγιση είναι αυτή των Boukadi et al. (2008) και (2009) που προτείνουν μια αρχιτεκτονική βασιμμένη σε μια υψηλού επιπέδου δομή που καλείται

⁶¹ http://www.infosys.tuwien.ac.at/prototypes/VieDAME/VieDAME_index.html

⁶² <http://www.activebpel.org/>

⁶³ <http://ode.apache.org>

«Πεδίο Υπηρεσίας» (Service Domain). Αυτή ενορχηστρώνει ένα σύνολο από υπηρεσίες βασιζόμενη σε διαδικασίες BPEL. Η έννοια του πεδίου υπηρεσίας αναπτύχθηκε για να συμπληρώσει και επεκτείνει τη συνεργατική χρήση των Web services (e-business collaborations) ενώ έχει και επίγνωση του πλαισίου. Η επίγνωση αυτή εξασφαλίζεται με την επέκταση του μηχανισμού BPEL με δυνατότητες θεματοστρεφούς προγραμματισμού. Ένα πεδίο υπηρεσίας υλοποιείται ως ένας κόμβος διαδικασίας αποτελούμενος από μια μονάδα εισόδου (Entry Module), μια μονάδα διαχείρισης των πληροφοριών του πλαισίου (Context Management Module), μια μονάδα ενορχήστρωσης υπηρεσιών (Service Orchestration Module), και μια μονάδα ενεργοποίησης θεμάτων (Aspect Activator Module). Όσον αφορά το πλαίσιο των πεδίων υπηρεσίας, οι Boukadi et al. (2008) προτείνουν μια οντολογία μοντελοποιημένη με τη γλώσσα OWL. Σύμφωνα με την προσέγγισή τους η προσαρμογή είναι μία διαδικασία αποτελούμενη από τρία βήματα. (α) Το πρώτο είναι η «Ανίχνευση του πλαισίου», ήτοι έλεγχος των πληροφοριών αυτού κατά την εκτέλεση του συστήματος, για ανίχνευση πιθανών αλλαγών στη διαμόρφωσή του. Εκτελείται από τη μονάδα διαχείρισης πληροφοριών του πλαισίου, η οποία έχει αναπτυχθεί ως ένα Web service σε μια διαδικασία BPEL. (β) Δεύτερο βήμα είναι η «Ενεργοποίηση Θεμάτων», όπου εδώ γίνεται εισαγωγή και αφαίρεση προκαθορισμένων θεμάτων στην διαδικασία BPEL, χρησιμοποιώντας τη μονάδα ενεργοποίησης θεμάτων. Αυτή είναι μια επέκταση του μηχανισμού εκτέλεσης BPEL παρόμοια με την AO4BPEL. (γ) Τελευταίο βήμα είναι η «Ενημέρωση της αρχικής διαδικασίας BPEL» με την ενεργοποίηση του κατάλληλου θέματος, το οποίο και ενσωματώνεται στη διαδικασία BPEL για να δημιουργήσει την προσαρμοσμένη στο νέο πλαίσιο διαδικασία.

Τέλος οι Li, Liu and Bouguettaya (2009) εισάγουν τη χρήση σημασιολογικών πληροφοριών (semantics) για εφαρμογές με επίγνωση του πλαισίου. Ειδικότερα χρησιμοποιούν σημασιολογικές πληροφορίες για την ενσωμάτωση θεμάτων, αντίθετα με τις βασισμένες σε συντακτικά προσεγγίσεις όπως είναι η AO4BPEL. Περιγράφουν δε το πλαίσιο ως μια έννοια μιας οντολογίας ενώ μοντελοποιούν τις υπηρεσίες ως αυτόματα (automata).

8.2.6 Περιορισμοί

Παρότι οι παραπάνω προσεγγίσεις χρησιμοποιούν τον θεματοστρεφή προγραμματισμό για την πραγματοποίηση των προσαρμογών, δεν εκμεταλλεύονται όλες τους επαρκώς τις πληροφορίες του περιβάλλοντος και του πλαισίου (context) για την ανίχνευση των αναγκών και την πραγματοποίηση προσαρμογών, αξιοποιώντας σύγχρονες τεχνικές όπως είναι η επεξεργασία γεγονότων, το μοντέλο αποστολής / εγγραφής (Pub/Sub) ή άλλες τεχνικές που παρέχει η οδηγούμενη από γεγονότα αρχιτεκτονική (EDA). Οι τεχνικές αυτές μπορούν να μεγεθύνουν τη δυναμική των προσαρμοστικών συστημάτων και να εξασφαλίσουν ότι οι αναγκαίες προσαρμογές θα γίνονται εγκαίρως και ακριβώς όταν χρειάζονται. Όσον τώρα αφορά την προσέγγιση των Hermosillo et al. (2010), αυτή ακολουθεί το υπόδειγμα του θεματοστρεφούς προγραμματισμού (AOP paradigm) και της οδηγούμενης από γεγονότα αρχιτεκτονικής (EDA paradigm). Ωστόσο περιορίζεται μόνο σε γεγονότα που αφορούν την Ποιότητα Υπηρεσίας (QoS) ενώ δεν καλύπτει πιο γενικές καταστάσεις. Επίσης αφορά την υλοποίηση προσαρμογών σε διαδικασίες μοντελοποιημένες με τη γλώσσα BPEL, η οποία είναι γλώσσα ενορχήστρωσης υπηρεσιών ιστού κι όχι γλώσσα εκτέλεσης επιχειρησιακών

διαδικασιών όπως η BPMN 2.0 (που περιλαμβάνει όλα τα αλληλεπιδρώντα συστήματα, χρήστες και σχετικές έννοιες).

Επίσης, σχεδόν σε όλες τις προσεγγίσεις οι προσαρμογές φέρονται να πραγματοποιούνται όταν διαπιστωθεί η σχετική ανάγκη. Στην πράξη όμως θα ήταν πιο ορθό να παρέχεται η δυνατότητα αυτές να προτείνονται στους υπευθύνους των διαδικασιών και αυτοί κατόπιν να επιλέγουν αν θα τις αποδεχτούν για εφαρμογή στις διαδικασίες τους.

8.3 Προσέγγιση της Λύσης

Στην παρούσα διατριβή προσεγγίζουμε την έννοια της προσαρμογής μιας διαδικασίας ως λειτουργίες και απαιτήσεις (concerns) ανεξάρτητες μεν αλλά τεμνόμενες με αυτές της κύριας διαδικασίας. Έχουμε κάνει αυτή την επιλογή διότι αφενός οι προσαρμογές έχουν ενδεχομένως σκοπούς ανεξάρτητους από αυτούς της κύριας διαδικασίας και αφετέρου αφορούν ή επηρεάζουν πολλά μέρη της. Συνεπώς χρήση των θεμάτων (aspects) και του θεματοστρεφούς προγραμματισμός αποτελούν μια φυσική επιλογή.

Επίσης εστιάζουμε σε διαδικασίες αποτυπωμένες στη γλώσσα BPMN 2.0, η οποία μπορεί να χρησιμοποιηθεί τόσο για τη μοντελοποίηση και επικύρωση των μοντέλων των διαδικασιών όσο και για την εκτέλεσή τους. Επίσης υποστηρίζει οποιοδήποτε είδος διαδικασίας κι όχι μόνο διαδικασίες ενορχήστρωσης υπηρεσιών ιστού όπως η BPEL.

Για την πραγματοποίηση των προσαρμογών επιλέγουμε μια προσέγγιση παρόμοια με την AO4BPMN (Charfi et al., 2010) καθότι επιτρέπει το σχεδιασμό και την υλοποίηση προσαρμογών, με την ελάχιστη δυνατή παρέμβαση στα μοντέλα των ήδη υφιστάμενων διαδικασιών. Επιπλέον, οι διαδικασίες είναι δυνατό να μοντελοποιούνται χωρίς να λαμβάνονται υπόψη (στον κυρίως σχεδιασμό) τυχόν λειτουργίες ή τροποποιήσεις (concerns) που χρειάζονται μόνο όταν οι συνθήκες το απαιτήσουν.

Συνεπώς, στην παρούσα διατριβή υιοθετούμε τη βασική προσέγγιση και τη σημειογραφία της AO4BPMN και τις προσαρμόζουμε για χρήση με τη γλώσσα BPMN 2.0. Την νέα μέθοδο που προκύπτει την καλούμε FlexiBPMN2.0.

8.3.1 Η μέθοδος FlexiBPMN2.0

Η AO4BPMN προτάθηκε αρχικά ως μια μέθοδος για τη μοντελοποίηση των προσαρμογών διαδικασιών κατά το χρόνο σχεδιασμού (design-time). Εξάλλου και η πρώτη έκδοση της BPMN αφορούσε μόνο τη μοντελοποίηση διαδικασιών κι όχι την εκτέλεσή τους. Ωστόσο η πιο πρόσφατη έκδοσή της (ήτοι η BPMN 2.0) είναι εκτελέσιμη. Τα XML αρχεία που παράγονται κατά τη μοντελοποίηση και που περιγράφουν τις διαδικασίες σύμφωνα με τη σημειογραφία, τους κανόνες και τις συμβάσεις της BPMN 2.0, μπορούν κατόπιν να εκτελεστούν από έναν συμβατό μηχανισμό εκτέλεσης ροών εργασιών και να παράγουν αποτελέσματα. Στην προσέγγισή μας επεκτείνουμε την AO4BPMN έτσι ώστε να μπορεί να μοντελοποιεί τις προσαρμογές των διαδικασιών κατά το χρόνο σχεδιασμού (design time) αλλά και να τις εφαρμόζει κατά την εκτέλεση (execution time). Ειδικότερα εστιάζουμε στην «ευελιξία κατά την αλλαγή» (flexibility by change), όπως αυτή περιγράφεται από τους Schonenberg, Mans, et al. (2008) (δείτε ενότητα 3.1), ώστε να είναι εφικτή η μεταβολή του

μοντέλου μιας διαδικασίας ακόμη και όταν υπάρχουν εκτελούμενα στιγμιότυπα αυτής. Η μεταβολή αυτή μπορεί να εφαρμόζεται και επιλεκτικά σε συγκεκριμένα στιγμιότυπα της διαδικασίας. Επιπρόσθετα, στο πλαίσιο αυτής της διατριβής ασχολούμαστε και με τη δυναμική δημιουργία θεμάτων, δυνατότητα που απουσιάζει από την AO4BPMN, καθώς και την εφαρμογή αυτών (weaving) κατά την εκτέλεση. Αυτό επιτυγχάνεται με την πραγματοποίηση κατάλληλων επεκτάσεων σε ένα μηχανισμό εκτέλεσης ρών εργασιών συμβατό με την BPMN 2.0, προκειμένου αυτός να μπορεί να υποστηρίζει και να εκτελεί τα θέματα προσαρμογών.

Για την ανίχνευση των αναγκών και των καταστάσεων που χρίζουν προσαρμογής των διαδικασιών, επιλέγουμε να αξιοποιήσουμε την πλατφόρμα και τις δυνατότητες που παρέχουν τα SANs, τα οποία παρουσιάστηκαν αναλυτικά στα κεφάλαια 5 και 6. Κατάλληλα σχεδιασμένα SANs μπορούν να παρακολουθούν την εξέλιξη των διαδικασιών και να προτείνουν (όταν απαιτείται) αντίστοιχες προσαρμογές. Χρησιμοποιούνται δε για να ορίζουν και να ανιχνεύουν τις καταστάσεις αυτές. Εδώ τα SANs λειτουργούν ως ένα μέσο παρακολούθησης της καλής εκτέλεσης των διαδικασιών και προκαλούν την ενεργοποίηση κατάλληλων προσαρμογών όταν οι καταστάσεις το απαιτούν.

Τα SANs αξιοποιούν πληροφορίες του περιβάλλοντος και του πλαισίου (context) των διαδικασιών χρησιμοποιώντας γεγονότα. Έτσι προκειμένου να λαμβάνουν την απαραίτητη πληροφόρηση εγγράφονται σε κατάλληλες ροές γεγονότων (event streams) ακολουθώντας το υπόδειγμα αποστολής/εγγραφής (Pub/Sub) ενώ κατόπιν τις αναλύουν με χρήση τεχνικών επεξεργασίας σύνθετων γεγονότων (CEP). Τα γεγονότα που αξιοποιούνται από τα SANs, στο πλαίσιο της FlexiBPMN2.0, μπορούν να προέρχονται από διάφορες πηγές και στοιχεία λογισμικού, συμπεριλαμβανομένου του μηχανισμού εκτέλεσης ρών εργασιών (workflow events). Η επεξεργασία σύνθετων γεγονότων παρέχει αρκετά προηγμένες δυνατότητες παρακολούθησης και αντίδρασης στο περιβάλλον των διαδικασιών, αφού επιτρέπει την προώθηση, φιλτράρισμα, συγκερασμό και συνδυασμό των πρωτογενών γεγονότων σε άλλα συνθετότερα που εκφράζουν καταστάσεις που χρίζουν προσαρμογών (Hinze, Sachs and Buchmann, 2009). Στην παρούσα διατριβή η επεξεργασία γεγονότων χρησιμοποιείται κυρίως για την παρακολούθηση της καλής εκτέλεσης των διαδικασιών, την ανίχνευση τυχόν προβλημάτων λειτουργίας ή υποβάθμιση της ποιότητας του παραγόμενου αποτελέσματος, προκειμένου να εκκινήσει μια διεργασία αναζήτησης μιας κατάλληλης πρότασης προσαρμογής. Με τη χρήση των SANs αποκομίζονται οφέλη και πλεονεκτήματα που έχουν ήδη αναφερθεί στην ενότητα 7.2, όπως η αποτελεσματικότερη και οικονομικότερη χρήση των ρών γεγονότων.

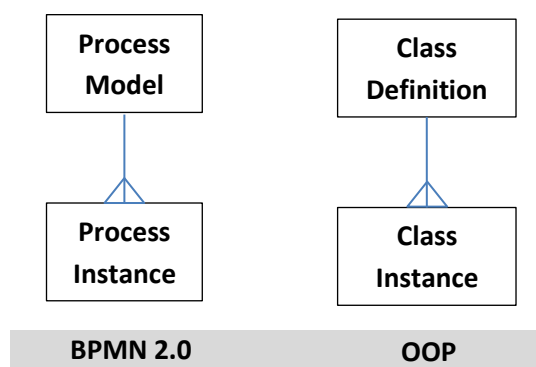
Τέλος επισημαίνουμε ότι η μέθοδος AO4BPMN 2.0 των Witteborg et al. (2014), είναι ότι πιο κοντινό έχουμε δει στην προτεινόμενη μέθοδο. Ωστόσο υπάρχουν σημαντικές διαφορές μεταξύ τους με κυριότερη το ότι η μεν πρώτη εφαρμόζεται πριν την εκτέλεση των διαδικασιών (κατά το σχεδιασμό ή κάποια στιγμή πριν την εκτέλεση) ενώ η δεύτερη εφαρμόζεται κατά την εκτέλεση των διαδικασιών, λαμβάνοντας έτσι υπόψη τις συνθήκες που επικρατούν την εκάστοτε στιγμή. Επιπλέον η προσαρμογή οδηγείται από γεγονότα (μέσω της χρήσης SANs όπως προαναφέρθηκε), ενώ η υλοποίηση της μεθόδου έχει γίνει χρησιμοποιώντας ένα πραγματικό μηχανισμό εκτέλεσης διαδικασιών BPMN2 κι όχι κάποιο

περιβάλλον μοντελοποίησης (όπως οι Witteborg et al. (2014) που χρησιμοποιούν το Eclipse BPMN2 Modeler).

8.3.2 Βασικές έννοιες της γλώσσας BPMN 2.0

Σε αυτό το σημείο κρίνεται σκόπιμο να αναφέρουμε λίγα πράγματα για τις διαδικασίες που μοντελοποιούνται με τη γλώσσα BPMN 2.0. Αυτές αποτελούνται από απλές εργασίες (tasks/activities), υπο-διαδικασίες (sub-processes), κόμβους αποστολής ή λήψης γεγονότων (send/receive event nodes) και πύλες αποφάσεων (gateways), συνδεδεμένα μεταξύ τους είτε σειριακά, είτε παράλληλα, είτε σε μικτές τοπολογίες. Ουσιαστικά οι εκτελέσιμες οντότητες στην BPMN 2.0 είναι οι εργασίες (tasks), που φέρουν εις πέρας κάποια συγκεκριμένη ενέργεια, και οι κόμβοι γεγονότων που στέλνουν ή λαμβάνουν μηνύματα και γεγονότα σε άλλες διαδικασίες ή στο περιβάλλον.

Μια σημαντική διάκριση είναι αυτή μεταξύ του μοντέλου μιας διαδικασίας (process model), δηλαδή της τυπικής περιγραφής της, και του στιγμιότυπου εκτέλεσης μιας διαδικασίας (process instance) που είναι ουσιαστικά η διεκπεραίωση στην πράξη των εργασιών που περιγράφονται στο μοντέλο με πραγματικά δεδομένα. Το μοντέλο μιας διαδικασίας είναι μοναδικό, ωστόσο μπορούν να υπάρχουν ταυτόχρονα πολλαπλά στιγμιότυπα αυτού. Είναι ακριβώς ανάλογο με τις κλάσεις αντικειμένων (classes) και τα στιγμιότυπα των κλάσεων (instances) στον αντικειμενοστραφή προγραμματισμό (Object-Oriented Programming).



Σχήμα 8-1. Μοντέλο και στιγμιότυπα διαδικασιών

8.3.3 Βασικά σημεία της FlexiBPMN2.0

Κεντρικές έννοιες τόσο στην AO4BPMN όσο και στην FlexiBPMN2.0 είναι τα *Θέματα (Aspects)*, τα *Σημεία Τομών (Pointcuts)* και οι *Συμβουλές (Advices)*. Τα θέματα είναι σύνθετες δομές αποτελούμενες από ένα ή περισσότερα σημεία τομών και μία συμβουλή. Έχουν δε ένα συγκεκριμένο σκοπό και παρέχουν μια λειτουργία ανεξάρτητη μεν αλλά «τεμνόμενη» (cross-cutting) με αυτές της κύριας διαδικασίας. Για παράδειγμα, η αντικατάσταση των εργασιών εκτύπωσης μιας διαδικασίας που αφορά έγκριση δανείου, με ισοδύναμες εργασίες που αποστέλλουν emails με το αντίστοιχο περιεχόμενο. Προφανώς ο σκοπός της κύριας διαδικασίας είναι η έκδοση του δανείου, ενώ της προσαρμογής η οικονομία χαρτιού.

Τα σημεία τομών είναι οι εκείνες δομές που επιτρέπουν την επιλογή των επιθυμητών σημείων και μερών στα μοντέλα των διαδικασιών (joint points), στα οποία μπορεί να συμβεί κάποια προσαρμογή. Η επιλογή αυτή μπορεί να γίνεται είτε με απλή επισήμανση των εν λόγω σημείων σύνδεσης (joint points) ή ακόμη και με χρήση κατάλληλων ερωτημάτων (queries), εκφράσεων (expressions) και συνθηκών (conditions). Σύμφωνα με τα προηγούμενα μπορεί να οριστεί η έννοια των δυναμικών και εξαρτώμενων απ' την κατάσταση σημείων τομής (dynamic and situation-dependent pointcuts). Αυτό επιτυγχάνεται συνδυάζοντας κατάλληλα τα σημεία τομών όπως τα περιγράψαμε παραπάνω με τα SANs που παρακολουθούν την εκτέλεση των διαδικασιών και προτείνουν τα αντίστοιχα θέματα προσαρμογών.

Οι συμβουλές απ' την άλλη, είναι διαδικασίες ή τμήματα διαδικασιών (process fragments) που υλοποιούν τις προσαρμογές, σύμφωνα πάντα με τους σκοπούς των θεμάτων που τις περιέχουν. Στο προηγούμενο παράδειγμα σημεία τομών θα ήταν όλες οι εργασίες εκτύπωσης και συμβουλή θα ήταν η εργασία αποστολής emails. Ο σκοπός μιας συμβουλής είναι να αντικαταστήσει ή συμπληρώσει τα επιλεγμένα σημεία τομών μιας διαδικασίας με μια νέα εργασία (ή τμήμα διαδικασίας).

Πρακτικά κατά την εκτέλεση μιας διαδικασίας από έναν μηχανισμό εκτέλεσης ροών εργασιών συμβατό με την FlexiBPMN2.0, κάθε φορά που η ροή εκτέλεσης φτάνει σε μια εργασία ή ένα κόμβο γεγονότων ή μια πύλη απόφασης, και προτού αυτή εκτελεστεί, ο μηχανισμός εξετάζει αν το εν λόγω σημείο αποτελεί σημείο τομής για κάποιο θέμα. Αν ισχύει κάτι τέτοιο τότε η συμβουλή του αντίστοιχου θέματος θα εκτελεστεί στη θέση της εργασίας ή του κόμβου γεγονότος ή της πύλης απόφασης που ακολουθεί. Επίσης, μετά την ολοκλήρωση της συμβουλής η ροή εκτέλεσης μπορεί να συνεχίζει είτε από το σημείο όπου παρεμβλήθηκε (εισήχθη) η συμβουλή ή από άλλο σημείο σύνδεσης, παραλείποντας έτσι έναν ή περισσότερους κόμβους της διαδικασίας (workflow reroute). Ακόμη είναι δυνατό να υπάρχουν περισσότερα από ένα θέματα που αφορούν ένα συγκεκριμένο σημείο. Τότε όλες οι αντίστοιχες συμβουλές θα πρέπει να εκτελεστούν με την κατάλληλη σειρά.

Ένας μηχανισμός εκτέλεσης ροών εργασιών συμβατός με την FlexiBPMN2.0 θα πρέπει να παράγει γεγονότα που αφορούν την εκτέλεση των διαφόρων μερών των διαδικασιών (π.χ. έναρξη/ολοκλήρωση/διακοπή/σφάλμα μιας εργασίας, έναρξη/ολοκλήρωση/διακοπή ενός στιγμιότυπου μιας διαδικασίας), τα οποία θα προωθεί στο μηχανισμό των SANs. Αντίστοιχα θα πρέπει να λαμβάνει από τα SANs γεγονότα που αφορούν προτάσεις προσαρμογών των διαδικασιών. Η αποδοχή ή μη των προτάσεων μπορεί να επαφίεται στον υπεύθυνο της διαδικασίας ή μπορεί να γίνονται αυτόματα δεκτές.

Αναφορικά με τη δημιουργία προτάσεων προσαρμογών από τα SANs, μπορούν να οριστούν υψηλού επιπέδου προσεγγίσεις για τη δημιουργία των προτάσεων προσαρμογών. Αυτές καλούνται *στρατηγικές προσαρμογών (adaptation strategies)* και αφορούν κυρίως τον τρόπο μοντελοποίησης των προσαρμογών αυτών καθ'αυτών και δευτερευόντως των SANs που τις δημιουργούν και τις προτείνουν. Οι στρατηγικές προσαρμογών μπορεί να κατηγοριοποιηθούν στις εξής κατηγορίες:

- **Στρατηγικές προσαρμογών οδηγούμενες από παραμέτρους ποιότητας ή από την λειτουργικότητα (QoS-Driven vs. Functional-Driven adaptation strategies):** Αυτή η κατηγοριοποίηση εστιάζει στο λόγο της προσαρμογής ανάλογα με το αν αφορά κάποιο θέμα σχετικό με την ποιότητα υπηρεσίας (QoS-driven issue) όπως για παράδειγμα την παραβίαση μιας SLA, ή αφορά κάποιο λειτουργικό θέμα (functional issue), όπως για παράδειγμα τη δυσλειτουργία μιας εργασίας με βάση το αποτέλεσμα που παράγει.
- **Στρατηγικές με γενική ή τοπική εφαρμογή (Global vs. Local adaptation strategies):** Αυτή η κατηγοριοποίηση αφορά στο εύρος (έκταση) των προσαρμογών. Μια προσαρμογή μπορεί να ονομαστεί “Global” (Γενική) αν οι συμβουλές που την υλοποιούν επηρεάζουν όλα τα στιγμιότυπα όλων των διαδικασιών, “Global” όταν επηρεάζουν όλα τα στιγμιότυπα κάποιας συγκεκριμένης διαδικασίας και “Local” (Τοπική) όταν επηρεάζουν μόνο συγκεκριμένα στιγμιότυπα.

8.3.4 Καινοτομίες της FlexiBPMN2.0

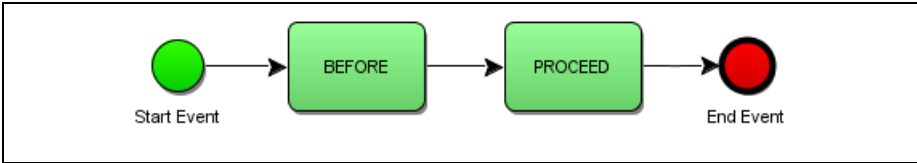
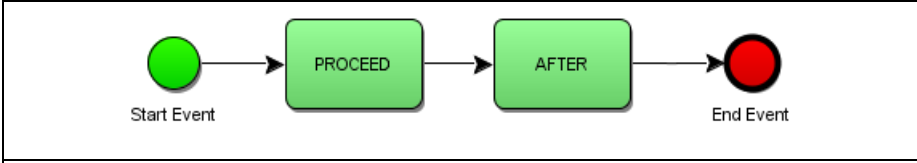
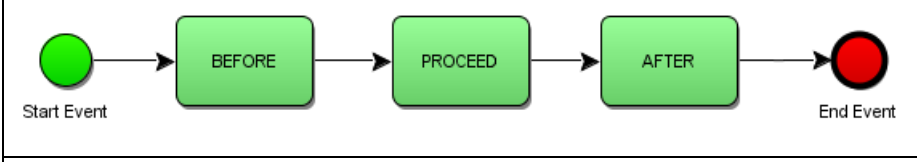
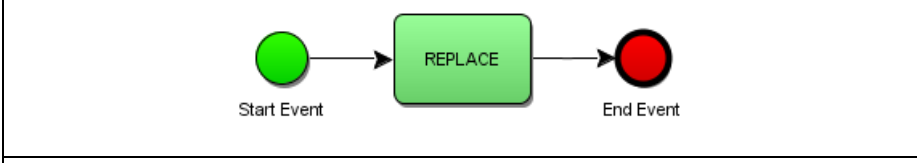
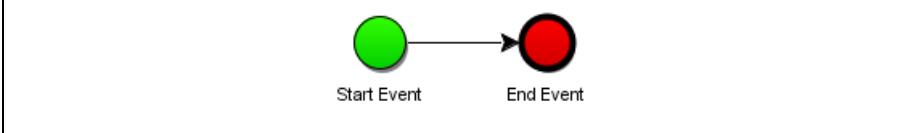
Συνοψίζουμε τις βασικές καινοτομίες της μεθόδου FlexiBPMN2.0 σε σχέση με άλλες μεθόδους θεματοστρεφούς προσαρμογής και ιδίως με την AO4BPMN.

- Προσαρμογή κατά την εκτέλεση των διαδικασιών κι όχι μόνο κατά το σχεδιασμό (όπως η AO4BPMN).
- Οδηγούμενη από γεγονότα ανίχνευση και αντίληψη των αναγκών και των καταστάσεων που χρίζουν προσαρμογής μιας διαδικασίας. Αξιοποίηση μεθόδων και τεχνικών της αρχιτεκτονικής EDA.
- Παραγωγή προτάσεων προσαρμογών που απαιτούν την έγκριση – αποδοχή του υπεύθυνου της διαδικασίας, αντί για απευθείας εφαρμογή των προσαρμογών. Φυσικά, αν αυτό είναι επιθυμητό, οι προτάσεις προσαρμογών μπορεί να γίνονται αυτόματα δεκτές.
- Εστιάζει σε διαδικασίες αποτυπωμένες με τη γλώσσα BPMN2.0, η οποία είναι εκτελέσιμη γλώσσα (σε αντίθεση με την BPMN1.0) κι επίσης ευρύτερης εφαρμογής σε σχέση με την BPEL (που εστιάζει στην ενορχήστρωση υπηρεσιών ιστού).

8.3.5 Είδη προσαρμογών

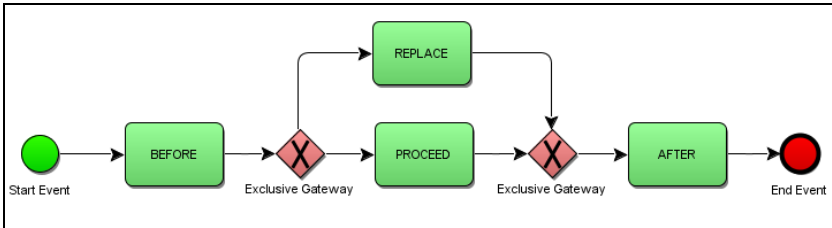
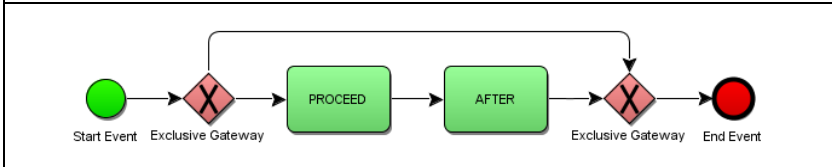
Ορίζονται πέντε στοιχειώδεις τύποι προσαρμογών (Πίνακας 8-1) που μπορούν να εφαρμοστούν σε ένα σημείο τομής μιας διαδικασίας. Συγκεκριμένα, μπορούν να εισαχθούν νέες εργασίες πριν από το προς εκτέλεση σημείο τομής, μετά από αυτό, και πριν και μετά (πέριξ) ή μπορούν να το αντικαταστήσουν. Μια οριακή περίπτωση είναι η παράλειψη του σημείου τομής (αντικαθίσταται από την «κενή» συμβουλή). Οι βασικοί αυτοί τύποι προσαρμογών απεικονίζονται στα ακόλουθα διαγράμματα με χρήση σημειογραφίας της BPMN 2.0. Η εργασία Proceed αναπαριστά το προς εκτέλεση σημείο τομής (μέρος) της κύριας διαδικασίας. Σημειώνουμε ότι ένα σημείο τομής μπορεί να είναι είτε μια απλή εργασία είτε να είναι σύνθετο, δηλαδή να αποτελείται από ένα τμήμα της διαδικασίας (process fragment).

Πίνακας 8-1. Βασικοί τύποι προσαρμογών

	<p>ΠΡΙΝ (BEFORE)</p>
	<p>ΜΕΤΑ (AFTER)</p>
	<p>ΠΕΡΙΞ ή ΠΡΙΝ και ΜΕΤΑ (AROUND)</p>
	<p>ΑΝΤΙΚΑΤΑΣΤΑΣΗ (REPLACE)</p>
	<p>ΠΑΡΑΛΕΙΨΗ (BYPASS)</p>

Η κάθε εργασία προσαρμογής στα διαγράμματα (με την ένδειξη BEFORE, AFTER, REPLACE) ενδέχεται στην πράξη να είναι ένα ολόκληρο τμήμα της διαδικασίας ή υπο-διαδικασία. Επίσης οι συμβουλές μπορεί να συμπεριλαμβάνουν και μια ειδική εργασία-κόμβο που καλείται “Proceed”. Η χρήση της Proceed ουσιαστικά υποδηλώνει ότι στο συγκεκριμένο σημείο της προσαρμογής πρέπει να εκτελεστεί το σημείο τομής της αρχικής διαδικασίας, το οποίο αντικαταστάθηκε απ’ τη συμβουλή. Κατ’ αυτό τον τρόπο μια συμβουλή μπορεί να υλοποιεί όλους τους βασικούς τύπους προσαρμογών, όπως την εισαγωγή νέων εργασιών πριν από τα σημεία τομής (before type), μετά από αυτά (after type), και πριν και μετά (around type), ή την αντικατάσταση αυτών (replace type) όταν δεν συμπεριλαμβάνεται η Proceed. Είναι ακόμη δυνατή η μίξη των βασικών τύπων προσαρμογής σε πιο σύνθετους, όπως στα παραδείγματα που δίνονται ακολούθως (Πίνακας 8-2).

Πίνακας 8-2. Παραδείγματα σύνθετων τύπων προσαρμογών

	<p>ΑΝΤΙΚΑΤΑΣΤΑΣΗ – ΠΕΡΙΞ (REPLACE – AROUND)</p>
	<p>ΜΕΤΑ – ΠΑΡΑΛΕΙΨΗ (AFTER – BYPASS)</p>

8.3.6 Τρόποι εφαρμογής των προσαρμογών

Η αξιοποίηση των SANs στην FlexiBPMN2.0 για την πραγματοποίηση των προσαρμογών μπορεί να επιτευχθεί με διάφορους τρόπους, αναλόγως των αναγκών της εκάστοτε εφαρμογής. Διακρίνουμε τους εξής:

- **Ασύγχρονος τρόπος (Asynchronous mode).** Σύμφωνα με τον τρόπο αυτό τα θέματα μοντελοποιούνται και καταχωρούνται σε κατάλληλο αποθετήριο θεμάτων (aspect repository). Τα SANs ενεργοποιούν και απενεργοποιούν τα θέματα του aspect repository ή προτείνουν την ενεργοποίηση και απενεργοποίησή τους. Ο μηχανισμός εκτέλεσης ροών εργασιών (FlexiBPMN2.0 workflow engine), ο οποίος λειτουργεί παράλληλα και ανεξάρτητα από τον SAN engine, πριν από κάθε εκτελέσιμο κόμβο της διαδικασίας ελέγχει αν υπάρχει κάποιο ενεργό σημείο τομής (στο aspect repository) που να αφορά το εν λόγω σημείο. Αν βρει κάποιο τότε εκτελεί την αντίστοιχη συμβουλή. Αξίζει ακόμη να αναφέρουμε ότι είναι δυνατό τα θέματα να δημιουργούνται και καταχωρούνται στο aspect repository δυναμικά.
- **Σύγχρονος τρόπος (Synchronous mode).** Τα SANs προσδιορίζουν δυναμικά αν κάποιο σημείο της διαδικασίας αποτελεί σημείο τομής κάποιου θέματος και αν πρέπει να εκτελεστεί κάποια συμβουλή. Αυτό επιτυγχάνεται με τον εξής τρόπο. Ο μηχανισμός εκτέλεσης, πριν από την εκτέλεση ενός κόμβου της διαδικασίας, ερωτά σχετικώς το κατάλληλο SAN. Αν υπάρχει ενεργό σημείο τομής τότε ο μηχανισμός SANs επιστρέφει την συμβουλή που θα πρέπει να εκτελεστεί, διαφορετικά η εκτέλεση συνεχίζεται κανονικά. Ο μηχανισμός SANs πρέπει να διατηρεί όλη την απαραίτητη γνώση τόσο της κατάστασης του περιβάλλοντος και της διαδικασίας (situation), όσο και των διαθέσιμων θεμάτων κι έτσι να κατευθύνει κατάλληλα τη διαδικασία προσαρμογής.
- **Τρόπος Παράκαμψης (Override mode).** Αυτός ο τρόπος αποτελεί παραλλαγή του προηγούμενου. Εδώ όμως τα SANs είναι που εκτελούν τις συμβουλές κι όχι ο μηχανισμός εκτέλεσης των ροών εργασιών. Τα SANs εκτελούν όλες τις εργασίες προσαρμογής χρησιμοποιώντας τα γεγονότα ροής εκτέλεσης (workflow events) που στέλνει ο μηχανισμός εκτέλεσης. Καθόσον χρόνο εκτελείται μια συμβουλή από τα SANs, η αντίστοιχη διαδικασία «παγώνει» προσωρινά στον μηχανισμό εκτέλεσης ροής εργασιών.

8.4 Η Πλατφόρμα Προτάσεως Προσαρμογών (ARP)

Για την εφαρμογή της προτεινόμενης μεθόδου FlexiBPMN2.0, σχεδιάστηκε και αναπτύχθηκε μια πλατφόρμα «παραγωγής προτάσεων προσαρμογών» ικανής να αναγνωρίζει τις καταστάσεις του περιβάλλοντος και τις ανάγκες των διαδικασιών, από γεγονότα που λαμβάνει και επεξεργάζεται, και να προτείνει αντίστοιχες θεματοστρεφείς προσαρμογές. Η πλατφόρμα αυτή καλείται Adaptation Recommendation Platform (ARP). Αποτελείται από δύο κύρια μέρη, το σύστημα επίγνωσης κατάστασης (SAS) που προτείνει τις προσαρμογές και της μονάδας προσαρμοστικής εκτέλεσης των διαδικασιών. Το SAS εκτελεί SANs κατάλληλα σχεδιασμένα ώστε να παρακολουθούν συγκεκριμένες διαδικασίες – στόχους (target processes) και να προτείνουν θεματοστρεφείς προσαρμογές όταν αυτό απαιτείται.

Το δεύτερο σύστημα πρόκειται για ένα μηχανισμό εκτέλεσης ροών εργασιών (workflow engine) συμβατό με τη γλώσσα BPMN 2.0 και την μέθοδο FlexiBPMN2.0. Ο μηχανισμός αυτός ονομάζεται «Σύστημα Προσαρμοστικής Εκτέλεσης Διαδικασιών» ή Adaptive Process Execution System (APES) και έχει τη δυνατότητα να εκτελεί διαδικασίες αποτυπωμένες σε BPMN 2.0, εφαρμόζοντάς τους προσαρμογές που λαμβάνει κατά το χρόνο εκτέλεσης. Το APES συνεργάζεται με τα SANs προκειμένου αυτά να του παρέχουν τις αναγκαίες προτάσεις προσαρμογών.

Η πλατφόρμα προτάσεων προσαρμογών καθαυτή δεν παρέχει κάποια γενικής χρήσης μέθοδο ή αλγόριθμο παραγωγής προτάσεων, αλλά δίνει τα εργαλεία για το σχεδιασμό και την υλοποίηση εξειδικευμένων κατά περίπτωση «μοντέλων προτάσεων προσαρμογών» υπό τη μορφή SANs. Οι λόγοι που κάναμε αυτή την επιλογή αναλύονται στην επόμενη ενότητα. Αναφέρουμε συμπληρωματικά ότι και οι Charfi et al. (2009) ακολούθησαν μια παρόμοια προσέγγιση.

8.4.1 Χαρακτηριστικά

Προτού παρουσιάσουμε αναλυτικότερα την πλατφόρμα, αναφέρουμε τις προδιαγραφές και τα επιθυμητά χαρακτηριστικά που πρέπει να διαθέτει. Επίσης αναφέρουμε ορισμένες προϋποθέσεις και συμβάσεις για τη χρήση της.

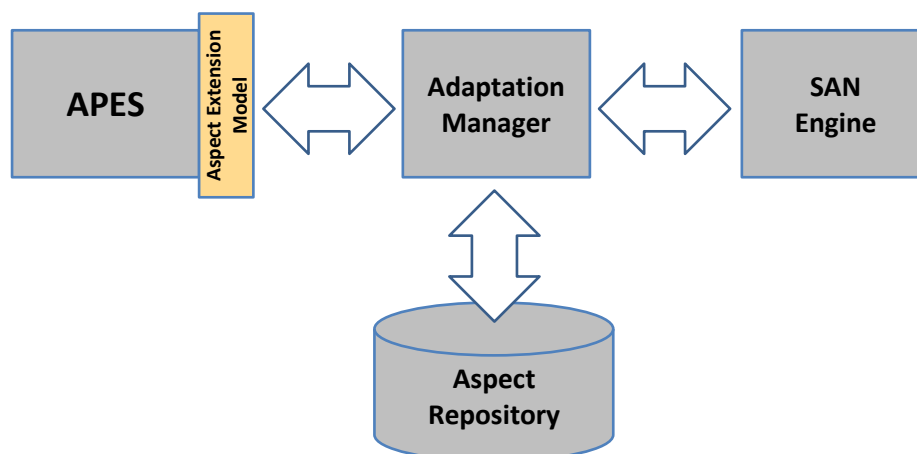
- **Προϋπόθεση 1 – «Τα πάντα είναι διαδικασίες».** Οι εφαρμογές, SBAs, υπηρεσίες ιστού ή οτιδήποτε άλλο αποτελεί «αντικείμενο» προσαρμογής (adaptation target), θα πρέπει να είναι υλοποιημένο ως διαδικασίες μοντελοποιημένες με τη γλώσσα BPMN 2.0. Ειδικότερα για τις βασισμένες σε υπηρεσίες εφαρμογές (SBAs), οι διαδικασίες BPMN μπορεί να λειτουργούν ως «μηχανισμός ενορχήστρωσης» τους (orchestration engine), δηλαδή να ελέγχουν και συντονίζουν την αλληλεπίδραση μεταξύ των υπηρεσιών που τις απαρτίζουν.
- **Προϋπόθεση 2 – «Εστιάζουμε στον βασικό σκοπό της διαδικασίας. Όλα τα άλλα είναι Θέματα».** Ο σχεδιασμός των διαδικασιών πρέπει να εστιάζει στους βασικούς σκοπούς τους. Δεν θα πρέπει να εξετάζει κάθε πιθανή περίπτωση προβλήματος και να την ενσωματώνει στη βασική ροή εργασιών. Η αντιμετώπιση των προβλημάτων που ανακύπτουν αποτελεί αντικείμενο των προσαρμογών, συνεπώς αυτές πρέπει να μοντελοποιούνται ξεχωριστά, ως θέματα, χρησιμοποιώντας τις αρχές και τις τεχνικές του θεματοστρεφούς προγραμματισμού. Μπορεί βέβαια να υπάρξουν και εξαιρέσεις όπου ο χειρισμός κάποιων ειδικών περιπτώσεων θα πρέπει να γίνει στην κύρια διαδικασία.
- **Προϋπόθεση 3 – «Επικοινωνία με γεγονότα».** Θα πρέπει να υπάρχει διαθέσιμος ένας δίαυλος γεγονότων με δυνατότητα Αποστολής/Εγγραφής (Pub/Sub event bus), απ' όπου θα διακινούνται τα γεγονότα που προέρχονται, προορίζονται ή αφορούν τις διαδικασίες και το περιβάλλον τους καθώς και την πλατφόρμα ARP. Επίσης είναι επιθυμητό να υπάρχει κι ένας μηχανισμός Επεξεργασίας Σύνθετων Γεγονότων (CEP engine) για τον συνδυασμό των απλών γεγονότων σε συνθετότερα που περιέχουν πληροφορίες ανώτερου σημασιολογικού επιπέδου. Τα γεγονότα μπορεί επίσης να είναι γεγονότα της ροής εργασιών ή γεγονότα πρότασης προσαρμογών.

Προδιαγραφές και επιθυμητά χαρακτηριστικά της πλατφόρμας προτάσεων προσαρμογών:

- Οι προτάσεις προσαρμογών πρέπει να διακινούνται ως κατάλληλα μορφοποιημένα γεγονότα (recommendation events), τα οποία θα αποστέλλονται στις κατάλληλες ροές γεγονότων – προτάσεων.
- Τα συστήματα παραγωγής προτάσεων πρέπει να μοντελοποιούνται ως SANs και να εκτελούνται από τον μηχανισμό εκτέλεσης SANs που περιλαμβάνει η πλατφόρμα. Είναι ευθύνη του σχεδιαστή να δημιουργήσει τα SANs και να τα τροφοδοτήσει στον μηχανισμό SANs, μαζί με όλα τα συνοδευτικά αρχεία που τυχόν θα χρειαστούν (πρότυπα θεμάτων, συμβουλών ή γεγονότων προτάσεων).
- Ο μηχανισμός εκτέλεσης ροών εργασιών (APES) πρέπει να αποστέλλει γεγονότα που αφορούν την εξέλιξη της εκτέλεσης των διαδικασιών (workflow events), σε κατάλληλη ροή γεγονότων (event topic). Τα γεγονότα αυτά θα πρέπει κατ' ελάχιστο να αφορούν:
 - την έναρξη εκτέλεσης ενός νέου στιγμιότυπου μιας διαδικασίας
 - τον τερματισμό ή διακοπή ενός στιγμιότυπου μιας διαδικασίας
 - την έναρξη εκτέλεσης μιας εργασίας ενός στιγμιότυπου διαδικασίας
 - την ολοκλήρωση ή διακοπή μιας εργασίας ενός στιγμιότυπου διαδικασίας
- Οι προτάσεις που παράγουν τα μοντέλα προτάσεων προσαρμογών (τα SANs δηλαδή) θα πρέπει να αναφέρουν ποιο θέμα (aspect) θα πρέπει να εφαρμοστεί και σε ποια στιγμιότυπα ποιας διαδικασίας.
- Οι προτάσεις μπορεί να αφορούν τόσο την ενεργοποίηση και εφαρμογή μιας προσαρμογής όσο και την απενεργοποίηση ή κατάργηση αυτής. Οπότε θα πρέπει να περιλαμβάνεται σχετική ένδειξη.
- Τόσο ο SAN engine και το APES, όσο και ο δίαυλος γεγονότων και ο CEP engine θα πρέπει να έχουν αρκετά γρήγορη απόκριση ούτως ώστε να αντιδρούν γρήγορα στην εμφάνιση «ειδικών καταστάσεων» (προβλήματα ή αλλαγές του περιβάλλοντος).

8.4.2 Αρχιτεκτονική της πλατφόρμας

Στη συνέχεια παρουσιάζεται η λογική αρχιτεκτονική (conceptual architecture) της πλατφόρμας προτάσεων προσαρμογών (Σχήμα 8-2). Τα μέρη που την απαρτίζουν είναι ο μηχανισμός εκτέλεσης SANs, το συμβατό με τις BPMN 2.0 και FlexiBPMN2.0 σύστημα προσαρμοστικής εκτέλεσης διαδικασιών (APES), ο διαχειριστής προσαρμογών (adaptation manager) και το αποθετήριο θεμάτων (aspect repository).



Σχήμα 8-2. Λογική αρχιτεκτονική της πλατφόρμας προτάσεων προσαρμογών

Μηχανισμός SANs (SAN Engine). Είναι το λογισμικό που εκτελεί τα SANs που απαρτίζουν το εκάστοτε σύστημα παραγωγής προτάσεων προσαρμογών για τις διαδικασίες που εκτελούνται στο APES. Αποτελείται από το αποθετήριο των SANs (SAN repository), το υποσύστημα εκτέλεσης των SANs (SAN execution component), το υποσύστημα διαχείρισης των γεγονότων (Event management component) και το υποσύστημα διαχείρισης των πληροφοριών του πλαισίου (Context management component). Περισσότερες πληροφορίες για τα SANs και τον μηχανισμό των SANs υπάρχουν στα αντίστοιχα κεφάλαια (5 και 6).

Σύστημα Προσαρμοστικής Εκτέλεσης Διαδικασιών (APES). Είναι το σύστημα εκτέλεσης ροών εργασιών (workflow engine) το οποίο εκτελεί διαδικασίες μοντελοποιημένες με την γλώσσα BPMN 2.0. Υποστηρίζει λειτουργίες όπως ο έλεγχος και η δυναμική τροποποίηση των ροών εργασιών (workflows), τόσο σε μεμονωμένα στιγμιότυπα διαδικασιών όσο και στο σύνολο των ενεργών και μελλοντικών στιγμιότυπων. Επίσης παράγει και αποστέλλει γεγονότα για την εξέλιξη της εκτέλεσης των διαδικασιών (δείτε στις προδιαγραφές). Αν στη θέση του APES χρησιμοποιηθεί κάποιος workflow engine που δεν υποστηρίζει εγγενώς την προσέγγιση FlexiBPMN2.0 τότε απαιτείται μία επέκταση αυτού, η οποία θα πραγματοποιεί την επικοινωνία με τα υπόλοιπα μέρη της αρχιτεκτονικής. Η επέκταση αυτή καλείται με τον γενικό όρο *Μονάδα Θεματοστρεφούς Επέκτασης* ή *Aspect-orientation Extension Module (AEM)*.

Διαχειριστής Προσαρμογών (Adaptation Manager). Αυτό το στοιχείο της αρχιτεκτονικής λειτουργεί ως επέκταση του APES. Ο κύριος ρόλος του είναι να παρακολουθεί στενά την εκτέλεση των διαφόρων στιγμιότυπων των διαδικασιών, να εντοπίζει τα σημεία των διαδικασιών όπου ακολουθεί εκτελέσιμος κόμβος (joint points), να ελέγχει αν υπάρχουν σημεία τομών (pointcuts) σε αυτά και να ενεργοποιεί τις αντίστοιχες συμβουλές (advices). Όταν απαιτείται, ο διαχειριστής προσαρμογών επικοινωνεί με τον SAN engine προκειμένου να ολοκληρώσει τα αναγκαία βήματα για την πραγματοποίηση των προσαρμογών (για παράδειγμα στον σύγχρονο τρόπο λειτουργίας του APES). Η αλληλεπίδραση με τον SAN engine και το APES μπορεί να επιτευχθεί:

- ανταλλάσσοντας γεγονότα διάμεσο του διαύλου γεγονότων (event bus), ή
- υλοποιώντας μια απευθείας «ιδιωτική» σύνδεση μεταξύ του APES και του SAN engine. Αυτή η εναλλακτική παρέχει ίσως καλύτερες επιδόσεις αφού παρακάμπτει τη χρήση του διαύλου γεγονότων και την επεξεργασία γεγονότων που μπορούν να επιφέρουν καθυστερήσεις στην επικοινωνία μεταξύ των δύο συστημάτων. Ωστόσο δεν παρέχει την ευελιξία και την ευκολία διαχείρισης και συντήρησης που παρέχει ο δίαυλος γεγονότων.

Αποθετήριο Θεμάτων (Aspects Repository). Αποθηκεύει τις περιγραφές των θεμάτων και των στοιχείων που τα απαρτίζουν. Μπορεί να είναι είτε μια συλλογή αρχείων (που περιέχουν τις τυπικές περιγραφές των θεμάτων), ή μια βάση δεδομένων, ή οποιοδήποτε προσβάσιμο σύστημα μόνιμης αποθήκευσης δεδομένων (persistent storage system).

8.4.3 Υλοποίηση του APES

Η υλοποίηση του APES στηρίχθηκε στο WorkToken⁶⁴, έναν ελαφρύ (light-weight), ανοιχτού κώδικα, βασισμένο σε Java, μηχανισμό εκτέλεσης ροών εργασιών για τη BPMN 2.0. Το WorkToken είναι ταυτόχρονα και ένα πλαίσιο εργασίας (framework) για την ανάπτυξη προσαρμοσμένων (custom) μηχανισμών εκτέλεσης BPMN 2.0 για εξειδικευμένες χρήσεις. Το πλαίσιο εργασίας είναι υπεύθυνο για τη δημιουργία των απαραίτητων αντικειμένων (εργασιών, κόμβων γεγονότων, πυλών αποφάσεων), την αποθήκευσή τους όταν χρειάζεται και τη διαχείριση των ροών εργασιών. Δεν παρέχει ωστόσο συγκεκριμένες (concrete) υλοποιήσεις των διαφόρων τύπων εργασιών, κόμβων γεγονότων, πυλών αποφάσεων που προβλέπει η BPMN 2.0 (εξάλλου και η ίδια η γλώσσα δεν προσδιορίζει συγκεκριμένους τρόπους υλοποίησης), ούτε άλλα υποστηρικτικά εργαλεία (όπως property editors, data mappers, scripting language support) που υπάρχουν σε πιο προηγμένους μηχανισμούς εκτέλεσης ροών εργασιών. Αντίθετα θα πρέπει ο προγραμματιστής να παρέχει τις υλοποιήσεις και τα εργαλεία αυτά, σύμφωνα με τις απαιτήσεις της εκάστοτε εφαρμογής. Αυτό είναι λογικό αφού σκοπός του WorkToken είναι η ανάπτυξη εξειδικευμένων BPMN2 workflow engines.

Στο APES παρέχουμε κάποιες εξορισμού (default) υλοποιήσεις για τους πιο συνηθισμένους τύπους εργασιών, κόμβων γεγονότων και πυλών αποφάσεων, όπως την εμφάνιση κατάλληλων πλαισίων διαλόγων (dialog boxes) για τα User Tasks, κλήσεις σε εξωτερικά web services για τα Service Tasks, εκτέλεση εντολών σε JavaScript για τα Script Tasks, ανταλλαγή γεγονότων διαμέσου του διαύλου γεγονότων για τους κόμβους γεγονότων (event nodes) και εμφάνιση κατάλληλων πλαισίων διαλόγου για τους πιο συνηθισμένους τύπους πυλών αποφάσεων. Αν απαιτηθεί, το APES μπορεί να επεκταθεί με την προσθήκη υλοποιήσεων και για χαρακτηριστικά της BPMN 2.0 που δεν υποστηρίζονται επί του παρόντος, ή για εναλλακτικές υλοποιήσεις των υφισταμένων. Οι επεκτάσεις αυτές θα πρέπει να υλοποιηθούν ως κλάσεις Java και να δηλωθούν στο αρχείο διαμόρφωσης του APES.

⁶⁴ <http://www.worktoken.com/>

Η υλοποίηση του APES συμπεριλαμβάνει *Μονάδα Θεματοστρεφούς Επέκτασης Θεμάτων (AEM)*, η οποία επεκτείνει τις δυνατότητες του WorkToken έτσι ώστε να υποστηρίζει θεματοστρεφείς προσαρμογές (aspects). Επιπλέον η AEM δίνει τη δυνατότητα στο APES να αποστέλλει γεγονότα που αφορούν την εξέλιξη της εκτέλεσης των διαφόρων στιγμιότυπων των διαδικασιών. Τα γεγονότα αυτά αφορούν την έναρξη/τερματισμό ενός στιγμιότυπου, και την έναρξη/τερματισμό/διακοπή μιας εργασίας (task). Το APES διαθέτει ακόμη περιβάλλον αλληλεπίδρασης με το χρήστη από τη γραμμή εντολών (command-line interface).

Τα θέματα καταγράφονται ως αρχεία XML (με κατάληξη .aspect) χρησιμοποιώντας ένα συγκεκριμένο συντακτικό βασισμένο στην XML. Κάθε αρχείο θέματος περιέχει επίσης και τους ορισμούς των αντίστοιχων σημείων τομών (pointcuts) και των αντίστοιχων συμβουλών (advices). Οι συμβουλές είναι υπο-διαδικασίες σε BPMN 2.0 οι οποίες ενεργοποιούνται στα κατάλληλα σημεία τομών. Για το λόγο αυτό στους ορισμούς των συμβουλών υπάρχουν αναφορές (με τη μορφή URLs) προς τα αντίστοιχα BPMN 2.0 αρχεία με τα μοντέλα των διαδικασιών των συμβουλών. Έτσι προκειμένου για να εφαρμοστεί ένα θέμα, το APES θα πρέπει να ανακτήσει και να φορτώσει το απαιτούμενο αρχείο BPMN 2.0 που περιέχει την υπο-διαδικασία της συμβουλής.

Αν απαιτηθεί η εκτέλεση και προσαρμογή διαδικασιών μοντελοποιημένων σε γλώσσα άλλη από την BPMN 2.0 (για παράδειγμα με BPEL), τότε προφανώς το υπο-σύστημα αυτό δεν μπορεί να χρησιμοποιηθεί. Ωστόσο είναι εφικτή η τροποποίηση αντίστοιχων διαθέσιμων μηχανισμών εκτέλεσης διαδικασιών (όπως είναι οι Apache ODE, jBPM, Activiti), έτσι ώστε να μπορούν να δέχονται τις προσαρμογές που προτείνει ο PAR. Φυσικά τότε θα πρέπει να ρυθμιστεί κατάλληλα ο PAR έτσι ώστε να παράγει περιγραφές προσαρμογών που θα είναι κατανοητές στον εκάστοτε μηχανισμό εκτέλεσης διαδικασιών.

8.5 Παράδειγμα χρήσης της πλατφόρμας ARP

Ακολουθως δίνεται ένα αναλυτικό παράδειγμα χρήσης της πλατφόρμας προτάσεων προσαρμογών για την προσαρμογή μιας διαδικασίας διαχείρισης κρίσεων, σχετική με ατυχήματα σε πυρηνικό σταθμό παραγωγής ηλεκτρικής ενέργειας. Το παράδειγμα αυτό θα βοηθήσει στην καλύτερη και πληρέστερη κατανόηση των δυνατοτήτων και του τρόπου λειτουργίας της πλατφόρμας.

8.5.1 Σενάριο του παραδείγματος

Μια μεγάλη ποσότητα ραδιενεργού υλικού διαρρέει στην ατμόσφαιρα λόγω ατυχήματος στο πυρηνικό εργοστάσιο. Προκειμένου να αντιμετωπιστεί η κρίση πρέπει να εμπλακούν αρκετοί διαφορετικοί φορείς που παρέχουν τις απαραίτητες υπηρεσίες. Τυπικά τέτοιες διαδικασίες είναι πολύ σύνθετες όμως χάριν ευκολίας έχουμε απλοποιήσει το παράδειγμα.

Στην περίπτωση ενός πυρηνικού ατυχήματος πρέπει να γίνουν μια σειρά από αποτρεπτικές ενέργειες (mitigating actions) που περιγράφονται ως διαδικασίες που εμπλέκουν φορείς όπως είναι η Αστυνομία, ο Στρατός, η Πυροσβεστική, το Ινστιτούτο Ραδιοπροστασίας και η Εθνική Αρχή Ατομικής Ενέργειας. Η Αρχή Διαχείρισης της κρίσης, οι οποία συντονίζει όλους

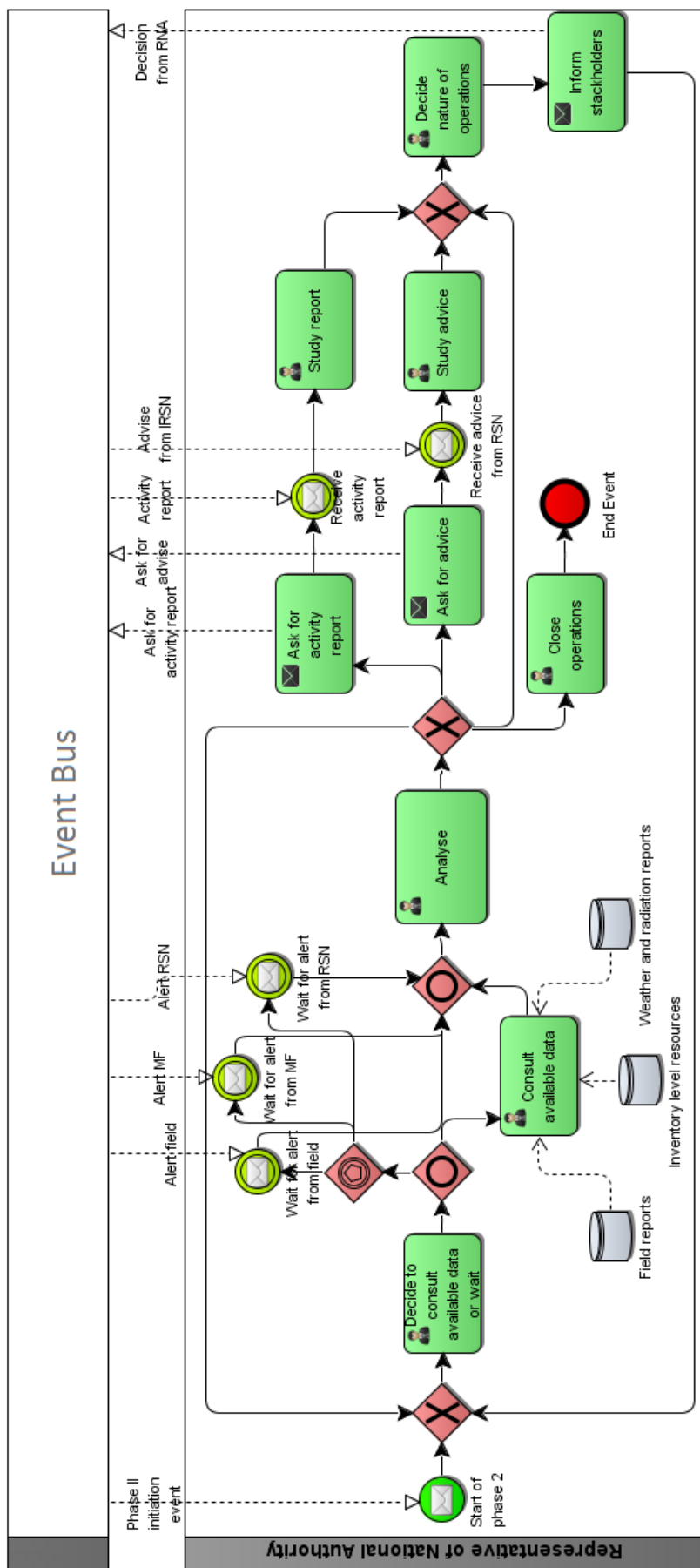
τους άλλους φορείς, θα πρέπει να έχει πρόσβαση στα τρέχοντα μετεωρολογικά δεδομένα, στις αναφορές των μονάδων που βρίσκονται στον τόπο του ατυχήματος, σε μετρήσεις που προέρχονται από το δίκτυο των μετρητών ραδιενέργειας (εγκατεστημένο στην ευρύτερη περιοχή) καθώς και σε ιστορικά δεδομένα. Με βάση τα στοιχεία αυτά μπορεί να λαμβάνει τις κατάλληλες αποφάσεις προκειμένου να ελαχιστοποιήσει τις συνέπειες του ατυχήματος. Ο βασικός κορμός της διαδικασίας διαχείρισης κρίσεων παρουσιάζεται στο Σχήμα 8-3, όμως αυτή πλαισιώνεται από εκατοντάδες άλλες διαδικασίες που πρέπει να εκτελεστούν κατά την αντιμετώπιση ενός πυρηνικού ατυχήματος.

Η διαδικασία του παραδείγματος αφορά την «Αρχή Ατομικής Ενέργειας», η οποία έχει την ευθύνη για τον συντονισμό και τη διαχείριση της κρίσεως. Στο πλαίσιο της διαδικασίας η Αρχή καλείται να εκτιμήσει την τρέχουσα κατάσταση και να αποφασίσει για τη φύση των ενεργειών και των μέτρων που θα πρέπει να ληφθούν (όπως εκκένωση της περιοχής πέριξ του εργοστασίου, διανομή σκευασμάτων ιωδίνης στον πληθυσμό και άλλα). Οι διάφορες απαιτούμενες πληροφορίες είναι διαθέσιμες ως γεγονότα διάμεσο ενός κοινού διαύλου ανταλλαγής γεγονότων.

Στο πλαίσιο της συγκεκριμένης διαδικασίας, ο υπεύθυνος της Αρχής μπορεί να μελετά όλα τα διαθέσιμα ιστορικά δεδομένα (προηγούμενες αναφορές επιτόπιων μονάδων, αρχείο μετεωρολογικών και ραδιολογικών μετρήσεων, διαθεσιμότητα και κατάσταση μονάδων και υλικών). Επίσης μπορεί να λαμβάνει σε πραγματικό χρόνο ειδοποιήσεις (alerts) για νεώτερες αναφορές και μετρήσεις, προκειμένου να ενημερώνεται άμεσα για την εξέλιξη της κατάστασης. Αφού συνεκτιμηθούν όλα τα στοιχεία ο υπεύθυνος έχει τις εξής επιλογές:

- να επαναλάβει την ανάλυση των στοιχείων, αν εν τω μεταξύ έχουν ληφθεί νεώτερες ειδοποιήσεις,
- να ζητήσει επιπλέον αναφορές από τις επιτόπιες μονάδες,
- να ζητήσει γνωμοδότηση από το Ινστιτούτο Ραδιοπροστασίας,
- να αποφασίσει για τις ενέργειες που θα πρέπει να γίνουν, και
- να τερματίσει τη διαδικασία όταν σταθεροποιηθεί η κατάσταση.

Με εξαίρεση την τελευταία περίπτωση, η διαδικασία επαναλαμβάνεται διαρκώς, δηλαδή πρώτα γίνεται μελέτη των ιστορικών και των πρόσφατων στοιχείων, κι έπειτα επιλέγεται μία από τις παραπάνω 5 ενέργειες. Ωστόσο είναι πιθανό να έρθουν νέα στοιχεία ή να συμβεί κάτι επείγον ενώ αναμένεται η ολοκλήρωση κάποιας εργασίας που απαιτεί αρκετό χρόνο (όπως είναι η αναμονή για τη γνωμοδότηση του Ινστιτούτου Ραδιοπροστασίας ή μία αναφορά από κάποια επιτόπια μονάδα ή η μελέτη των ιστορικών στοιχείων). Τότε θα πρέπει να παρεμβληθούν στην κανονική διαδικασία ορισμένα επιπλέον βήματα κατά την εκτέλεση των οποίων θα συνεκτιμηθούν τα νέα δεδομένα. Αυτό μπορεί να συμβεί οποτεδήποτε και οπουδήποτε μέσα στη διαδικασία. Είναι προφανές ότι είναι δύσκολο να μοντελοποιηθούν όλες οι πιθανές περιπτώσεις κατά τον σχεδιασμό. Αντιθέτως είναι πολύ πιο εύκολο οι όποιες «έκτακτες» περιπτώσεις να μοντελοποιηθούν ως ανεξάρτητες τροποποιήσεις (προσαρμογές), γεγονός που θα οδηγήσει σε πιο ξεκάθαρα μοντέλα κύριας διαδικασίας και προσαρμογών. Ακόμη, επιπλέον προσαρμογές μπορεί να προστίθενται και κατά τον χρόνο εκτέλεσης.



Σχήμα 8-3. Το μοντέλο της διαδικασίας διαχείρισης κρίσεων σε BPRM 2.0

Η χρήση της ARP μπορεί να βοηθήσει προς αυτή την κατεύθυνση. Κατάλληλα SANs μπορούν να ανιχνεύουν τότε υπάρχουν διαθέσιμες νέες πληροφορίες ενώ η κύρια διαδικασία καθυστερεί σε κάποια χρονοβόρα εργασία. Αναλόγως με το σημείο της διαδικασίας και την κρισιμότητα της κατάστασης, τα SANs μπορεί να προτείνουν είτε την εκτίμηση των νέων στοιχείων αφού ολοκληρωθεί η τρέχουσα εργασία (με προσθήκη επιπλέον βημάτων «μετά»), είτε να τα αγνοήσουν (θα συνεκτιμηθούν σε επόμενη επανάληψη της διαδικασίας), είτε να διακόψουν την τρέχουσα εργασία και να εισαγάγουν νέα βήματα για την άμεση μελέτη των νέων στοιχείων.

8.5.2 Παραδείγματα προσαρμογών

Με βάση το προηγούμενο σενάριο παρουσιάζουμε μερικές πιθανές προσαρμογές (Σχήμα 8-4) στην κύρια διαδικασία (Σχήμα 8-3).

- *Προσαρμογή 1:* εισαγωγή της εργασίας “Analyse Alerts issued in the last 2 minutes” πριν από βήματα (tasks) της διαδικασίας που έπονται μιας εργασίας που διαρκεί πάνω από 2 λεπτά, αν στο μεταξύ διάστημα παραληφθούν νέες ειδοποιήσεις που δεν έχει μελετήσει ο υπεύθυνος.

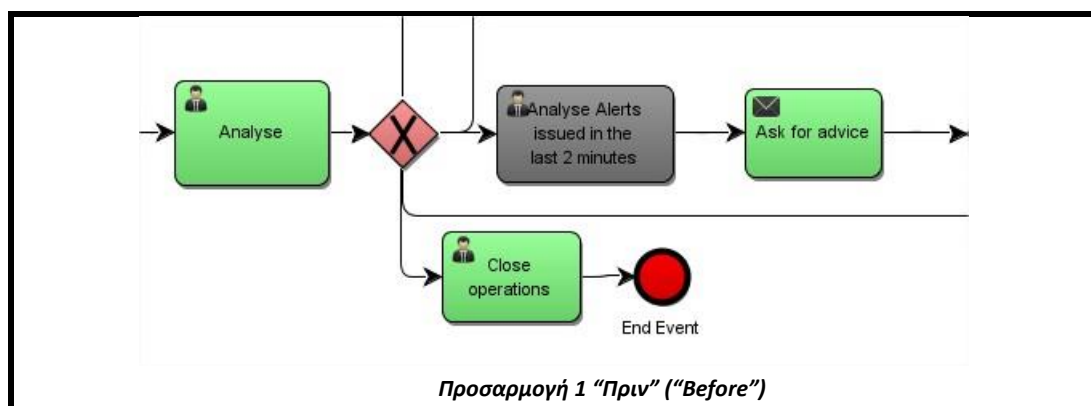
*Προσαρμογή τύπου “Πριν”. Σχετικό θέμα το **Aspect 1**.*

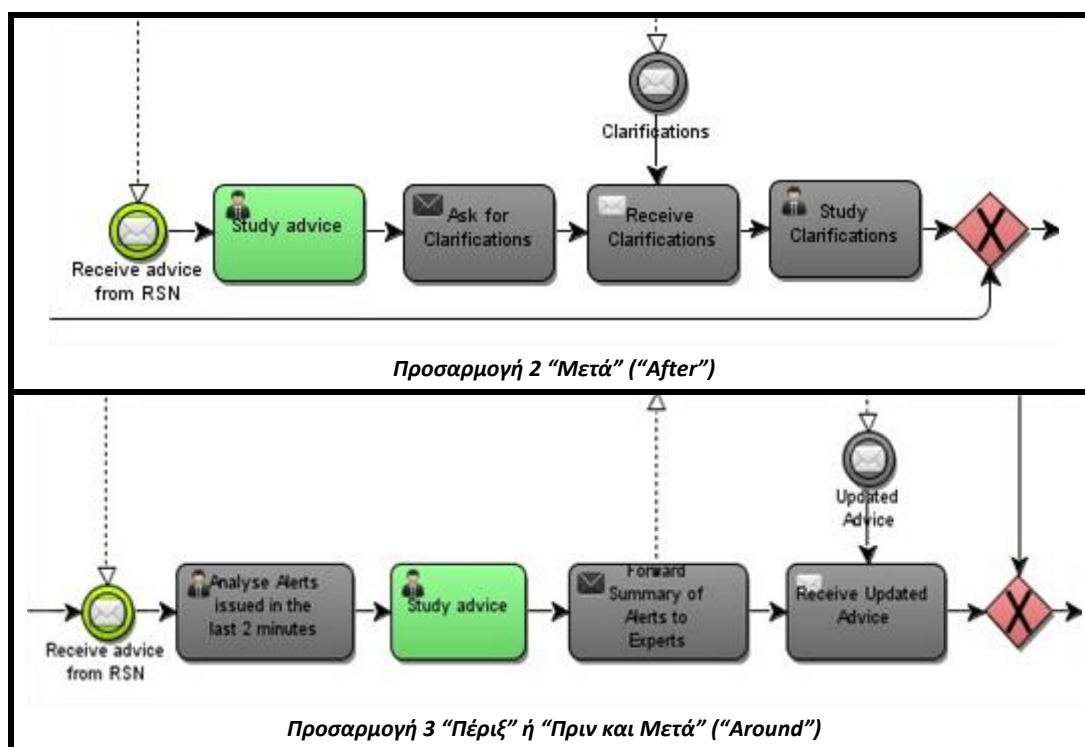
- *Προσαρμογή 2:* εισαγωγή μετά την εργασία “Study advice” της σειριακής υπο-διαδικασίας “Ask for clarifications”(task), “Receive Clarifications” και “Study clarifications”. Η προσαρμογή αυτή προτείνεται όταν η εργασία “Study advice” διαρκέσει πάνω από 30 λεπτά, οπότε ζητείται η διακοπή της και η εκτέλεση μιας προσαρμοσμένης υποδιαδικασίας (συμβουλή 2).

*Προσαρμογή τύπου “Μετά”. Σχετικό θέμα το **Aspect 2**.*

- *Προσαρμογή 3:* εισαγωγή νέων βημάτων πριν και μετά την εργασία “Study advice”. Στην περίπτωση που πριν αρχίσει η εκτέλεση της εργασίας, και ενώ αναμένεται γνωμοδότηση από το Ινστιτούτο Ραδιοπροστασίας, ληφθούν ειδοποιήσεις και στοιχεία που δεν έχει μελετήσει ο υπεύθυνος τότε πριν την “Study advice” μπαίνει το βήμα “Analyse Alerts issued in the last 2 minutes” ενώ μετά τα βήματα “forward summary of alerts to experts” and “Study updated advice”.

*Προσαρμογή τύπου “Πέριξ” ή “Πριν και Μετά”. Σχετικό θέμα το **Aspect 3**.*

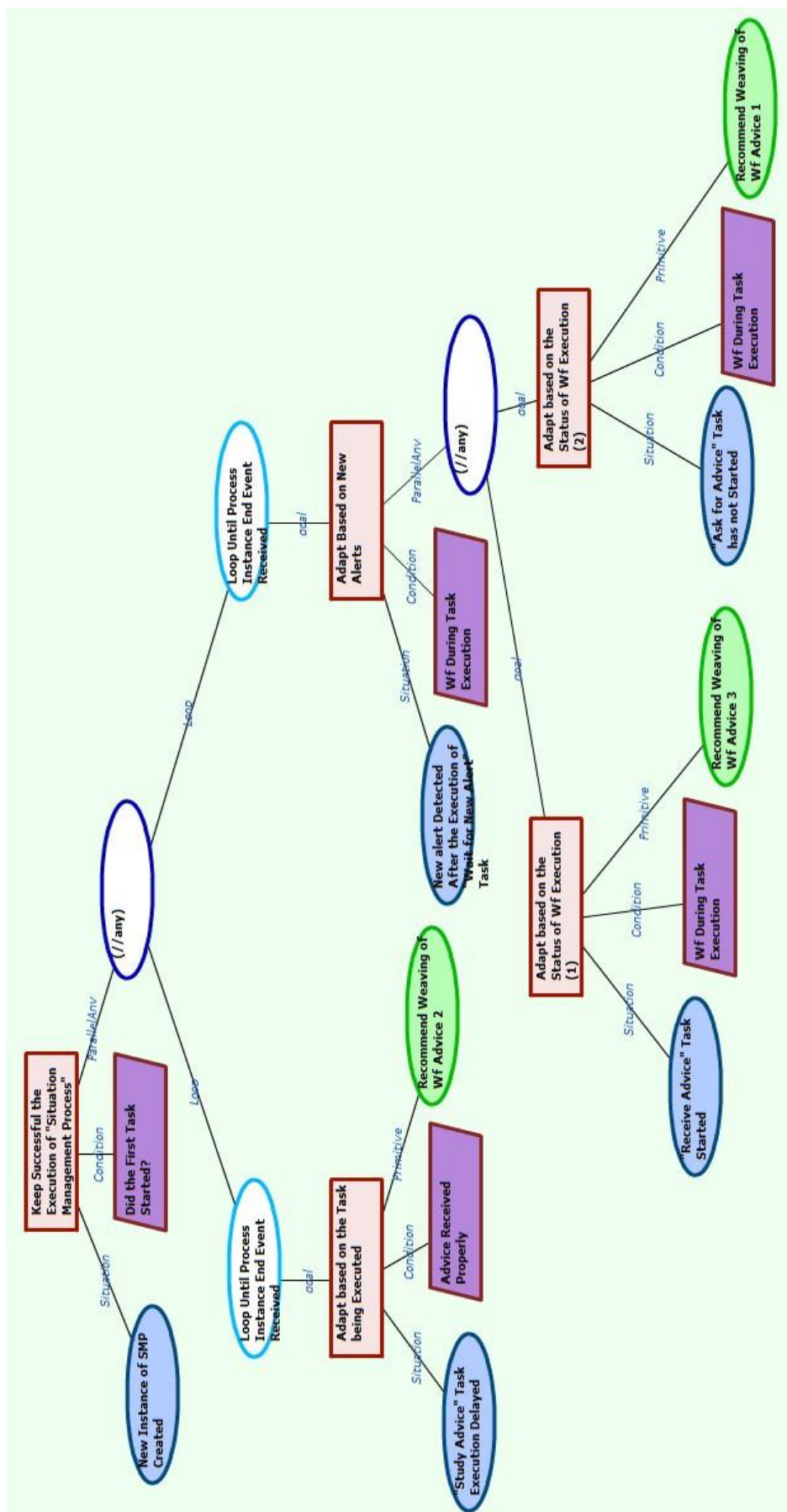




Σχήμα 8-4. Προσαρμογές “Πριν”, “Μετά” και “Πέριξ” για τη διαδικασία διαχείρισης κρίσεων

Για την πρόταση των παραπάνω προσαρμογών κατά την εκτέλεση της διαδικασίας, έχει υλοποιηθεί το SAN που απεικονίζεται στο Σχήμα 8-5, το οποίο ουσιαστικά είναι ένα σύστημα παραγωγής προτάσεων εξειδικευμένο στη διαδικασία διαχείρισης κρίσεων. Με την εκτέλεση του εν λόγω SAN, ο SAN engine μπορεί πλέον να παρακολουθεί την εξέλιξη της διαδικασίας διαχείρισης κρίσεων (που εκτελείται στο APES), να ανιχνεύει τις καταστάσεις όπου χρειάζεται να γίνουν προσαρμογές, να επιλέγει το εκάστοτε κατάλληλο θέμα (που περιγράφει την προσαρμογή που χρειάζεται) και να την προτείνει στο APES.

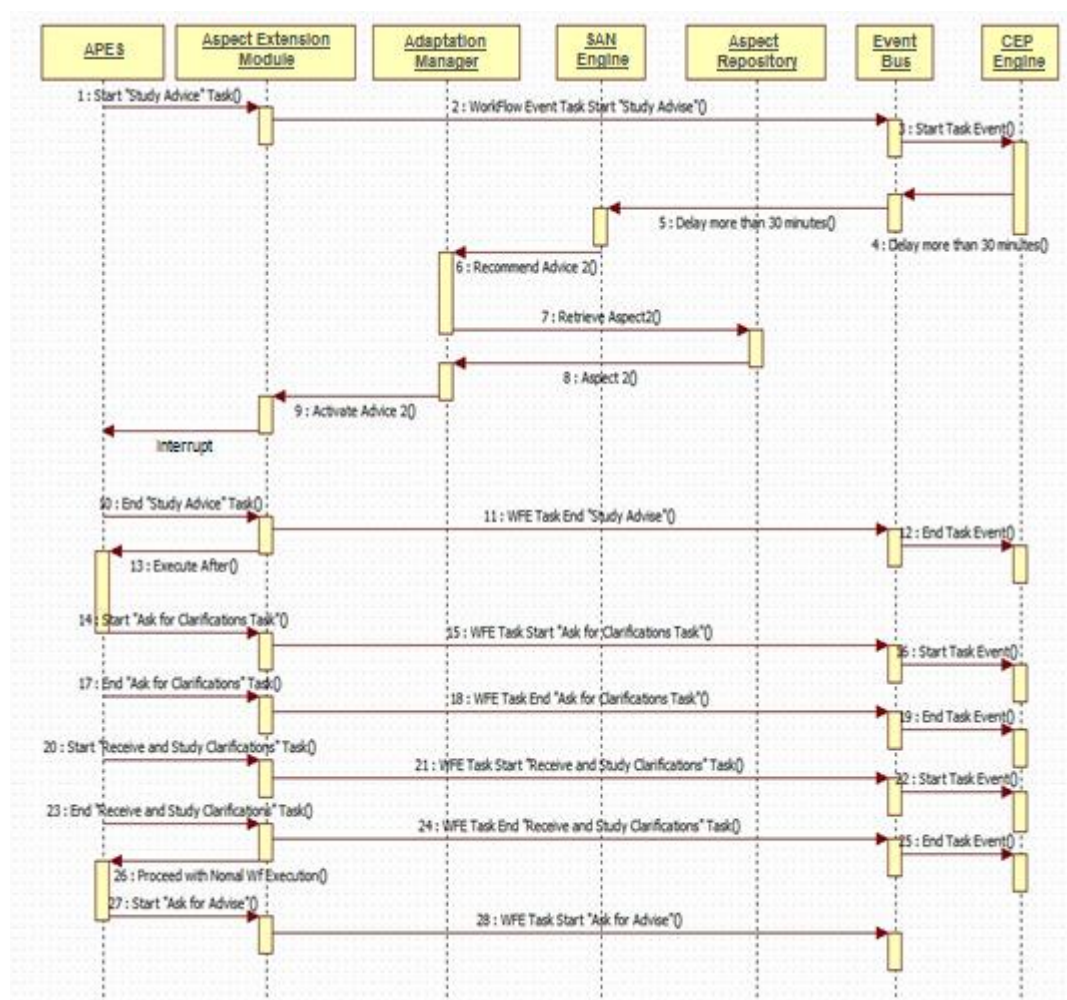
Για κάθε στιγμιότυπο της διαδικασίας δημιουργείται και ένα αντίστοιχο στιγμιότυπο του SAN, προκειμένου να παρακολουθεί τη ροή εκτέλεσής του. Ο αριστερός κλάδος του SAN (βλέπε Σχήμα 8-5) εξετάζει αν η εργασία “Study Advice” δεν ολοκληρωθεί μέσα σε 30 λεπτά (execution delay event από τον CEP engine). Τότε προτείνει την προσαρμογή 2 (δηλαδή διακοπή της “Study Advice” και αίτηση για επιπλέον διευκρινίσεις από το Ινστιτούτο Ραδιοπροστασίας). Ο δεξιός κλάδος του SAN ελέγχει αν έχουν ληφθεί νέες ειδοποιήσεις μετά την ολοκλήρωση της εργασίας “Wait for new alerts” (άρα δεν έχουν μελετηθεί ακόμη από τον υπεύθυνο). Σε αυτή την περίπτωση εξετάζει αν η διαδικασία έχει ήδη στείλει αίτημα για γνωμοδότηση στο Ινστιτούτο Ραδιοπροστασίας και αναμένει την απάντηση ή αν δεν έχει αποσταλεί τέτοιο αίτημα. Στη μεν πρώτη περίπτωση προτείνει την προσαρμογή 3 (μετά την παραλαβή της γνωμοδότησης, μελετώνται οι νέες ειδοποιήσεις, έπειτα εκτελείται η “Study Advice” και τέλος ζητείται και λαμβάνεται συμπληρωματική γνωμοδότηση). Στην δε δεύτερη περίπτωση προτείνεται η προσαρμογή 1 (εισαγωγή του “Analyze alerts issued in last 2 minutes” πριν την αποστολή αιτήματος για γνωμοδότηση).



Σχήμα 8-5. SAN για τη δημιουργία προτάσεων προσαρμογών για τη διαδικασία διαχείρισης κρίσεων

Στο Σχήμα 8-6 παρουσιάζεται ένα διάγραμμα αλληλεπιδράσεων (sequence diagram) το οποίο δίνει μια εποπτική εικόνα της προσαρμογής της διαδικασίας διαχείρισης κρίσεων με χρήση του θέματος Aspect 2. Περιλαμβάνει λεπτομέρειες που αφορούν την αλληλεπίδραση μεταξύ των μερών της πλατφόρμας προτάσεως προσαρμογών, και αποτυπώνει τα βήματα της προσαρμογής από την ανίχνευση της ανάγκης μέχρι και την εφαρμογή της αντίστοιχης συμβουλής (advice) και την εκτέλεση των (νέων) εργασιών που αυτή περιλαμβάνει. Ειδικότερα, οι αλληλεπιδράσεις που περιγράφονται αφορούν :

- το APES (WorkToken implentation)
- τη Μονάδα Θεματοστρεφούς Επέκτασης (AEM) (μέρος της υλοποίησης του APES)
- τον διαχειριστή προσαρμογών (Adapt. Manager) (μέρος της υλοποίησης του APES)
- το μηχανισμό SANs (SAN Engine)
- το αποθετήριο προσαρμογών (Aspects Repository)
- τον δίαυλο γεγονότων (Event bus – εξωτερικό σύστημα)
- τον επεξεργαστή σύνθετων γεγονότων (CEP engine – εξωτερικό σύστημα)

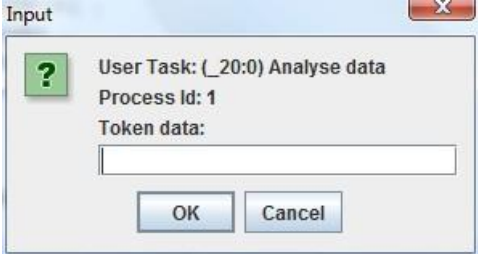


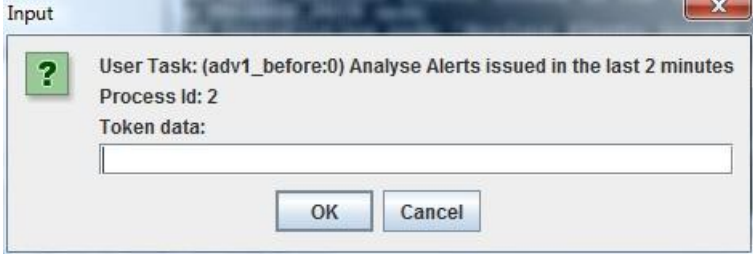
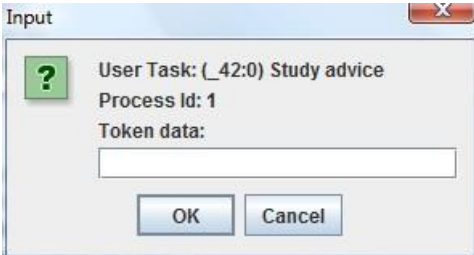
Σχήμα 8-6. Διάγραμμα αλληλεπίδρασης κατά την εφαρμογή της προσαρμογής Aspect 2

Στους ακόλουθους πίνακες δίνονται εικόνες (snapshots) των κυριότερων βημάτων της εφαρμογής (weave) των τριών προσαρμογών που παρουσιάσαμε, συμπεριλαμβανομένης της ανίχνευσης της ανάγκης προσαρμογής από τον SAN engine και την εκτέλεση των νέων

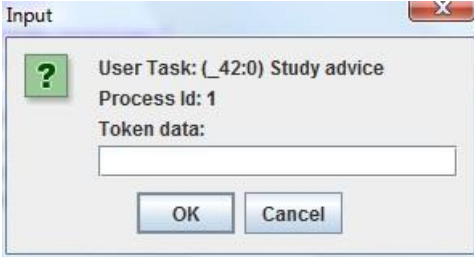
εργασιών που εισάγουν οι προσαρμογές. Συγκεκριμένα Πίνακας 8-3 παρουσιάζει την προσαρμογή 1, ενώ ο Πίνακας 8-4 και ο Πίνακας 8-5 δίνουν αντίστοιχα τις προσαρμογές 2 και 3.

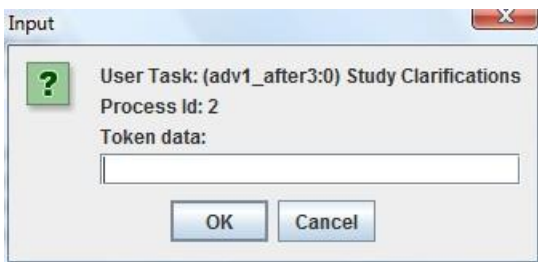
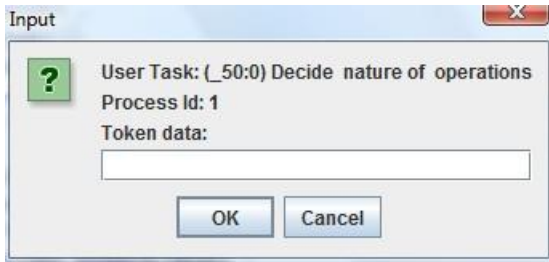
Πίνακας 8-3. Εικόνες από την εφαρμογή της Aspect 1 (“Before”)

1	<p>Η διαδικασία καθυστερεί στην εργασία “Analyse data” καθώς ο υπεύθυνος της Αρχής μελετά τα διαθέσιμα στοιχεία. (Εκτέλεση της κύριας διαδικασίας)</p> 
2	<p>Στο μεταξύ λαμβάνεται μια νέα ειδοποίηση και ο μηχανισμός SANs ανταποκρίνεται παράγοντας μια πρόταση προσαρμογής της διαδικασίας με χρήση του θέματος Aspect 1. Ακολούθως φαίνεται τα γεγονότα που περιέχει την πρόταση, όπως αυτό καταγράφηκε στον δίαυλο γεγονότων. Το περιεχόμενο του γεγονότος έχει εκφραστεί με χρήση της γλώσσας RDF.</p> <pre data-bbox="459 996 1212 1691"> ADAPTATION_RECOMMENDATION_EVENT <http://events.event-processing.org/ids/arp-1348316657072000000#event> a :ADAPTATION_RECOMMENDATION_EVENT ; :endTime "2012-09-22T12:24:17Z"^^xsd:dateTime ; :stream <http://streams.event-processing.org/ids/AsyncAdaptationRecommendations#stream> ; <http://imu.ntua.gr/play/aop4bpmn2/adaptation-events/1#ask-user> "no" ; <http://imu.ntua.gr/play/aop4bpmn2/adaptation-events/1#engine-id> "aop4bpmn2-engine-1348316550195" ; <http://imu.ntua.gr/play/aop4bpmn2/adaptation-events/1#event-type> "IS_RECOMMENDATION" ; <http://imu.ntua.gr/play/aop4bpmn2/adaptation-events/1#id> "arp-1348316657072000000" ; <http://imu.ntua.gr/play/aop4bpmn2/adaptation-events/1#message> "" ; <http://imu.ntua.gr/play/aop4bpmn2/adaptation-events/1#session-id> "aop4bpmn2-session-6c7131f6_af74_4e73_9183_77d5839e7aa0" ; <http://imu.ntua.gr/play/aop4bpmn2/adaptation-recommendation-events/1#action> "activate-aspect" ; <http://imu.ntua.gr/play/aop4bpmn2/adaptation-recommendation-events/1#aspects> "T:/examples/aop4bpmn2/nuclear-SMP-adaptation-ADVISE-1.aspect.xml" </pre>
3	<p>Το APES παραλαμβάνει την πρόταση και αμέσως φορτώνει και ενεργοποιεί το σχετικό θέμα. (Μηνύματα του APES στην κονσόλα). Σημειώνουμε ότι το APES έχει ρυθμιστεί ώστε να δέχεται αυτόματα τις προτάσεις προσαρμογών.</p> <pre data-bbox="327 1863 1361 1966"> Loading aspects from T:/examples/aop4bpmn2/nuclear-SMP-adaptation-ADVISE-1.aspect.xml...ok ** Aspect added and activated: T:/examples/aop4bpmn2/nuclear-SMP-adaptation-ADVISE-1.aspect.xml#nuclear_SMP_adaptation_ASPECT_1 on session aop4bpmn2-session-6c7131f6_af74_4e73_9183_77d5839e7aa0 </pre>

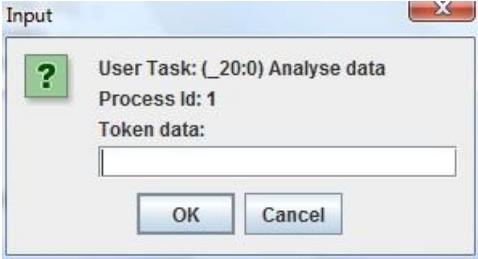
<p>4</p>	<p>Ο υπεύθυνος ολοκληρώνει τη μελέτη των στοιχείων (“Analyse data”) κι επιλέγει να ζητήσει γνωμοδότηση από το Ινστιτούτο Ραδιοπροστασίας. Τότε το APES ανιχνεύει ότι πρέπει να εφαρμοστεί η προσαρμογή Aspect 1. Έτσι εισάγεται και εκτελείται η νέα εργασία “Analyse Alerts issued in last 2 minutes” πριν (before) απ’ την κανονική εργασία που είναι η “Ask for Advice”. (Εκτέλεση της προσαρμογής)</p> 
<p>5</p>	<p>Όταν ο υπεύθυνος ολοκληρώσει την ανάλυση των νέων στοιχείων η διαδικασία προχωρά κανονικά στην αποστολή του αιτήματος γνωμοδότησης στο Ινστιτούτο Ραδιοπροστασίας. (Εκτέλεση της κύριας διαδικασίας)</p> <pre data-bbox="464 835 1107 1093"> NUCLEAR_PHASE_II_RNA_PROCESS_ASK_FOR_ADVICE <http://events.event-processing.org/ids/0ddaa073-68cb-446a-a2cd-8763adc10ee3#event> a :NUCLEAR_PHASE_II_RNA_PROCESS_ASK_FOR_ADVICE ; :endTime "2012-09-22T12:26:24Z"^^xsd:dateTime ; <http://imu.ntua.gr/play/aop4bpnm2/nuclear-example/1#id> "0ddaa073-68cb-446a-a2cd-8763adc10ee3" ; <http://imu.ntua.gr/play/aop4bpnm2/nuclear-example/1#message> "" ; </pre>
<p>6</p>	<p>Λήψη της γνωμοδότησης. Η διαδικασία από το σημείο αυτό και μετά συνεχίζει κανονικά. (Εκτέλεση της κύριας διαδικασίας)</p> <pre data-bbox="464 1245 1107 1503"> NUCLEAR_PHASE_II_RNA_PROCESS_ADVICE_RESPONSE <http://events.event-processing.org/ids/2c8a4002-69ac-4753-8282-0b8fe948a9d4#event> a :NUCLEAR_PHASE_II_RNA_PROCESS_ADVICE_RESPONSE ; :endTime "2012-09-22T12:26:27Z"^^xsd:dateTime ; <http://imu.ntua.gr/play/aop4bpnm2/nuclear-example/1#id> "2c8a4002-69ac-4753-8282-0b8fe948a9d4" ; <http://imu.ntua.gr/play/aop4bpnm2/nuclear-example/1#message> "" ; </pre>
<p>7</p>	<p>Μελέτη της γνωμοδότησης. (Εκτέλεση της κύριας διαδικασίας)</p> 

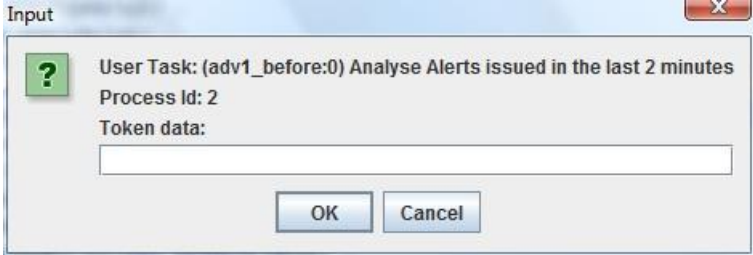
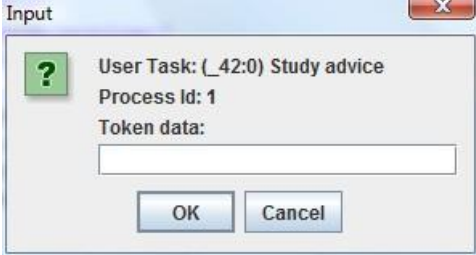
Πίνακας 8-4. Εικόνες από την εφαρμογή της Aspect 2 (“After”)

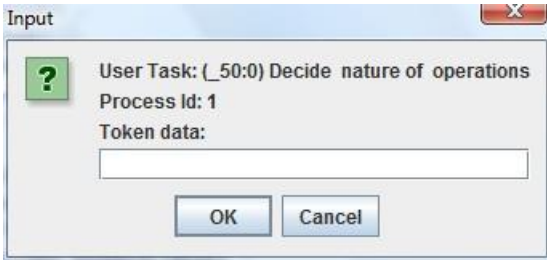
1	<p>Η διαδικασία βρίσκεται στην εργασία “Study advice” καθώς ο υπεύθυνος της Αρχής μελετά τη γνωμοδότηση του Ινστιτούτου Ραδιοπροστασίας. (Εκτέλεση της κύριας διαδικασίας)</p> 
2	<p>Επειδή όμως η μελέτη της γνωμοδότησης παίρνει αρκετό χρόνο ο μηχανισμός SANS προτείνει την εφαρμογή της προσαρμογής Aspect 2 προκειμένου να ζητηθούν επιπλέον διευκρινίσεις. Λόγω της κρισιμότητας της κατάστασης προτείνεται η άμεση διακοπή της “Study Advice”. Ακολούθως φαίνεται το γεγονός που περιέχει την πρόταση, όπως αυτό καταγράφηκε στον διάλογο γεγονότων.</p> <pre data-bbox="466 884 1214 1570"> ADAPTATION_RECOMMENDATION_EVENT <http://events.event-processing.org/ids/arp-1348313325688999936#event> a :ADAPTATION_RECOMMENDATION_EVENT ; :endTime "2012-09-22T11:28:45Z"^^xsd:dateTime ; :stream <http://streams.event-processing.org/ids/AsyncAdaptationRecommendations#stream> ; <http://imu.ntua.gr/play/aop4bpmn2/adaptation-events/1#ask-user> "no" ; <http://imu.ntua.gr/play/aop4bpmn2/adaptation-events/1#engine-id> "aop4bpmn2-engine-1348313103489" ; <http://imu.ntua.gr/play/aop4bpmn2/adaptation-events/1#event-type> "IS_RECOMMENDATION" ; <http://imu.ntua.gr/play/aop4bpmn2/adaptation-events/1#id> "arp-1348313325688999936" ; <http://imu.ntua.gr/play/aop4bpmn2/adaptation-events/1#message> "" ; <http://imu.ntua.gr/play/aop4bpmn2/adaptation-events/1#session-id> "aop4bpmn2-session-bde2df86_2fd5_4d87_92d7_7adc79b71ee2" ; <http://imu.ntua.gr/play/aop4bpmn2/adaptation-recommendation-events/1#action> "activate-aspect" ; <http://imu.ntua.gr/play/aop4bpmn2/adaptation-recommendation-events/1#aspects> "T:/examples/aop4bpmn2/nuclear-SMP-adaptation-ADVISE-2.aspect.xml" </pre>
3	<p>Το APES παραλαμβάνει την πρόταση και αμέσως φορτώνει και ενεργοποιεί το σχετικό θέμα. (Μηνύματα του APES στην κονσόλα)</p> <pre data-bbox="325 1693 1362 1800"> Loading aspects from T:/examples/aop4bpmn2/nuclear-SMP-adaptation-ADVISE-2.aspect.xml...ok *** Aspect added and activated: T:/examples/aop4bpmn2/nuclear-SMP-adaptation-ADVISE-2.aspect.xml#nuclear_SMP_adaptation_ASPECT_2 on session aop4bpmn2-session-bde2df86_2fd5_4d87_92d7_7adc79b71ee2 </pre>
4	<p>Με την ενεργοποίηση του θέματος, το APES διαπιστώνει ότι θα πρέπει να εφαρμόσει άμεσα την προσαρμογή. Έτσι διακόπτει την τρέχουσα εργασία (“Study advice”) και εισάγει τις εργασίες του θέματος Aspect 2 αμέσως μετά (after) από αυτήν. Η πρώτη εργασία της προσαρμογής στέλνει αίτημα για επιπλέον διευκρινίσεις στο Ινστιτούτο Ραδιοπροστασίας. (Εκτέλεση προσαρμοσμένης διαδικασίας)</p>

	<pre> NUCLEAR_PHASE_II_RNA_PROCESS_ASK_FOR_CLARIFICATIONS___# <http://events.event-processing.org/ids/9fa95e1e-6049-4551-a7cf-f45634810762#event> a :NUCLEAR_PHASE_II_RNA_PROCESS_ASK_FOR_CLARIFICATIONS___ADVISE_2 ; :endTime "2012-09-22T11:32:27Z"^^xsd:dateTime ; <http://imu.ntua.gr/play/aop4bpnm2/nuclear-example/1#id> "9fa95e1e-6049-4551-a7cf-f45634810762" ; <http://imu.ntua.gr/play/aop4bpnm2/nuclear-example/1#message> "" ; <http://imu.ntua.gr/play/aop4bpnm2/nuclear-example/1#task> "(adv1_after) Ask for Clarifications" . </pre>
5	<p>Λήψη των διευκρινίσεων από το Ινστιτούτο Ραδιοπροστασίας. (Εκτέλεση προσαρμοσμένης διαδικασίας)</p> <pre> NUCLEAR_PHASE_II_RNA_PROCESS_CLARIFICATIONS_RESPONSE___# <http://events.event-processing.org/ids/4951af09-557f-4a4f-920c-78c19c23c5da#event> a :NUCLEAR_PHASE_II_RNA_PROCESS_CLARIFICATIONS_RESPONSE___ADVISE_2 ; :endTime "2012-09-22T11:32:29Z"^^xsd:dateTime ; <http://imu.ntua.gr/play/aop4bpnm2/nuclear-example/1#id> "4951af09-557f-4a4f-920c-78c19c23c5da" ; <http://imu.ntua.gr/play/aop4bpnm2/nuclear-example/1#message> "" ; <http://imu.ntua.gr/play/aop4bpnm2/nuclear-example/1#task> "(adv1_after2) Clarifications Response" . </pre>
6	<p>Ζητείται από τον υπεύθυνο να μελετήσει τις διευκρινίσεις (εργασία “Study Clarifications”). (Εκτέλεση προσαρμοσμένης διαδικασίας)</p> 
7	<p>Με την ολοκλήρωση της “Study Clarifications” ολοκληρώνεται και η προσαρμογή Aspect 2 κι έτσι συνεχίζεται κανονικά η εκτέλεση της κύριας διαδικασίας. (Εκτέλεση κύριας διαδικασίας)</p> 

Πίνακας 8-5. Εικόνες από την εφαρμογή της Aspect 3 (“Around”)

1	<p>Ο υπεύθυνος ολοκληρώνει την εργασία “Analyse data” και επιλέγει να ζητήσει τη γνωμοδότηση του Ινστιτούτου Ραδιοπροστασίας. (Εκτέλεση της κύριας διαδικασίας)</p> 
2	<p>Αποστέλλεται το αίτημα για γνωμοδότηση. (Εκτέλεση της κύριας διαδικασίας)</p> <pre data-bbox="507 701 1082 936"> NUCLEAR_PHASE_II_RNA_PROCESS_ASK_FOR_ADVICE <http://events.event-processing.org/ids/d2b39488-45e7-4a21-bce1-0e9884c254df#event> a :NUCLEAR_PHASE_II_RNA_PROCESS_ASK_FOR_ADVICE ; :endTime "2012-09-22T10:45:05Z"^^xsd:dateTime ; <http://imu.ntua.gr/play/aop4bpmn2/nuclear-example/1#id> "d2b39488-45e7-4a21-bce1-0e9884c254df" ; <http://imu.ntua.gr/play/aop4bpmn2/nuclear-example/1#message> = ; </pre>
3	<p>Καθώς αναμένεται η γνωμοδότηση του Ινστιτούτου λαμβάνονται νέες ειδοποιήσεις. Ο μηχανισμός SANs αντιδρά προτείνοντας την προσαρμογή Aspect 3.</p> <pre data-bbox="507 1077 1177 1709"> ADAPTATION_RECOMMENDATION_EVENT <http://events.event-processing.org/ids/arp-1348310786148999936#event> a :ADAPTATION_RECOMMENDATION_EVENT ; :endTime "2012-09-22T10:46:26Z"^^xsd:dateTime ; :stream <http://streams.event-processing.org/ids/AsyncAdaptationRecommendations#stream> ; <http://imu.ntua.gr/play/aop4bpmn2/adaptation-events/1#ask-user> "no" ; <http://imu.ntua.gr/play/aop4bpmn2/adaptation-events/1#engine-id> "aop4bpmn2-engine-1348310583083" ; <http://imu.ntua.gr/play/aop4bpmn2/adaptation-events/1#event-type> "IS_RECOMMENDATION" ; <http://imu.ntua.gr/play/aop4bpmn2/adaptation-events/1#id> "arp-1348310786148999936" ; <http://imu.ntua.gr/play/aop4bpmn2/adaptation-events/1#message> = ; <http://imu.ntua.gr/play/aop4bpmn2/adaptation-events/1#session-id> "aop4bpmn2-session-b1b56f32_ede0_40bb_b1f7_c587469304d2" ; <http://imu.ntua.gr/play/aop4bpmn2/adaptation-recommendation-events/1#action> "activate-aspect" ; <http://imu.ntua.gr/play/aop4bpmn2/adaptation-recommendation-events/1#aspects> "T:/examples/aop4bpmn2/nuclear-SMP-adaptation-ADVISE-3.aspect.xml" </pre>
4	<p>Το APES παραλαμβάνει την πρόταση και αμέσως φορτώνει και ενεργοποιεί το σχετικό θέμα. (Μηνύματα του APES στην κονσόλα)</p> <pre data-bbox="327 1832 1361 1933"> Loading aspects from I:/examples/aop4bpmn2/nuclear-SMP-adaptation-ADVISE-3.aspect.xml...ok ** Aspect added and activated: I:/examples/aop4bpmn2/nuclear-SMP-adaptation-ADVISE-3.aspect.xml#nuclear_SMP_adaptation_ASPECT_3 on session aop4bpmn2-session-b1b56f32_ede0_40bb_b1f7_c587469304d2 </pre>

5	<p>Κατόπιν λαμβάνεται η γνωμοδότηση του Ινστιτούτου Ραδιοπροστασίας.</p> <pre> NUCLEAR_PHASE_II_RNA_PROCESS_ADVICE_RESPONSE <http://events.event-processing.org/ids/2159644e-239a-4d1d-af22-dcd8865edb9b#event> a :NUCLEAR_PHASE_II_RNA_PROCESS_ADVICE_RESPONSE ; :endTime "2012-09-22T10:45:05Z"^^xsd:dateTime ; <http://imu.ntua.gr/play/aop4bpmn2/nuclear-example/1#id> "2159644e-239a-4d1d-af22-dcd8865edb9b" ; <http://imu.ntua.gr/play/aop4bpmn2/nuclear-example/1#message> "" ; </pre>
6	<p>Το APES ανιχνεύει ότι θα πρέπει να εφαρμοστεί η προσαρμογή Aspect 3 στο αμέσως επόμενο βήμα (πριν τη "Study advice"). Συνεπώς νέες εργασίες εισάγονται πριν και μετά (before and after) από την "Study advice". Η πρώτη εργασία του θέματος Aspect 3 είναι η "Analyse Alerts issued in last 2 minutes". (Εκτέλεση προσαρμοσμένης διαδικασίας)</p> 
7	<p>Έπειτα η διαδικασία συνεχίζει με την εκτέλεση της "Study advice". Βήμα Proceed της Aspect 3. (Εκτέλεση προσαρμοσμένης διαδικασίας)</p> 
8	<p>Κατόπιν η διαδικασία συνεχίζει με την αποστολή μιας περίληψης των ειδοποιήσεων προς το Ινστιτούτο Ραδιοπροστασίας. (Εκτέλεση προσαρμοσμένης διαδικασίας)</p> <pre> NUCLEAR_PHASE_II_RNA_PROCESS_FORWARD_SUMMARY_OF_ALERT <http://events.event-processing.org/ids/fa574278-15ec-4bab-be1e-10150d06ada8#event> a :NUCLEAR_PHASE_II_RNA_PROCESS_FORWARD_SUMMARY_OF_ALERTS__ADV ; :endTime "2012-09-22T10:50:00Z"^^xsd:dateTime ; <http://imu.ntua.gr/play/aop4bpmn2/nuclear-example/1#id> "fa574278-15ec-4bab-be1e-10150d06ada8" ; <http://imu.ntua.gr/play/aop4bpmn2/nuclear-example/1#message> "" ; <http://imu.ntua.gr/play/aop4bpmn2/nuclear-example/1#task> "(adv1_after) Forward summary of alerts to experts" ; </pre>

9	<p>Το Ινστιτούτο Ραδιοπροστασίας απαντά με μια ενημερωμένη γνωμοδότηση. (Εκτέλεση προσαρμοσμένης διαδικασίας)</p> <pre> NUCLEAR_PHASE_II_RNA_PROCESS_ADVICE_RESPONSE <http://events.event-processing.org/ids/b1cf0c27-0a78-4857-afa2-0547e9627516#event> a :NUCLEAR_PHASE_II_RNA_PROCESS_ADVICE_RESPONSE ; :endTime "2012-09-22T10:50:00Z"^^xsd:dateTime ; <http://imu.ntua.gr/play/aop4bpmn2/nuclear-example/1#id> "b1cf0c27-0a78-4857-afa2-0547e9627516" ; <http://imu.ntua.gr/play/aop4bpmn2/nuclear-example/1#message> ... </pre>
10	<p>Η Aspect 3 ολοκληρώνεται και συνεχίζει η εκτέλεση της κύριας διαδικασίας. (Εκτέλεση κύριας διαδικασίας)</p> 

8.6 Σύγκριση με Συστήματα Προτάσεων (recommender systems)

Κλείνοντας το κεφάλαιο αξίζει να σημειωθεί ότι τα συστήματα παραγωγής προτάσεων που υλοποιούνται στην πλατφόρμα προτάσεων προσαρμογών (ARP), διαφέρουν ουσιαστικά από τα συστήματα προτάσεων (recommender systems) που χρησιμοποιούνται σε πολλούς γνωστούς ιστοτόπους (websites) του Διαδικτύου. Αυτά ως επί τω πλείστο παρέχουν προτάσεις προς χρήστες – καταναλωτές, οι οποίες αφορούν συνήθως τυποποιημένα προϊόντα. Οι αλγόριθμοι που τυπικά χρησιμοποιούν στηρίζονται σε τεχνικές διήθησης περιεχομένου (content-based filtering), σε τεχνικές συνεργατικής διήθησης (collaborative filtering) ή σε υβριδικές μεθόδους.

Αντίθετα, η πλατφόρμα ARP δίνει τη δυνατότητα στον σχεδιαστή να μοντελοποιεί τη μέθοδο παραγωγής προτάσεων προσαρμογών που αυτός κρίνει ως καταλληλότερη για την εκάστοτε διαδικασία ή εφαρμογή την οποία σχεδιάζει. Ο λόγος που επιλέχθηκε αυτή η προσέγγιση έχει να κάνει με το γεγονός ότι οι γενικής χρήσης αλγόριθμοι των συστημάτων προτάσεων (συνεργατική διήθηση, διήθηση βασισμένη στο περιεχόμενο και οι παραλλαγές τους) είναι πιο κατάλληλοι στο να προτείνουν συγκεκριμένα προϊόντα, όπως για παράδειγμα ταινίες, μουσική, πακέτα ταξιδίων. Δεν μπορούν όμως να διαμορφώνουν νέες προτάσεις λαμβάνοντας υπόψη τις επικρατούσες συνθήκες του περιβάλλοντος. Ακολουθως αναφέρουμε ορισμένους επιπλέον λόγους.

- Τα γενικής χρήσης συστήματα προτάσεων επιλέγουν και προτείνουν από ένα σύνολο εναλλακτικών, είτε με βάση κάποια χαρακτηριστικά τους (content-based filtering), είτε με βάση ιστορικά στοιχεία όπως παλαιότερες χρήσεις ή βαθμολογίες χρηστών (collaborative filtering). Στην μεν πρώτη περίπτωση οι εναλλακτικές θα πρέπει να είναι περίπου ομοειδείς (π.χ. ταινίες ή βιβλία) ενώ στην δεύτερη

απαιτείται αρχείο με ιστορικά στοιχεία. Ωστόσο, στο πλαίσιο μιας συγκεκριμένης διαδικασίας, οι αναγκαίες προσαρμογές δεν μπορεί να είναι πάντα ομοειδείς (π.χ. με τον ίδιο σκοπό) ή να διαθέτουν συγκρίσιμα χαρακτηριστικά, ενώ συχνά δεν υπάρχουν ιστορικά στοιχεία χρήσης ή προτίμησής τους.

- Μια άλλη κατηγορία συστημάτων προτάσεων προσαρμογών είναι αυτή που αφορά συστήματα που εστιάζουν στην παραγωγή προσαρμογών με βάση μη λειτουργικά χαρακτηριστικά (non-functional attributes) των προσαρμοζόμενων διαδικασιών όπως το CEVICHE (Hermosillo et al., 2010). Ενδεικτικά αναφέρουμε το χρόνο απόκρισης, το ρυθμό εμφάνισης σφαλμάτων, τη διαθεσιμότητα, την ασφάλεια και άλλα. Ωστόσο αυτά αδυνατούν να προτείνουν προσαρμογές που εστιάζουν στα ιδιαίτερα λειτουργικά χαρακτηριστικά των εφαρμογών, τα οποία συνήθως διαφέρουν από περίπτωση σε περίπτωση.

Η χρήση εξειδικευμένων μοντέλων προτάσεων προσαρμογών για κάθε διαδικασία ή κατηγορία διαδικασιών ή πεδίο εφαρμογής (domain), μπορεί κατά τη δημιουργία των προτάσεων να λαμβάνει υπόψη την εσωτερική οργάνωση, τον τρόπο λειτουργίας και τα ιδιαίτερα χαρακτηριστικά αυτών. Επιπλέον οι προτάσεις μπορεί να παράγονται δυναμικά με βάση τις συνθήκες του περιβάλλοντος, την εσωτερική κατάσταση αλλά και τη γνώση του τρόπου λειτουργίας και των σκοπών των διαδικασιών.

Προκειμένου να διασφαλιστεί η ευρύτερη δυνατή εφαρμοσιμότητα της πλατφόρμας προτάσεων προσαρμογών (ARP) προτιμήθηκε η ανάπτυξη εξειδικευμένων μοντέλων προτάσεων (μοντελοποιημένων με SANs) αντί για χρήση κάποιων γενικών αλγορίθμων, ανεξάρτητων εφαρμογής και πεδίου. Η επιλογή αυτή έχει το μειονέκτημα ότι απαιτεί την ανάπτυξη ενός νέου μοντέλου προτάσεων για κάθε διαδικασία ή κατηγορία διαδικασιών ή πεδίο εφαρμογής. Με τη χρήση όμως των SANs και την προσανατολισμένη σε στόχους (goal-oriented) προσέγγισή τους, επιχειρείται η άμβλυση ως ένα βαθμό αυτού του προβλήματος. Τέλος, δεν αποκλείεται σε ορισμένες περιπτώσεις η χρήση αλγορίθμων προτάσεων πιο γενικής χρήσης, εφόσον όμως αυτοί ανταποκρίνονται στις ανάγκες και ιδιαιτερότητες κάποιας συγκεκριμένης διαδικασίας.

Στα κεφάλαια που ακολουθούν παρουσιάζεται μια επέκταση των SANs, η οποία προσθέτει τη δυνατότητα δυναμικής τους εξέλιξης, υπό την έννοια ότι επιτρέπει τον προσδιορισμό μερών των SANs που αφορούν στην επιλογή κατάλληλης αντίδρασης ακόμη και κατά την διάρκεια της εκτέλεσης. Με την επέκταση αυτή ολοκληρώνεται η πρόταση της διατριβής.

9 Δυναμικά Εξελισσόμενα SANs

Τα SANs αποτελούν μια καινοτόμα προσέγγιση μοντελοποίησης ικανή να λαμβάνει υπόψη τις καταστάσεις του περιβάλλοντος κατά την εκτέλεση και προσαρμογή διαδικασιών και συστημάτων. Ωστόσο η αποτύπωση αυτή αφορά τις εκ των προτέρων (a priori) γνωστές και άρα ως ένα βαθμό αναμενόμενες καταστάσεις. Κατά το σχεδιασμό όμως δεν είναι πάντα δυνατό να προβλεφθεί κάθε πιθανή κατάσταση, ούτε να προδιαγραφούν οι αντίστοιχες απαιτούμενες ενέργειες. Έτσι αν και καινοτόμα η προσέγγιση δεν παρέχει αφ' εαυτής προηγμένες δυνατότητες μοντελοποίησης απαραίτητες για την υλοποίηση εξελιγμένων χαρακτηριστικών, όπως είναι η ικανότητα αυτο-ρύθμισης (self-*⁶⁵), αυτό-ίασης (self-healing) και δυναμικής εξέλιξης.

Στο κεφάλαιο αυτό προτείνουμε μια επέκταση των SANs, η οποία προσθέτει δυνατότητες μοντελοποίησης SANs με δυναμικά και εξελίξιμα χαρακτηριστικά. Αυτό επιτυγχάνεται με τον ορισμό κατάλληλων δομών που επιτρέπουν μεν την περιγραφή των στόχων και των απαιτούμενων συμπεριφορών, χωρίς όμως να παρέχουν συγκεκριμένα σχέδια επίτευξης αυτών (δηλαδή κάποιας ανάλυσής τους σε κόμβους καταστάσεων, συνθηκών πλαισίου και ενεργειών). Συνεπώς κατά το σχεδιασμό οι στόχοι παραμένουν σε αφηρημένο (abstract) επίπεδο όσον αφορά τον τρόπο επίτευξής τους στην πράξη. Τα σχέδια επίτευξης των στόχων επιλέγονται και συγκεκριμενοποιούνται κατά την εκτέλεση του αντίστοιχου κόμβου-στόχου, σύμφωνα με κανόνες και παραμέτρους που έχουν τεθεί κατά τον σχεδιασμό.

Ωστόσο, είναι σημαντικό να υπάρχει ένας τυπικός τρόπος έκφρασης (formalism) που θα περιγράφει τη μέθοδο επιλογής και καθορισμού των εκάστοτε υλοποιήσεων των αφηρημένων στόχων ανάλογα με τις εκάστοτε επικρατούσες καταστάσεις. Η ορθή επιλογή ενεργειών και υποστόχων, ως ανταπόκριση σε συγκριμένες καταστάσεις, προϋποθέτει τη διαθεσιμότητα και αξιοποίηση πληροφοριών του περιβάλλοντος καθώς και πληροφοριών του πλαισίου (contextual and operational information) που μπορεί να αφορούν χρήστες, γεγονότα, υπηρεσίες ή άλλες οντότητες.

⁶⁵ Ο όρος "Self-*" ή "Self Star" αναφέρεται σε οποιαδήποτε ιδιότητα ή διαδικασία ενός συστήματος που ενεργοποιείται και τη διαχειρίζεται το ίδιο το σύστημα αφ' εαυτού. Είναι απαραίτητο χαρακτηριστικό των ζωντανών οργανισμών (βιολογία) καθώς και συστημάτων με τεχνητή νοημοσύνη που μιμούνται τις συμπεριφορές και ιδιότητες φυσικών αυτο-οργανωνόμενων και αυτο-ρυθμιζόμενων συστημάτων. [Πηγή: http://wiki.cas-group.net/index.php?title=Self-Star_Properties]

9.1 Εισαγωγή

Στο κεφάλαιο αυτό περιγράφουμε την προσθήκη ορισμένων προηγμένων χαρακτηριστικών στα SANs που επιτρέπουν την αυτόματη συμπλήρωση και επέκταση των δικτύων τους. Έτσι καθίσταται δυνατή η μοντελοποίηση SANs για τα οποία δεν γνωρίζουμε πλήρως (κατά το σχεδιασμό) όλες τις λεπτομέρειες λειτουργίας τους ή των σχετιζόμενων καταστάσεων του περιβάλλοντος. Έτσι παρέχεται η δυνατότητα για ευελιξία με ελλιπείς προδιαγραφές (flexibility by underspecification), όπως περιγράφηκε στην ενότητα 3.1. Τα μη επαρκώς προσδιορισμένα μοντέλα θα συμπληρώνονται, συγκεκριμενοποιούνται και υλοποιούνται κατά την εκτέλεσή τους, σύμφωνα πάντα με μεθόδους και στρατηγικές που θα προδιαγράφονται κατά τον σχεδιασμό. Τα οφέλη αυτής της προσέγγισης είναι διττά. Αφενός μειώνεται ο σχεδιαστικός φόρτος του αναλυτή που αναπτύσσει τα SANs και αφετέρου επιτρέπεται η προσθήκη νέων εναλλακτικών υλοποιήσεων για τις ελλιπώς προσδιορισμένες λειτουργίες, ακόμη και μέχρι τη στιγμή πριν την εκτέλεσή τους.

Ένας τρόπος για να επιτευχθεί η παραπάνω δυνατότητα είναι με προσθήκη νέων τύπων κόμβων, οι οποίοι θα επιλέγουν αυτόματα τις κατάλληλες υλοποιήσεις των συμπεριφορών από «βιβλιοθήκες επαναχρησιμοποιούμενων υλοποιήσεων». Τους κόμβους αυτούς τους καλούμε «Κόμβους Αφηρημένων Ενεργειών» (Abstract Action Nodes), υπό την έννοια ότι δεν προσδιορίζουν κάποια συγκεκριμένη ενέργεια παρά μόνο σε ένα υψηλό ή αφαιρετικό επίπεδο. Χρησιμοποιούνται στο δεξιό κλάδο των κόμβων-στόχων, όπως και οι κόμβοι απλών ενεργειών. Επίσης, τις συλλογές των εναλλακτικών (υποψήφιων) υλοποιήσεων των αφηρημένων ενεργειών τις καλούμε «Δεξαμενές Ενεργειών» (Action Pools). Οι μέθοδοι αναζήτησης και επιλογής των εναλλακτικών υλοποιήσεων μέσα στις δεξαμενές ενεργειών, καθώς και οι παράμετροί τους, επίσης αποτελούν αντικείμενο μελέτης αυτού του κεφαλαίου. Για τους σκοπούς αυτής της διατριβής χρησιμοποιούμε δύο συγκεκριμένες μεθόδους λήψης αποφάσεων με πολλαπλά κριτήρια (Multiple Criteria Decision Making (MCDM) methods), κυρίως λόγω της απλότητάς τους και της ευρείας αποδοχής τους. Στη βιβλιογραφία όμως έχουν προταθεί πολύ περισσότερες προσεγγίσεις που χρησιμοποιούν διαφορετικές τεχνικές και θεωρίες. Ευθύς αμέσως παρουσιάζουμε τις πιο ενδιαφέρουσες.

9.2 Μέθοδοι και εργαλεία για Λήψη Αποφάσεων με Πολλαπλά Κριτήρια

Οι Μέθοδοι Λήψης Αποφάσεων με Πολλαπλά Κριτήρια ή *Multiple Criteria Decision-Making (MCDM)*, ορίζονται ως ένα σύνολο μεθοδολογιών για τη σύγκριση, ταξινόμηση και επιλογή από ένα σύνολο εναλλακτικών επιλογών που χαρακτηρίζονται από πολλαπλές παραμέτρους (Habiba and Asghar, 2009) και οι οποίες χρησιμοποιούνται ως κριτήρια. Στη βιβλιογραφία έχουν προταθεί διάφορες κατηγοριοποιήσεις των μεθόδων αυτών που εξαρτώνται κυρίως από την οπτική του κάθε ερευνητή ή τη φύση της μεθοδολογίας κατηγοριοποίησης που χρησιμοποιήθηκε. Στην παρούσα εργασία υιοθετούμε την πρόταση των Zorounidis and Pardalos (2010) για κατηγοριοποίηση των MCDM μεθόδων, σε αυτές που εμπίπτουν στη Θεωρία Χρησιμότητας με Πολλαπλά Χαρακτηριστικά (Multi-Attribute Utility Theory ή MAUT), σε αυτές που εξετάζουν Σχέσεις Υπεροχής (Outranking Relations),

στον Προγραμματισμό προτιμήσεων (Preference Programming) καθώς και στις μεθόδους Βελτιστοποίησης Πολλαπλών Στόχων (Multi-Objective Optimization ή MOO).

9.2.1 Μέθοδοι της Θεωρίας Χρησιμότητας με Πολλαπλά Κριτήρια

Οι μέθοδοι που ανήκουν στην κατηγορία αυτή (Multi-Attribute Utility Theory, MAUT) εστιάζουν στον συγκερασμό (aggregation) των διαφορετικών απόψεων των αποφασιζόντων (decision makers), σε σχέση με συγκεκριμένα κριτήρια, σε μία ενιαία συνάρτηση χρησιμότητας (utility function). Στην πράξη εξετάζουν τις μαθηματικές συνθήκες του συγκερασμού. Η χρησιμότητα (utility) είναι ένας πραγματικός αριθμός που αναπαριστά τη «συνολική» προτίμηση σε μια υπό εξέταση εναλλακτική επιλογή. Οι συναρτήσεις χρησιμότητας (utility functions) χρησιμοποιούνται για να μετασχηματίσουν σε μία κοινή και αδιάστατη κλίμακα, τις εκάστοτε τιμές (συνήθως αριθμητικές) που λαμβάνει κάθε εναλλακτική επιλογή σε σχέση με τα υπό εξέταση κριτήρια. Οι τιμές των κριτηρίων μπορεί να είναι είτε «ακριβείς», δηλαδή αντικειμενικές, σαφείς και ποσοτικοποιημένες, είτε «κατ' εκτίμηση», δηλαδή υποκειμενικές και ποιοτικές. Οι προσεγγίσεις που στηρίζονται στην MAUT προβλέπουν την με κάποιο τρόπο ανάθεση στα κριτήρια κατάλληλων βαρών που απεικονίζουν τη σχετική τους σημαντικότητα (Fülöp, 2005). Επίσης, οι μέθοδοι αυτές επιτρέπουν πλήρη αντιστάθμιση μεταξύ των κριτηρίων, δηλαδή το «κέρδος» ενός κριτηρίου μπορεί να αντισταθμίσει την «απώλεια» ενός άλλου (Keeney, 1976). Παραδείγματα μεθόδων MAUT είναι η Διαδικασία Ιεραρχικής Ανάλυσης (Analytical Hierarchy Process ή AHP) (Saaty, 1980), η Διαδικασία Δικτυακής Ανάλυσης (Analytic Network Process ή ANP) (Saaty, 1996), η Απλή Τεχνική Βαθμολόγησης Πολλαπλών Κριτηρίων (Simple Multi-Attribute Rating Technique ή SMART) (Edwards and Barron, 1994; Edwards, 1977), και η μέθοδος των Γενικευμένων Μέσων (Generalized Means) (Mészáros and Rapcsák, 1996). Αυτές οι μέθοδοι υλοποιούνται και από αντίστοιχα πακέτα λογισμικού όπως το Logical decision, το Expert Choice και το Criterium.

Έχουν προταθεί αρκετές απλές μέθοδοι MAUT, συμπεριλαμβανομένης της SMART, καθώς και αρκετές μέθοδοι υπολογισμού διαφόρων μέσων. Η SMART είναι η απλούστερη όλων αφού η τελική χρησιμότητα κάθε εναλλακτικής υπολογίζεται ως ο σταθμισμένος μέσος όρος των τιμών που λαμβάνει σε κάθε κριτήριο. Στη βιβλιογραφία περιγράφονται αρκετές τεχνικές υπολογισμού διαφόρων τύπων μέσων όρων (Mészáros and Rapcsák, 1996).

9.2.2 Διαδικασία Ιεραρχικής Ανάλυσης και Διαδικασία Δικτυακής Ανάλυσης

Η Διαδικασία Ιεραρχικής Ανάλυσης (AHP) αναπτύχθηκε από τον Thomas L. Saaty (Saaty, 1980) και πρόκειται για μία μέθοδο υποστήριξης στη λήψη περίπλοκων αποφάσεων. Βασίζεται σε δυαδικές συγκρίσεις (ανά ζεύγη) που πραγματοποιούν εμπειρογνώμονες, για εκτίμηση της σχετικής προτίμησής τους στις εναλλακτικές λύσεις ενός προβλήματος, επί τη βάση ποσοτικών και ποιοτικών κριτηρίων. Η AHP προβλέπει τη δόμηση των κριτηρίων σε μία (δενδρική) ιεραρχία, την εκτίμηση της σχετικής τους σημαντικότητας, τη σύγκριση των εναλλακτικών επιλογών για κάθε κριτήριο ξεχωριστά και τέλος τον προσδιορισμό μιας συνολικής κατάταξης των εναλλακτικών. Η AHP αναλύει περίπλοκες αποφάσεις σε μια σειρά από δυαδικές συγκρίσεις κι έπειτα συνθέτει το τελικό αποτέλεσμα.

Η Διαδικασία Δικτυακής Ανάλυσης (ANP) επίσης αναπτύχθηκε από τον Thomas L. Saaty και αποτελεί μια γενίκευση της AHP. Είναι μια διαδικασία για τη δόμηση των κριτηρίων επιλογής σε ένα δίκτυο από αλληλοεξαρτώμενους κόμβους, την εκτίμηση της σχετικής τους σημαντικότητας, τη δυαδική σύγκριση ανά κριτήριο των εναλλακτικών και τέλος τον προσδιορισμό της τελικής κατάταξής τους (Saaty, 1999). Η ANP υπολογίζει τη σχετική σημαντικότητα των εναλλακτικών από τις ξεχωριστές τιμές των κόμβων που αλληλεπιδρούν στο δίκτυο. Οι κόμβοι του δικτύου οργανώνονται σε συστάδες (network clusters) και επιτρέπεται η αλληλεπίδραση και ανατροφοδότηση τόσο μεταξύ κόμβων που ανήκουν στην ίδια συστάδα (καλείται εσωτερική αλληλεξάρτηση), όσο και των συστάδων μεταξύ τους (καλείται εξωτερική αλληλεξάρτηση) (Saaty, 1999). Η ANP είναι πιο γενική από την AHP αφού δομεί το πρόβλημα της λήψης αποφάσεων σε ένα γενικό δίκτυο σχέσεων αντί σε μία ιεραρχία ανεξάρτητων κριτηρίων όπως στην AHP.

9.2.3 Ασαφή Σύνολα και Ασαφής Πολυκριτηριακή Λήψη Αποφάσεων

Τα Ασαφή Σύνολα (Fuzzy Sets) και οι Ασαφείς Αριθμοί (Fuzzy Numbers) εισήχθησαν το 1965 από τον Zadeh στη πρωτοποριακή του εργασία “Fuzzy Sets” (Zadeh, 1965). Στα χρόνια που ακολούθησαν πολλοί ερευνητές πρότειναν διάφορους ορισμούς για αριθμητικές και πράξεις με ασαφείς αριθμούς, ενώ αρκετές κλασικές μέθοδοι και αλγόριθμοι μετατράπηκαν ή επεκτάθηκαν έτσι ώστε να μπορούν να χειρίζονται και ασαφή δεδομένα. Κατ’ αυτόν τον τρόπο κατέστη δυνατή η εφαρμογή τους και σε προβλήματα που χαρακτηρίζονται από ασάφεια. Η μετατροπή μεθόδων και αλγορίθμων στις αντίστοιχες ασαφείς επιτυγχάνεται ως επί τω πλείστο χρησιμοποιώντας την Αρχή Επέκτασης του Zadeh (Extension Principle of Zadeh) (Zadeh, 1975).

Όπως αναφέρουν οι Bellman and Zadeh (1970), «Ως λήψη αποφάσεων σε ένα ασαφές περιβάλλον νοείται η διαδικασία λήψης αποφάσεων στην οποία οι στόχοι ή/και οι περιορισμοί, αλλά όχι αναγκαία και το υπό εξέταση σύστημα, είναι από τη φύση τους ασαφείς. Αυτό σημαίνει ότι οι στόχοι και οι περιορισμοί συγκροτούν κλάσεις εναλλακτικών, των οποίων τα όρια δεν είναι επακριβώς ορισμένα»⁶⁶.

9.2.4 Γλωσσική Διατεταγμένη Σταθμισμένη Άθροιση

Η μέθοδος της Γλωσσικής Διατεταγμένης Σταθμισμένης Άθροισης (Linguistic Ordered Weighted Average, LOWA) περιγράφηκε από τους Herrera et al. (1996) και Herrera et al. (1997). Βασίζεται στις έννοιες της Διατεταγμένης Σταθμισμένης Άθροισης (Ordered Weighted Average ή OWA) που προτάθηκε από τον Yager (1988) σε συνδυασμό με τις γλωσσικές ετικέτες που ορίστηκαν από τους Delgado et al. (1993). Πρακτικά η LOWA είναι μια πράξη που εφαρμόζεται σε γλωσσικές μεταβλητές (linguistic variables), όπως λέξεις ή φράσεις μιας φυσικής ή τεχνητής γλώσσας, και παράγει ένα επίσης γλωσσικό αποτέλεσμα. Όπως υποστηρίζουν οι Ben-Arieh and Chen (2006), «Στον πραγματικό κόσμο η αβεβαιότητα, οι περιορισμοί ακόμη και η ασάφεια στη γνώση των εμπειρογνομώνων υποδηλώνουν ότι οι

⁶⁶ “By decision-making in a fuzzy environment is meant a decision process in which the goals and/or the constraints, but not necessarily the system under control, are fuzzy in nature. This means that the goals and/or the constraints constitute classes of alternatives whose boundaries are not sharply defined”. (Bellman & Zadeh, 1970)

αποφασίζοντας δεν δύνανται να παρέχουν ακριβείς τιμές για να εκφράσουν τις γνώμες τους. Η χρήση γλωσσικών όρων καθιστά τις επιλογές των εμπειρογνομόνων πιο αξιόπιστες και συνεπείς»⁶⁷.

Η LOWA εφαρμόζεται σε γλωσσικές εκτιμήσεις (linguistic evaluations) που δίνονται από τους αποφασίζοντας, στις εναλλακτικές λύσεις ενός προβλήματος, ως προς ένα σύνολο από κριτήρια. Οι γλωσσικές εκτιμήσεις λαμβάνουν τιμές από ένα διατεταγμένο σύνολο γλωσσικών όρων (linguistic term set), όπως για παράδειγμα «Πολύ λίγο, Λίγο, Μέτριο, Αρκετό και Πάρα πολύ». Κάθε όρος έχει μια θέση μέσα στο διατεταγμένο σύνολο, ήτοι το «Πολύ λίγο» έχει τη θέση 1 ενώ το «Πάρα πολύ» τη θέση 5. Η LOWA υπολογίζει μια συνολική εκτίμηση της επίδοσης κάθε εναλλακτικής, λαμβάνοντας ως είσοδο (α) μια σειρά από γλωσσικές εκτιμήσεις των επιδόσεών της ως προς τα κριτήρια και (β) τα αντίστοιχα βάρη των κριτηρίων. Το αποτέλεσμα της LOWA είναι επίσης ένας γλωσσικός όρος.

9.2.5 Μέθοδοι Σχέσεων Υπεροχής

Οι μέθοδοι αυτές επικεντρώνονται στην αποτίμηση σχέσεων μεταξύ των εναλλακτικών που είναι γνωστές ως σχέσεις υπεροχής (outranking relations) και αναπαριστούν τις προτιμήσεις του αποφασίζοντα (Escobar-Toledo and López-García, 2005). Η βασική ιδέα είναι ότι μια εναλλακτική A υπερέχει της εναλλακτικής B αν για ένα μεγάλο μέρος των κριτηρίων η A είναι τουλάχιστο το ίδιο καλή με τη B (συνθήκη συμφωνίας ή concordance condition), ενώ για τα υπόλοιπα κριτήρια η A έχει αποδεκτές επιδόσεις (συνθήκη μη διαφωνίας ή non-discordance condition). Κατ' αυτόν τον τρόπο, αφού προσδιοριστεί για κάθε ζεύγος εναλλακτικών ποια απ' τις δύο υπερέχει, συνδυάζονται τα αποτελέσματα των συγκρίσεων σε μία μερική ή πλήρη διάταξη (ταξινόμηση) των εναλλακτικών (Füßler, 2005). Οι σχέσεις υπεροχής προτάθηκαν αρχικά από τον Roy (1968).

Στις μεθόδους υπεροχής ενυπάρχει η υπόθεση ότι ο αποφασίζων δεν επιθυμεί ή δεν μπορεί να καθορίσει «αμοιβαίες αντισταθμίσεις» (trade-offs) μεταξύ των επιδιωκόμενων στόχων (Roy and Bouyssou, 1993). Αυτό έχει ως επακόλουθο να μην είναι δυνατή η παραγωγή κάποιας συνάρτησης χρησιμότητας (όπως στις μεθόδους MAUT). Έτσι η οποιαδήποτε ταξινόμηση ή κατηγοριοποίηση των εναλλακτικών θα πρέπει να γίνει στη βάση δυαδικών συγκρίσεων (pair-wise comparisons). Παραδείγματα μεθόδων υπεροχής είναι η οικογένεια μεθόδων PROMETHEE (Brans et al., 1984), η οικογένεια μεθόδων ELECTRE (Roy, 1968), και η MACBETH (Bana e Costa and Vansnick, 1997).

Αντίθετα με τις μεθόδους MAUT όπου οι εναλλακτικές με την καλύτερη χρησιμότητα θεωρούνται ως οι πλέον κατάλληλες προς επιλογή, οι ταξινομήσεις που δίνουν οι μέθοδοι υπεροχής μπορεί να μην καταδεικνύουν απευθείας την καλύτερη επιλογή. Με αυτές είναι δυνατό να προσδιοριστεί ένα υποσύνολο των εναλλακτικών που υπερέχουν όλων των άλλων. Ο στόχος τους είναι αυτό το υποσύνολο να γίνεται όσο το δυνατό πιο μικρό και έτσι

⁶⁷ "In the real world, the uncertainty, constraints, and even the vague knowledge of the experts imply that decision makers cannot provide exact numbers to express their opinions. The use of linguistic labels makes expert judgment more reliable and consistent." (Ben-Arieh & Chen, 2006)

ουσιαστικά να αποτελεί μια shortlist μέσα απ' την οποία θα επιλεγεί η πλέον κατάλληλη εναλλακτική, με χρήση όμως άλλων μεθόδων ή με κάποιους συμβιβασμούς (Fülöp, 2005).

Μια αξιοπρόσεκτη υποκατηγορία μεθόδων υπεροχής είναι η οικογένεια ELECTRE. Η προσέγγιση που ακολουθούν χρησιμοποιεί τα δεδομένα του προβλήματος καθώς και κάποιες τιμές κατωφλίων (threshold values) για να εκτιμήσει τον βαθμό στον οποίο μία εναλλακτική υπερέρχει των υπολοίπων (X. Wang, 2007). Η βασική μέθοδος εισήχθη αρχικά από τους Benayoun et al. (1966). Σε σύντομο όμως χρονικό διάστημα από την παρουσίαση της πρώτης εκδοχής της μεθόδου, γνωστής ως ELECTRE I (Roy, 1968), αναπτύχθηκε ένας αριθμός παραλλαγών. Σήμερα δύο από τις πλέον χρησιμοποιούμενες εκδοχές της είναι η ELECTRE II (Roy and Bertier, 1971 και 1973) και η ELECTRE III (Roy, 1978). Επίσης μια άλλη παραλλαγή της ELECTRE είναι και η μέθοδος TOPSIS (Hwang and Yoon, 1981).

9.2.6 Προγραμματισμός Προτιμήσεων και Μέθοδοι Ανάλυσης Προτιμήσεων

Οι μέθοδοι προγραμματισμού προτιμήσεων (Preference Programming) συμπεριλαμβάνουν τεχνικές που μοντελοποιούν μη πλήρεις πληροφορίες για τις προτιμήσεις των αποφασιζόντων και παρέχουν συστάσεις για λήψη αποφάσεων βασισμένες σε έννοιες κυριαρχίας (dominance concepts) και σε κανόνες λήψης αποφάσεων (decision rules). Επίσης υποστηρίζουν την επαναληπτική διερεύνηση των προτιμήσεων των αποφασιζόντων (Salo and Hämmäläinen, 2010).

Παρόμοια με τον προγραμματισμό προτιμήσεων, οι τεχνικές ανάλυσης προτιμήσεων (Preference Disaggregation) αναπτύσσουν ένα μοντέλο αποφάσεων υποστηρίζοντας μια διαδικασία εκμαίευσης (elicitation). Αυτή η προσέγγιση είναι ιδιαίτερα χρήσιμη όταν ο αποφασίζων δεν δύναται να προσδιορίσει λεπτομερώς τις παραμέτρους των προτιμήσεών του που απαιτούνται από μια διαδικασία λήψης αποφάσεων. Η ανάλυση προτιμήσεων χρησιμοποιεί τεχνικές παρόμοιες με την παρεμβολή (regression-like techniques) για να «εξάγει» τις ζητούμενες παραμέτρους από ένα σύνολο παραδειγμάτων (Siskos and Grigoroudis, 2010). Μια αντιπροσωπευτική μέθοδος αυτής της κατηγορίας είναι η UTA.

Η μέθοδος UTA (Jacquet-Lagrange and Siskos, 1982) αποσκοπεί στην εξαγωγή μοντέλων λήψης αποφάσεων από εκ των προτέρων (a priori) γνωστές αποφάσεις ή προτιμήσεις, υπό τη μορφή προσθετικών συναρτήσεων χρησιμότητας (additive utility functions). Λαμβάνει δε ως είσοδο τις εκτιμήσεις ενός αποφασίζοντα για τις επιδόσεις των εναλλακτικών ως προς ένα σύνολο κριτηρίων. Μια προσθετική συνάρτηση χρησιμότητας είναι ένα πραγματικό πολυώνυμο, πολλών μεταβλητών, πρώτης τάξης, όπου η κάθε μεταβλητή αναπαριστά την εκτίμηση (ενός αποφασίζοντα) για μια εναλλακτική σε σχέση με κάποιο κριτήριο. Η διαδικασία ξεκινά με την επιλογή ενός αντιπροσωπευτικού δείγματος εναλλακτικών που καλείται σύνολο αναφοράς, για το οποίο δίνεται από τον αποφασίζοντα μια ταξινόμησή τους από την καλύτερη στη χειρότερη ή απ' την προτιμότερη στη λιγότερη προτιμώμενη. Ακολούθως χρησιμοποιούνται τεχνικές γραμμικού προγραμματισμού για να εξαχθεί το προσωπικό μοντέλο λήψης αποφάσεων του αποφασίζοντα, υπό τη μορφή μιας συνάρτησης χρησιμότητας που ταξινομεί το σύνολο αναφοράς όσο πιο όμοια γίνεται με τη δεδομένη ταξινόμηση. Τέλος γίνεται έλεγχος και βελτιστοποίηση του αποτελέσματος με μια τεχνική ανάλυσης ευαισθησίας. Ο αρχικός αλγόριθμος της UTA έχει προσαρμοστεί και επεκταθεί

για διάφορες εφαρμογές με συνέπεια να εξελιχθεί σε μια ολόκληρη οικογένεια μεθόδων. Μια αναδρομή στην ιστορία της UTA, τις βασικές αρχές της και των παραλλαγών της δίνεται από τους Siskos et al. (2005). Πιο πρόσφατα οι Patiniotakis et al. (2011) πρότειναν τη Fuzzy UTASTAR, μια επέκταση της μεθόδου UTASTAR, η οποία μπορεί να χειρίζεται τόσο σαφή όσο και ασαφή δεδομένα.

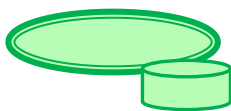
9.2.7 Βελτιστοποίηση Πολλαπλών Στόχων

Η Βελτιστοποίηση Πολλαπλών Στόχων (Multi-Objective Optimization, MOO) εξετάζει προβλήματα που αφορούν την ταυτόχρονη βελτιστοποίηση περισσότερων της μίας συναρτήσεων χρησιμότητας ή αλλιώς συναρτήσεων στόχου (objective functions). Η MOO μπορεί να εφαρμοστεί σε προβλήματα όπου η βέλτιστη απόφαση θα πρέπει να ληφθεί με συμβιβασμούς και αμοιβαία αντιστάθμιση (trade-off) μεταξύ αλληλοσυγκρουόμενων στόχων. Έχουν αναπτυχθεί αρκετές προσεγγίσεις MOO όπως οι μέθοδοι «Μη προτίμησης» (No-preference methods) (Hwang and Masud, 1979), οι «Εκ των προτέρων» και «Εκ των υστέρων» μέθοδοι (A priori and A posteriori methods), «Αλληλεπιδραστικές μέθοδοι» (interactive methods) (Miettinen et al., 2008; Miettinen, 1999) και υβριδικές μέθοδοι. Ο Προγραμματισμός Στόχων (Goal Programming) και η Εξελικτική Βελτιστοποίηση Πολλαπλών Στόχων (Evolutionary Multiobjective Optimization ή EMO) είναι παραδείγματα «Εκ των υστέρων» μεθόδων MOO. Ενδιαφέροντα παραδείγματα υβριδικών μεθόδων είναι οι συνδυασμοί μεθόδων MCDM και EMO (Miettinen et al., 2008; Sindhya et al., 2008; Sindhya et al., 2011).

Κλείνοντας αυτή τη συνοπτική παρουσίαση πολυκριτηριακών μεθόδων λήψης αποφάσεων αναφέρουμε ότι οι περισσότερες μπορούν να αξιοποιηθούν για την επιλογή μεταξύ των εναλλακτικών υλοποιήσεων αφηρημένων ή μη επαρκώς προσδιορισμένων στόχων των SANs (abstract or underspecified goals), δηλαδή συγκεκριμένων (concrete) ενεργειών και σχεδίων εκτέλεσης. Στη συνέχεια του κεφαλαίου παρουσιάζεται αναλυτικά η προτεινόμενη προσέγγιση για επέκταση των SANs με δυνατότητες δυναμικής μοντελοποίησης και εξέλιξης, με αξιοποίηση μεθόδων πολυκριτηριακής λήψης αποφάσεων.

9.3 Προσέγγιση Επέκτασης των SANs

Η λεπτομερής μοντελοποίηση των διαδικασιών και ιδίως η εξαντλητική αποτύπωση όλων των πιθανών καταστάσεων και συνθηκών μπορεί να αποδειχθεί μια πολύ κοπιαστική και χρονοβόρα εργασία. Επιπλέον δεν είναι πάντα εκ των προτέρων (a priori) γνωστές όλες οι παράμετροι μιας διαδικασίας ή προβλήματος ή μπορεί να υπάρχουν πολλές εναλλακτικές αναλύσεις στόχων, με επακόλουθο να καθίσταται μη πρακτική ή και εντελώς αδύνατη η ανάπτυξη λεπτομερών μοντέλων διαδικασιών. Τα SANs, ως μέθοδος μοντελοποίησης των αντιδράσεων των διαδικασιών σε καταστάσεις του περιβάλλοντος «πάσχει» από το προαναφερθέν πρόβλημα. Για την επίλυσή του προτείνουμε μια επέκταση που θα τους επιτρέπει να επιλέγουν δυναμικά (κατά την εκτέλεση) τις αναλύσεις των στόχων σε ενέργειες και λεπτομερή σχέδια εκτέλεσης.



Γι' αυτόν το λόγο εισάγουμε έναν νέο τύπο κόμβου στα βασικά δομικά στοιχεία των SANs, ο οποίος ονομάζεται κόμβος «Αφηρημένης Ενέργειας» (*Abstract Action node*) και με γραφικό συμβολισμό όπως φαίνεται στο σχήμα παραπλεύρως. Πρόκειται για τερματικούς κόμβους (*leaf nodes*) των SANs, οι οποίοι προσδιορίζουν δυναμικά τις ενέργειες που θα πρέπει να εκτελέσουν, τη στιγμή που η διάσχιση του γράφου του αντίστοιχου στιγμιότυπου SAN φθάνει σε αυτούς. Τότε εκτελείται ένας προκαθορισμένος (από το σχεδιασμό) αλγόριθμος επιλογής της κατάλληλης ενέργειας ή της κατάλληλης ανάλυσης σε υποστόχους (και ενέργειες). Δηλαδή αν και λέγεται κόμβος αφηρημένης «ενέργειας», αφορά τόσο ενέργειες όσο και υποστόχους. Στην πράξη δεν είναι πάντα γνωστό εκ των προτέρων αν ένας κόμβος αφηρημένης ενέργειας θα υλοποιηθεί από μια απλή ενέργεια ή από μια ανάλυση σε υποστόχους.

Κάθε μηχανισμός SANs μπορεί να παρέχει τους δικούς του αλγόριθμους επιλογής. Στο πλαίσιο της παρούσας διατριβής προτείνουμε και περιγράφουμε έναν ευέλικτο τέτοιο αλγόριθμο, ο οποίος στηρίζεται στην αναζήτηση εναλλακτικών υλοποιήσεων μέσα από ένα σύνολο επιλογών που καλείται «*Δεξαμενή Ενεργειών*» (*Action Pool*). Ο αλγόριθμος αυτός είναι αρκετά ευέλικτος και μπορεί να χρησιμοποιήσει διάφορες μεθόδους αναζήτησης και επιλογής (*search and selection methods*). Τόσο η δεξαμενή ενεργειών όσο και η μέθοδος αναζήτησης και επιλογής μπορούν να προσδιορίζονται είτε κατά το σχεδιασμό είτε δυναμικά κατά την εκτέλεση του SAN ως παράμετροι του κόμβου αφηρημένης ενέργειας. Έτσι, το SAN εμφανίζεται να έχει την ικανότητα να αυτό-ρυθμίζεται (*self-configuration*). Οι εναλλακτικές που περιέχονται στις δεξαμενές ενεργειών μπορεί να είναι από απλές ενέργειες (*primitive actions*) έως και ολόκληρα SANs, χωρίς να υπάρχει περιορισμός στην ανάμιξη διαφορετικών τύπων εναλλακτικών μέσα στην ίδια δεξαμενή. Μετά την επιλογή μιας εναλλακτικής ο κόμβος αφηρημένης ενέργειας αντικαθιστάται από αυτήν και εκτελείται.

Κατά την αναζήτηση σε μια δεξαμενή ενεργειών μπορεί να επιλεγεί καμία, μία ή πολλές εναλλακτικές. Αν επιλεγεί ακριβώς μία τότε εκτελείται και η ένδειξη επιτυχίας ή αποτυχίας (*success / failure*) που επιστρέφει είναι επίσης η επιστρεφόμενη ένδειξη και του κόμβου αφηρημένης ενέργειας. Αν δεν επιλεγεί καμία ενέργεια τότε ο κόμβος αποτυγχάνει (*failure*). Αν ταιριάζουν περισσότερες από μία εναλλακτικές τότε χρησιμοποιείται μια πολιτική επίλυσης (*resolution policy*) προκειμένου να αποφασιστεί πως θα αντιδράσει ο κόμβος σε αυτήν την κατάσταση. Η πολιτική επίλυσης καθορίζει επίσης και τον τρόπο προσδιορισμού της επιστρεφόμενης ένδειξης επιτυχίας ή αποτυχίας του κόμβου.

Όπως πρέπει ήδη να διαφαίνεται από τα παραπάνω, ένας κόμβος αφηρημένης ενέργειας χρειάζεται τρεις παραμέτρους για να ρυθμιστεί και να λειτουργήσει σωστά. Αυτές είναι μια δεξαμενή ενεργειών, μια μέθοδος αναζήτησης και επιλογής εναλλακτικών και μια πολιτική επίλυσης.

9.3.1 Δεξαμενές Ενεργειών (*Action Pools*)

Μια δεξαμενή ενεργειών περιέχει εναλλακτικές υλοποιήσεις μιας αφηρημένης ενέργειας, οι οποίες (εναλλακτικές) είναι είτε απλές ενέργειες (*primitive actions*) είτε αναλύσεις

στόχων και υποστόχων. Όλες τους θα πρέπει να παρέχουν πιθανές λύσεις του ίδιου προβλήματος. Με άλλα λόγια να παρέχουν ισοδύναμες λειτουργικότητες – συμπεριφορές που επιτυγχάνουν τον ίδιο στόχο. Μια δεξαμενή ενεργειών μπορεί να χρησιμοποιείται από πολλούς κόμβους αφηρημένων ενεργειών που ανήκουν ενδεχομένως σε διαφορετικά SANs. Για το λόγο αυτό οι δεξαμενές ενεργειών μπορεί να είναι αποθηκευμένες ξεχωριστά από τα SANs που τις χρησιμοποιούν.

Οι κόμβοι αφηρημένων ενεργειών, αντί για προκαθορισμένες απ' το σχεδιασμό δεξαμενές ενεργειών μπορεί να χρησιμοποιούν εκφράσεις (expressions) που αποτιμούνται σε κάθε τους εκτέλεση, προκειμένου να προσδιορίζουν δυναμικά τη δεξαμενή ενεργειών που θα χρησιμοποιούν κάθε φορά. Έτσι η αντιστοίχιση μεταξύ κόμβων αφηρημένων ενεργειών και δεξαμενών ενεργειών μπορεί να μεταβάλλεται δυναμικά κατά τη διάρκεια της εκτέλεσης ενός SAN. Αξίζει ακόμη να σημειωθεί ότι οι δεξαμενές ενεργειών μπορεί να είναι είτε προκαθορισμένα εκ των προτέρων σύνολα ενεργειών (απλών ή σύνθετων), είτε μπορεί να παράγονται, συνθέτονται ή άλλως αποκτώνται όταν ζητηθεί (on-demand), από κατάλληλες πηγές, όπως για παράδειγμα από βάσεις δεδομένων ή υπηρεσίες ιστού (web services).

9.3.2 Μετα-δεδομένα των SANs και Δεξαμενές Ενεργειών

Τα μετα-δεδομένα των SANs είναι πληροφορίες που εμπλουτίζουν και συμπληρώνουν την περιγραφή και τη σημασιολογία των κόμβων των SANs καθώς και άλλων σχετικών δομών (όπως για παράδειγμα οι δεξαμενές ενεργειών). Μπορούν να προστεθούν σε οποιοδήποτε είδος κόμβου και διαφέρουν από στιγμιότυπο σε στιγμιότυπο. Τα μετα-δεδομένα ορίζονται ως τριάδες αποτελούμενες από ένα μοναδικό όνομα (ανά κόμβο), έναν τύπο δεδομένων (ακέραιος αριθμός, πραγματικός αριθμός, ημερομηνία, αλφαριθμητικό ή λογική τιμή), και μια τιμή (value) με τύπο ίδιο με αυτόν που αναφέρεται στον τύπο δεδομένων.

Μετα-δεδομένο ::= (Όνομα, Τύπος, Τιμή)

Σε ένα συγκεκριμένο στιγμιότυπο ενός αντικειμένου (κόμβος, SAN ή άλλο) δεν μπορεί να χρησιμοποιηθεί το ίδιο όνομα μετα-δεδομένου περισσότερες από μία φορές.

Οι εναλλακτικές των δεξαμενών ενεργειών μπορούν επίσης να εμπλουτιστούν με μετα-δεδομένα (metadata), τα οποία θα επιτρέψουν την ευφυέστερη αναζήτηση και επιλογή τους. Προφανώς τα απαιτούμενα μετα-δεδομένων εξαρτώνται από τις μεθόδους αναζήτησης και επιλογής που πρόκειται να χρησιμοποιηθούν.

Επιπλέον και οι ίδιες οι δεξαμενές ενεργειών ως οντότητες μπορεί να έχουν μετα-δεδομένα. Στην υλοποίηση που προτείνουμε χρησιμοποιούμε την εξής σύμβαση. Τα μετα-δεδομένα των δεξαμενών ενεργειών ορίζουν το ελάχιστο κοινό σύνολο μετα-δεδομένων που θα πρέπει να έχουν όλες οι εναλλακτικές που περιέχονται σε αυτές. Με άλλα λόγια, οποιαδήποτε μετα-δεδομένα (ως ονόματα και τύπους) έχει μια δεξαμενή ενεργειών θα πρέπει να έχουν και οι εναλλακτικές που περιέχει. Φυσικά αυτές μπορεί να διαθέτουν και κάποια επιπλέον μετα-δεδομένα, ωστόσο όλες τους πρέπει να διαθέτουν το ελάχιστο κοινό σύνολο. Κατ' αυτόν τον τρόπο ορίζεται τρόπος τινά στη δεξαμενή ενεργειών, μια «σύμβαση» (contract) που εγγυάται τη διαθεσιμότητα συγκεκριμένων μετα-πληροφοριών σε όλες τις εναλλακτικές. Έτσι καθίσταται δυνατή η σύγκρισή τους επί τη βάσει κοινών

κριτηρίων, τα οποία επιλέγονται από τα μετα-δεδομένα της δεξαμενής ενεργειών. Σημειώνουμε ότι ο έλεγχος της ύπαρξης ενός μετα-δεδομένου γίνεται με βάση το όνομα και τον τύπο του. Τα μετα-δεδομένα των εναλλακτικών θα πρέπει να περιέχουν τιμές που περιγράφουν (ή άλλως χαρακτηρίζουν) τις αντίστοιχες εναλλακτικές. Για παράδειγμα “45Kg” ή “8gr” για το μετα-δεδομένο «βάρος», “2m” ή “3.5cm” για το «μήκος» μιας συγκεκριμένης εναλλακτικής, και ούτω καθεξής. Αντιθέτως, τα μετα-δεδομένα των δεξαμενών ενεργειών δε χρειάζεται να έχουν τιμές αφού αυτά υπάρχουν μόνο για να ορίσουν τη «σύμβαση», δηλαδή το ελάχιστο κοινό σύνολο μετα-δεδομένων των εναλλακτικών. Προφανώς εναλλακτικές που δεν διαθέτουν όλα τα απαιτούμενα από τη «σύμβαση» μετα-δεδομένα δεν μπορούν να εισαχθούν στην αντίστοιχη δεξαμενή ενεργειών ή δεν λαμβάνονται υπόψη κατά την αναζήτηση και επιλογή.

9.3.3 Μέθοδοι Αναζήτησης και Επιλογής (Search and Selection Methods)

Μια μέθοδος αναζήτησης και επιλογής (search and selection method) είναι υπεύθυνη για τη σάρωση των εναλλακτικών μιας δεξαμενής ενεργειών και την εξέταση των ιδιοτήτων και των μετα-δεδομένων τους, προκειμένου να αποφασίσει ποια ή ποιες από αυτές ταιριάζουν καλύτερα στον κόμβο αφηρημένης ενέργειας που ζήτησε την αναζήτηση. Σε διαφορετικούς κόμβους αφηρημένων ενεργειών πιθανώς να απαιτηθεί η χρήση διαφορετικών μεθόδων αναζήτησης και επιλογής καθώς αυτοί μοντελοποιούν διαφορετικούς στόχους. Για το λόγο αυτό τα SANs ως μέθοδος μοντελοποίησης καθώς και ο αλγόριθμος εκτέλεσής τους δεν προσδιορίζουν κάποια συγκεκριμένη μέθοδο αναζήτησης και επιλογής. Αντιθέτως αυτή δίνεται ως παράμετρος σε κάθε κόμβο αφηρημένης ενέργειας ξεχωριστά.

Στην υλοποίηση του μηχανισμού SANs που προτείνουμε η μέθοδος αναζήτησης και επιλογής ενός κόμβου αφηρημένης ενέργειας δίνεται ως το πλήρες όνομα μιας κλάσης Java (package και class name) που υλοποιεί τη μέθοδο σύμφωνα με μια συγκεκριμένη προγραμματιστική διεπαφή (API). Όταν εκτελείται μια μέθοδος αναζήτησης και επιλογής τότε ανακτά ορισμένες πληροφορίες απαραίτητες για τη λειτουργία της. Οι πληροφορίες αυτές μπορεί να καθοριστούν είτε ως μετα-δεδομένα του κόμβου αφηρημένης ενέργειας που ζήτησε την αναζήτηση, είτε ως πληροφορίες του πλαισίου εκτέλεσης (context elements). Οι πληροφορίες αυτές αφορούν τα κριτήρια επιλογής και τα αντίστοιχα βάρη τους (αν πρόκειται για πολυκριτηριακή μέθοδο), το πλήθος και την ταξινόμηση των εναλλακτικών που τελικά επιλέγονται (result size and ordering) καθώς και άλλες λεπτομέρειες της μεθόδου. Σημειώνουμε ότι σε ορισμένες περιπτώσεις είναι επιθυμητό να λαμβάνονται οι N καλύτερες εναλλακτικές κι όχι μόνο μία. Έπειτα αυτές μπορεί να συνδυαστούν σύμφωνα με μια πολιτική επίλυσης. Όταν επιλέγονται περισσότερες από μία εναλλακτικές τότε μπορεί να καθοριστεί και μια ταξινόμησή τους (ordering).

9.3.4 Πολιτική Επίλυσης (Resolution Policy)

Μια πολιτική επίλυσης (resolution policy) απαιτείται όταν η μέθοδος αναζήτησης και επιλογής επιστρέφει περισσότερες από μία εναλλακτικές. Αυτή είναι επιφορτισμένη με την τελική απόφαση επιλογής εκείνης της εναλλακτικής (από αυτές που επέστρεψε η μέθοδος αναζήτησης και επιλογής) που θα χρησιμοποιηθεί τελικά για την αντικατάσταση του κόμβου αφηρημένης ενέργειας. Επιτρέπεται όμως να επιλεγούν και παραπάνω από μία,

οπότε στην περίπτωση αυτή η πολιτική επίλυσης προσδιορίζει και τον τρόπο που θα συνδυαστούν, δημιουργώντας έτσι έναν νέο υποστόχο ή σύνθετη ενέργεια. Όπως και στην περίπτωση της μεθόδου αναζήτησης και επιλογής, η πολιτική επίλυσης δεν καθορίζεται από τα SANs ή τον αλγόριθμο εκτέλεσής τους αλλά δίνεται ως παράμετρος του εκάστοτε κόμβου αφηρημένης ενέργειας. Οι διαθέσιμες πολιτικές επίλυσης παρέχονται από την εκάστοτε υλοποίηση του μηχανισμού των SANs.

9.3.5 Πλεονεκτήματα της προτεινόμενης επέκτασης

Πέραν από μια λύση στο πρόβλημα της εξαντλητικής μοντελοποίησης των SANs κατά το σχεδιασμό (που περιγράψαμε στην εισαγωγή του κεφαλαίου), ο δυναμικός προσδιορισμός της υλοποίησης των κόμβων αφηρημένων ενεργειών παρέχει και ορισμένα πρόσθετα πλεονεκτήματα.

- Κατ' αρχήν επιτρέπει την ευέλικτη και ξεκάθαρη μοντελοποίηση των διαδικασιών αφού οι λεπτομέρειες υλοποίησης των υποστόχων μπορούν να αποτυπωθούν ξεχωριστά από το κύριο μοντέλο της διαδικασίας και να επαναχρησιμοποιηθούν και σε άλλα SANs όταν αυτό είναι δυνατό. Έτσι βελτιώνεται η σαφήνεια του κυρίως μοντέλου ενώ οι εναλλακτικές υλοποιήσεις των διαφόρων αφηρημένων ενεργειών μπορεί να συγκροτήσουν «βιβλιοθήκες» και να επαναχρησιμοποιηθούν και σε άλλα SANs, συμβάλλοντας έτσι στην αποφυγή επαναλήψεων (redundancy).
- Κατά δεύτερον προσθέτει ένα επίπεδο αφαίρεσης επάνω από την κύρια διαδικασία μοντελοποίησης, «αποκρύπτοντας» έτσι τις λεπτομέρειες της αναζήτησης, επιλογής και εκτέλεσης των αναλύσεων των αφηρημένων ενεργειών.
- Τέλος, οι εναλλακτικές υλοποιήσεις ομοειδών αφηρημένων ενεργειών μπορεί να σχεδιαστούν και αναπτυχθούν (εν είδει δεξαμενών ενεργειών) από τρίτους ή ακόμη και συνεργατικά από μια ομάδα σχεδιαστών.

9.4 Μέθοδοι Αναζήτησης & Επιλογής και Πολιτικές Επίλυσης

Όπως έχει προαναφερθεί οι μέθοδοι αναζήτησης και επιλογής χρησιμοποιούνται για την εξέταση των εναλλακτικών των δεξαμενών ενεργειών προκειμένου να αναγνωρίσουν εκείνες που ικανοποιούν τα εκάστοτε κριτήρια αναζήτησης. Ξεχωριστές υλοποιήσεις του μηχανισμού των SANs μπορεί να παρέχουν διαφορετικές μεθόδους αναζήτησης και επιλογής. Η δική μας υλοποίηση παρέχει δύο μεθόδους που ονομάζονται SMART και LOWA. Πρόκειται για πολυκριτηριακές μεθόδους λήψης αποφάσεων, οι οποίες περιγράφονται αναλυτικά στη βιβλιογραφία. Η πρώτη χρησιμοποιεί αριθμητικές τιμές (που λαμβάνει από το πλαίσιο εκτέλεσης και τα μετα-δεδομένα) προκειμένου να επιλέξει και ταξινομήσει τις εναλλακτικές, ενώ η δεύτερη χρησιμοποιεί γλωσσικούς όρους για να πετύχει το ίδιο. Στη συνέχεια παρουσιάζεται ο τρόπος υλοποίησής τους και περιγράφονται συνοπτικά οι ροές εργασιών τους. Τέλος, στην υλοποίηση του μηχανισμού των SANs ορίσαμε πέντε πολιτικές επίλυσης.

9.4.1 Αναζήτηση και επιλογή με τη μέθοδο SMART

Η βασισμένη στη μέθοδο SMART αναζήτηση και επιλογή εναλλακτικών λαμβάνει τις ρυθμίσεις και τις παραμέτρους που χρειάζεται από τον (εκάστοτε) κόμβο αφηρημένης ενέργειας. Οι παράμετροι αυτές έχουν τη μορφή μετα-δεδομένων του κόμβου ή είναι πληροφορίες του τοπικού πλαισίου εκτέλεσης. Κατά την κλήση της μεθόδου αναζήτησης και επιλογής παρέχονται ως είσοδοι οι εξής πληροφορίες:

- Αναφορά προς τον κόμβο αφηρημένης ενέργειας που καλεί τη μέθοδο.
- Αναφορά προς τη δεξαμενή ενεργειών που θα χρησιμοποιηθεί.
- Αναφορά στο τοπικό πλαίσιο εκτέλεσης (local context).

Με τις παραπάνω παραμέτρους η μέθοδος εκτελεί τα ακόλουθα βήματα.

1. Ανακτά όλες τις εναλλακτικές από τη δεξαμενή ενεργειών.
2. Εξάγει τα ονόματα των κριτηρίων επιλογής από το μετα-δεδομένο (παράμετρο) "SMART CRITERIA" του κόμβου αφηρημένης ενέργειας.
3. Εξάγει τα βάρη των κριτηρίων από τα μετα-δεδομένα του κόμβου αφηρημένης ενέργειας που αναφέρονται στην παράμετρο "SMART CRITERIA".
4. Για κάθε εναλλακτική γίνονται τα εξής:
 - α. Εξάγει τις αριθμητικές τιμές για κάθε κριτήριο από τα μετα-δεδομένα.
 - β. Υπολογίζει και αποθηκεύει τη συνολική βαθμολογία της εναλλακτικής με χρήση της μεθόδου SMART. Η SMART χρησιμοποιεί τα βάρη κριτηρίων (από τον κόμβο αφηρημένης ενέργειας) και τις τιμές (από την εναλλακτική) προκειμένου να υπολογίσει τη συνολική βαθμολογία.
5. Εξάγει την κατεύθυνση της ταξινόμησης από την παράμετρο "Ascending Order" του κόμβου αφηρημένης ενέργειας.
6. Οι εναλλακτικές ταξινομούνται σε αύξουσα ή φθίνουσα σειρά, με βάση τη συνολική τους βαθμολογία και την κατεύθυνση που έχει καθοριστεί.
7. Αν έχει οριστεί η παράμετρος "Results Count" στον κόμβο αφηρημένης ενέργειας (ως μετα-δεδομένο), τότε χρησιμοποιείται στον υπολογισμό του πόσες και ποιες εναλλακτικές θα συμπεριληφθούν στη λίστα αποτελεσμάτων. Για παράδειγμα TOP 3, BOTTOM 1, TOP 20%.
8. Η λίστα αποτελεσμάτων επιστρέφεται στον κόμβο αφηρημένης ενέργειας.

9.4.2 Αναζήτηση και επιλογή με τη μέθοδο LOWA

Η βασισμένη στη μέθοδο LOWA αναζήτηση και επιλογή εναλλακτικών, λαμβάνει τις ρυθμίσεις και παραμέτρους που χρειάζεται από τον (εκάστοτε) κόμβο αφηρημένης ενέργειας. Οι παράμετροι αυτές έχουν τη μορφή μετα-δεδομένων του κόμβου ή είναι πληροφορίες του τοπικού πλαισίου εκτέλεσης. Κατά την κλήση της μεθόδου αναζήτησης και επιλογής παρέχονται ως είσοδοι οι εξής πληροφορίες:

- Αναφορά προς τον κόμβο αφηρημένης ενέργειας που καλεί τη μέθοδο.
- Αναφορά προς τη δεξαμενή ενεργειών που θα χρησιμοποιηθεί.
- Αναφορά στο τοπικό πλαίσιο εκτέλεσης (local context).

Με τις παραπάνω παραμέτρους η μέθοδος εκτελεί τα ακόλουθα βήματα.

1. Ανακτά όλες τις εναλλακτικές από τη δεξαμενή ενεργειών.
2. Εξάγει τα ονόματα των κριτηρίων επιλογής από το μετα-δεδομένο (παράμετρο) "LOWA CRITERIA" του κόμβου αφηρημένης ενέργειας.
3. Εξάγει τους επιτρεπτούς γλωσσικούς όρους από την παράμετρο (μετα-δεδομένο) "Allowed Values" του κόμβου αφηρημένης ενέργειας.
4. Εξάγει (αν καθορίζονται) από την παράμετρο "Mapping" του κόμβου αφηρημένης ενέργειας, τις αντιστοιχίσεις μεταξύ των τιμών των κριτηρίων των εναλλακτικών και των επιτρεπτών γλωσσικών όρων.
5. Εξάγει τα βάρη των κριτηρίων από τα μετα-δεδομένα του κόμβου αφηρημένης ενέργειας που αναφέρονται στην παράμετρο "LOWA CRITERIA".
6. Για κάθε εναλλακτική γίνονται τα εξής:
 - α. Εξάγει τις τιμές για κάθε κριτήριο από τα μετα-δεδομένα της εναλλακτικής. Οι τιμές μπορεί να είναι είτε αριθμητικές είτε γλωσσικές (string).
 - β. Μετατροπή των αριθμητικών τιμών των κριτηρίων σε γλωσσικούς όρους χρησιμοποιώντας την αντιστοίχιση (αν καθορίζεται κάποια).
 - γ. Υπολογίζει και αποθηκεύει τη συνολική βαθμολογία της εναλλακτικής με χρήση της μεθόδου LOWA. Η LOWA χρησιμοποιεί τα βάρη κριτηρίων (από τον κόμβο αφηρημένης ενέργειας) και τους γλωσσικούς όρους (από την εναλλακτική και την αντιστοίχιση) προκειμένου να υπολογίσει τη συνολική γλωσσική βαθμολογία.
7. Αν έχει οριστεί κάποιο φίλτρο στην παράμετρο "Results Filter" του κόμβου αφηρημένης ενέργειας, χρησιμοποιείται για να αφαιρέσει τις εναλλακτικές με συνολικές γλωσσικές βαθμολογίες τις οποίες απορρίπτει το φίλτρο.
8. Εξάγει την κατεύθυνση της ταξινόμησης από την παράμετρο "Ascending Order" του κόμβου αφηρημένης ενέργειας.
9. Οι εναλλακτικές ταξινομούνται σε αύξουσα ή φθίνουσα σειρά, με βάση τη συνολική τους βαθμολογία και την κατεύθυνση που έχει καθοριστεί.
10. Αν έχει οριστεί η παράμετρος "Results Count" στον κόμβο αφηρημένης ενέργειας, τότε χρησιμοποιείται στον υπολογισμό του πόσες και ποιες εναλλακτικές θα συμπεριληφθούν στη λίστα αποτελεσμάτων.
11. Η λίστα αποτελεσμάτων επιστρέφεται στον κόμβο αφηρημένης ενέργειας.

9.4.3 Πολιτικές επίλυσης που υλοποιήθηκαν

Η υλοποίηση των SANs που έχουμε αναπτύξει παρέχει τις εξής πέντε πολιτικές επίλυσης.

- Χρήση της εναλλακτικής που βρίσκεται στην πρώτη θέση της ταξινομημένης λίστας αποτελεσμάτων («καλύτερη εναλλακτική»).
- Επιλογή μιας τυχαίας εναλλακτικής από τη λίστα αποτελεσμάτων.
- Συνδυασμός όλων των εναλλακτικών της λίστας αποτελεσμάτων σε έναν κόμβο παράλληλης εκτέλεσης τύπου ANY.
- Συνδυασμός όλων των εναλλακτικών της λίστας αποτελεσμάτων σε έναν κόμβο παράλληλης εκτέλεσης τύπου ALL.
- Συνδυασμός όλων των εναλλακτικών της λίστας αποτελεσμάτων σε έναν κόμβο παράλληλης εκτέλεσης με χρονικό περιορισμό.

9.5 Υλοποίηση – Επέκταση της πλατφόρμας των SANs

Η λειτουργικότητα της δυναμικής εξέλιξης των SANs που περιγράφηκε στο παρόν κεφάλαιο ενσωματώθηκε στην υλοποίηση του μηχανισμού των SANs, επεκτείνοντας τη βασική προγραμματιστική διεπαφή του (package org.iccs.san.api). Επίσης σε όποια υποσυστήματά του ήταν απαραίτητο δοθήκαν νέες υλοποιήσεις, όπως στον αλγόριθμο διάσχισης (σύστημα εκτέλεσης) και στο αποθετήριο των SANs. Όλες οι επεκτάσεις έγιναν με τη γλώσσα προγραμματισμού Java, η οποία χρησιμοποιήθηκε και για την αρχική υλοποίηση του μηχανισμού των SANs.

Η βασισμένη στη SMART μέθοδος αναζήτησης και επιλογής, υλοποιήθηκε ως μια κλάση Java, την `org.iccs.san.util.mcdm.MCDMSearchMethod`. Παρόμοια, η βασισμένη στη μέθοδο LOWA, μέθοδος αναζήτησης και επιλογής, υλοποιήθηκε με την κλάση `org.iccs.san.util.mcdm.LowaMCDMSearchMethod`.

9.5.1 Παραμετροποίηση της αναζήτησης και επιλογής του μηχανισμού των SANs

Στη συνέχεια παρουσιάζουμε τον τρόπο ρύθμισης των κόμβων αφηρημένης ενέργειας και των δεξαμενών ενεργειών προκειμένου να μπορέσουν να χρησιμοποιηθούν από τις δύο μεθόδους αναζήτησης και επιλογής που παρέχει η υλοποίηση του μηχανισμού των SANs. Τονίζουμε ότι τα παρακάτω αφορούν τη συγκεκριμένη υλοποίηση και δεν αποτελούν απαίτηση των SANs ή της επέκτασής τους για δυναμική εξέλιξη.

Οι δύο μέθοδοι αναζήτησης και επιλογής που παρέχονται χρησιμοποιούν παρόμοιες παραμέτρους και ρυθμίσεις. Συνεπώς απαιτούνται τα ίδια βήματα για τη ρύθμιση και τη χρήση τους. Ειδικότερα, και οι δύο απαιτούν τον ορισμό συγκεκριμένων παραμέτρων ως μετα-δεδομένα, στους κόμβους αφηρημένων ενεργειών, στις δεξαμενές ενεργειών καθώς και στις εναλλακτικές που περιέχουν.

Στους κόμβους αφηρημένων ενεργειών προστίθενται μετα-δεδομένα που προσδιορίζουν ποια θα είναι τα κριτήρια επιλογής καθώς και τα βάρη τους. Επίσης καθορίζουν τον τρόπο ταξινόμησης των εναλλακτικών, το πλήθος των αποτελεσμάτων καθώς και κάποιο φίλτρο.

- "SMART CRITERIA" ή "LOWA CRITERIA". Η παράμετρος αυτή περιέχει μια λίστα με τα ονόματα των μετα-δεδομένων τα οποία θα χρησιμοποιηθούν ως κριτήρια επιλογής. Μετα-δεδομένα που δεν περιέχονται σε αυτή τη λίστα και δεν έχουν κάποια άλλη ειδική σημασία (δείτε στη συνέχεια) απλά θα αγνοούνται από τη μέθοδο αναζήτησης και επιλογής.
- Όλα τα κριτήρια που αναφέρονται στην παράμετρο "SMART/LOWA CRITERIA" πρέπει να υπάρχουν τόσο στον κόμβο αφηρημένης ενέργειας όσο και στη δεξαμενή ενεργειών (δηλαδή να ανήκουν στην «σύμβαση» της δεξαμενής ενεργειών). Είναι δυνατό σε διαφορετικούς κόμβους αφηρημένων ενεργειών να χρησιμοποιούνται διαφορετικά κριτήρια και βάρη, ακόμη κι αν χρησιμοποιούν την ίδια δεξαμενή ενεργειών.
- Τα μετα-δεδομένα των κριτηρίων στους κόμβους αφηρημένων ενεργειών, πρέπει να είναι αριθμητικού τύπου (numeric) και οι τιμές τους πρέπει να περιέχουν τα αντίστοιχα βάρη που θα κυμαίνονται μεταξύ 0 και 1. Το δε άθροισμα των βαρών όλων των κριτηρίων πρέπει να ισούται ακριβώς με 1.
- "Ascending Order". Η παράμετρος αυτή καθορίζει αν οι εναλλακτικές (που θα περιληφθούν) στα αποτελέσματα της αναζήτησης, θα είναι ταξινομημένες κατ' αύξουσα ή φθίνουσα βαθμολογία. Εναλλακτικές με την ίδια βαθμολογία θα τοποθετούνται με τυχαία σειρά μεταξύ τους αλλά σίγουρα πριν από τις εναλλακτικές με υψηλότερες βαθμολογίες και μετά από τις εναλλακτικές με μικρότερες βαθμολογίες. Η παράμετρος αυτή δέχεται τις λογικές (Boolean) τιμές true και false. Αν παραλειφθεί τότε θεωρείται false και γίνεται φθίνουσα ταξινόμηση.
- "Results Count". Η παράμετρος αυτή καθορίζει το πλήθος των εναλλακτικών που επιλέγονται και επιστρέφονται ως αποτέλεσμα της αναζήτησης. Οι αποδεκτές τιμές έχουν την ακόλουθη μορφή:

<TOP or BOTTOM> <number or percentage>

Για παράδειγμα:

TOP 3 ή TOP 10% ή BOTTOM 4 ή BOTTOM 10%

Η μέθοδος που βασίζεται στη LOWA χρησιμοποιεί τρεις επιπλέον παραμέτρους.

- "Allowed Values". Η παράμετρος αυτή καθορίζει ποιοι θα είναι οι αποδεκτοί γλωσσικοί όροι (linguistic terms) που θα χρησιμοποιηθούν ως τιμές των κριτηρίων στις εναλλακτικές της δεξαμενής ενεργειών καθώς και στις επόμενες παραμέτρους. Οι γλωσσικοί όροι πρέπει να δοθούν με αύξουσα διάταξη, για παράδειγμα «Πολύ λίγο», «Λίγο», «Κανονικό», «Πολύ» και «Πάρα πολύ».
- "Results Filter". Η παράμετρος αυτή μπορεί να χρησιμοποιηθεί για να διατηρήσει ή να απορρίψει εναλλακτικές από τα αποτελέσματα, με βάση τη συνολική βαθμολογία τους που θα υπολογιστεί με τη μέθοδο LOWA και η οποία επίσης είναι ένας γλωσσικός όρος. Αν για παράδειγμα η τιμή της παραμέτρου είναι

“MEDIUM, HIGH” τότε θα διατηρηθούν στα αποτελέσματα μόνο εκείνες οι εναλλακτικές με βαθμολογία LOWA ίση με MEDIUM ή HIGH. Αν στην αρχή της τιμής της παραμέτρου τεθεί ο τελεστής **NOT**, τότε η ενέργεια του φίλτρου αντιστρέφεται. Για παράδειγμα αν η τιμή της παραμέτρου είναι “NOT BAD” τότε θα απορριφθούν όλες τις εναλλακτικές με βαθμολογία BAD.

- ο “Mapping”. Η παράμετρος αυτή καθορίζει πως οι τιμές των μετα-δεδομένων των κριτηρίων των εναλλακτικών «μεταφράζονται» σε γλωσσικούς όρους. Η μετάφραση αυτή λαμβάνει χώρα πριν εφαρμοστεί η LOWA για να υπολογίσει τις συνολικές βαθμολογίες των εναλλακτικών. Προφανώς αν οι τιμές των μετα-δεδομένων των εναλλακτικών είναι ήδη γλωσσικοί όροι (απ’ το ίδιο σύνολο με αυτό του κόμβου αφηρημένης ενέργειας) τότε δε χρειάζεται μετάφραση.

Ο τύπος της παραμέτρου είναι αλφαριθμητικό πολλών γραμμών (multi-line string parameter). Σε κάθε γραμμή δίνονται οι αντιστοιχίσεις (ως λίστα) για ένα μόνο κριτήριο, χρησιμοποιώντας το ακόλουθο συντακτικό (σε BNF) :

- Αν το κριτήριο είναι αλφαριθμητικού τύπου (string):

```
<parameter> ::= <str mapping line> (<NL> <str mapping line>)*
<str mapping line> ::= <criterion name> "=" <str_map_list>
<str_map_list> ::= <value_map> ("," <value_map>)*
<value_map> ::= (<actual str val> | "*" ) ":" <linguistic term>
<actual str val> ::= <string>
<linguistic term> ::= <string>
<criterion name> ::= <string>
<NL> ::= "\n" -- new line character or sequence
```

Ο αστερίσκος (*) μπορεί να χρησιμοποιηθεί για να υποδηλώσει οποιαδήποτε άλλη τιμή (string value) που δεν αναφέρεται στη λίστα των αντιστοιχίσεων.

- Αν το κριτήριο είναι αριθμητικού τύπου τότε πρέπει να αντιστοιχισθεί ο κάθε γλωσσικός όρος με συγκεκριμένα εύρη τιμών. Αυτό γίνεται ορίζοντας μια λίστα με ζεύγη από εύρη τιμών και με τους αντίστοιχους γλωσσικούς όρους. Ένας γλωσσικός όρος μπορεί να εμφανίζεται περισσότερες από μία φορές στην ίδια λίστα, για διαφορετικά εύρη τιμών. Ακολούθως δίνονται μερικά παραδείγματα για το σύνολο BAD, OK, GOOD, PERFECT:

```
(-inf..0]:BAD, (0..5]:OK, (5..12]:GOOD, (12,+inf):PERFECT
```

Ένα εύρος τιμών ορίζεται από το κάτω και το άνω όριο του, διαχωρισμένα με δύο τελείες “. .”. Επίσης περικλείεται σε παρενθέσεις () ή σε αγκύλες [] ή σε οποιονδήποτε συνδυασμό αυτών που υποδηλώνουν ότι η αντίστοιχη οριακή τιμή περιλαμβάνεται στο εύρος (με χρήση αγκυλών) ή δεν περιλαμβάνεται (με χρήση παρενθέσεων). Ο όρος “inf” υποδηλώνει το «άπειρο».

Η αντιστοίχιση μιας συγκεκριμένης τιμής σε ένα γλωσσικό όρο είναι δυνατή περικλείοντάς την σε αγκύλες []. Επίσης μπορεί να χρησιμοποιηθεί ο αστερίσκος (*) για να υποδηλώσει όλες τις τιμές που δεν ανήκουν σε κάποιο από τα εύρη τιμών (για το συγκεκριμένο κριτήριο). Η μορφή των αντιστοιχίσεων των αριθμητικών κριτηρίων σε BNF είναι η εξής:

```
<parameter> ::= <num mapping line> (<NL> <num mapping line>)*
<num mapping line> ::= <criterion name> "=" <range_map_list>
<range_map_list> ::= <range_map> ("," <range_map>)*
<range_map> ::= <range_spec> ":" <linguistic term>
<range_spec> ::= <range> | <single_value> | "*"
<range> ::= "(" | "[" <number> ".." <number> ")" | "]"
<single_value> ::= "[" <number> "]"
<linguistic term> ::= <string>
<criterion name> ::= <string>
<NL> ::= "\n" -- new line character or sequence
```

Ακολούθως δίνεται ένα πλήρες παράδειγμα αντιστοιχίσεων, για την παράμετρο Mapping:

```
Manufacturer = M1:OK, M2:GOOD, M3:BAD, *:BAD
Owner         = O1:GOOD, O2:GOOD, O3:OK, *:BAD
Throughput    = [0..10]:BAD, (10..20):OK, (10..inf):GOOD
Age           = [0..2]:GOOD, (2..5):OK, (5.. inf):BAD
Accuracy      = [0..0.1]:GOOD, (0.1..0.5):OK, *:UNACCEPTABLE
```

Οι δεξαμενές ενεργειών πρέπει να έχουν τις ακόλουθες παραμέτρους (ως μετα-δεδομένα).

- *Μοναδικό αναγνωριστικό.* Πρόκειται για ένα μοναδικό URI το οποίο μπορεί να χρησιμοποιηθεί για αναφορά σε μια συγκεκριμένη δεξαμενή ενεργειών.
- *Παράμετροι Κριτηρίων.* Κάθε μετα-δεδομένο που προστίθεται σε μια δεξαμενή ενεργειών ορίζει (ή υποδηλώνει) ότι είναι «υποχρεωτικό» όλες οι εναλλακτικές που περιέχονται σε αυτή να έχουν ένα αντίστοιχο μετα-δεδομένο, δηλαδή ένα μετα-δεδομένο με το ίδιο όνομα και τύπο. Με άλλα λόγια όλες οι εναλλακτικές θα πρέπει να έχουν τουλάχιστο τα ίδια μετα-δεδομένα με τη δεξαμενή ενεργειών που τις περιέχει. Κατ' αυτόν τον τρόπο τα μετα-δεδομένα μιας δεξαμενής ενεργειών ορίζουν ουσιαστικά μια «σύμβαση μετα-δεδομένων» για τη δεξαμενή.

Μερικά από τα μετα-δεδομένα ή το σύνολο αυτών, μπορούν να χρησιμοποιηθούν ως κριτήρια επιλογής από τις δύο πολυκριτηριακές μεθόδους αναζήτησης και επιλογής που παρουσιάστηκαν παραπάνω. Φυσικά τα ίδια μετα-δεδομένα μπορεί να χρησιμοποιηθούν και από οποιαδήποτε άλλη μέθοδο, πολυκριτηριακή ή μη, προστεθεί στο μέλλον. Εναλλακτικές που δεν έχουν όλα τα απαιτούμενα μετα-δεδομένα (αυτά που χρησιμοποιούνται ως κριτήρια επιλογής) δεν εξετάζονται κατά την αναζήτηση και επιλογή των εναλλακτικών και συνεπώς δεν θα συμπεριληφθούν

στα αποτελέσματα των αναζητήσεων. Αντιθέτως, επιπλέον μετα-δεδομένα (πέραν αυτών που απαιτούνται από τη δεξαμενή ενεργειών) επιτρέπονται αλλά δεν λαμβάνονται υπόψη.

Όλες οι εναλλακτικές ενέργειες θα πρέπει να έχουν τα εξής μετα-δεδομένα.

- *Μοναδικό αναγνωριστικό.* Πρόκειται για ένα μοναδικό URI που μπορεί να χρησιμοποιηθεί για αναφορά σε μια συγκεκριμένη εναλλακτική.
- *Τιμές Κριτηρίων (Values).* Οι τιμές των μετα-δεδομένων πρέπει να αντιστοιχούν σε αυτά που έχουν οριστεί στη δεξαμενή ενεργειών. Η αντιστοίχιση γίνεται με βάση το όνομα και τον τύπο τους. Οι τιμές των μετα-δεδομένων χρησιμοποιούνται για τον υπολογισμό των μέσων όρων (συνολικών βαθμολογιών) των εναλλακτικών με χρήση των τεχνικών SMART ή LOWA. Στην περίπτωση της LOWA οι τιμές μπορεί να είναι είτε γλωσσικοί όροι (αυτοί που επιτρέπει η παράμετρος "Allowed Values" του κόμβου αφηρημένης ενέργειας), είτε αριθμητικές ή αλφαριθμητικές, οπότε και θα πρέπει να οριστεί η παράμετρος Mapping στον κόμβο αφηρημένης ενέργειας για να προσδιορίσει πως θα αντιστοιχισθούν οι τιμές των μετα-δεδομένων των εναλλακτικών με τις επιτρεπτές γλωσσικές τιμές (γλωσσικούς όρους) του κόμβου.

Ένα πλήρες παράδειγμα κόμβου αφηρημένης ενέργειας με όλες του τις ρυθμίσεις καθώς και την αντίστοιχη δεξαμενή ενεργειών, δίνεται στο «ΠΑΡΑΡΤΗΜΑ II – Παράδειγμα κόμβου αφηρημένης ενέργειας & δεξαμενής ενεργειών».

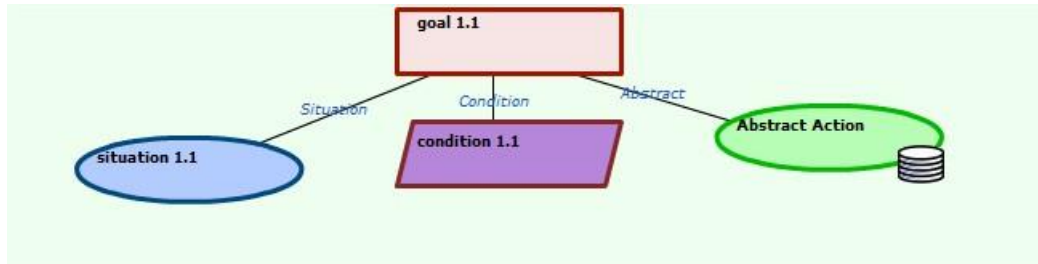
9.6 Επεξεργασία SANs με κόμβους αφηρημένων ενεργειών

Εκτός από τον μηχανισμό εκτέλεσης των SANs επεκτάθηκε και ο επεξεργαστής SANs προκειμένου να επιτρέπει την προσθήκη, επεξεργασία και παραμετροποίηση των κόμβων αφηρημένων ενεργειών και των δεξαμενών ενεργειών. Επίσης προστέθηκε η δυνατότητα προσθήκης μετα-δεδομένων στους κόμβους των SANs, αλλά και στις οντότητες (entity metadata) καθώς και μετα-δεδομένα κοινά για ολόκληρη την εφαρμογή (global metadata).



Σχήμα 9-1. SAN με κόμβο αφηρημένης ενέργειας και οι ιδιότητες αυτού

Χρησιμοποιώντας τις νέες δυνατότητες του επεξεργαστή SANs ο σχεδιαστής μπορεί να προσθέσει κόμβους αφηρημένων ενεργειών και να τους συσχετίσει με δεξαμενές ενεργειών, οι οποίες περιέχουν εναλλακτικές ενέργειες ή υποστόχους που μπορούν να αντικαταστήσουν τους κόμβους αυτούς (Σχήμα 9-1). Όταν προστεθεί ένας κόμβος αφηρημένης ενέργειας τότε εμφανίζονται στο δεξί μέρος του επεξεργαστή SANs, ρυθμίσεις που αφορούν τη δεξαμενή ενεργειών, τη μέθοδο αναζήτησης και επιλογής και την πολιτική επίλυσης. Στην παρούσα υλοποίηση του επεξεργαστή παρέχονται οι δύο πολυκριτηριακές μέθοδοι αναζήτησης και επιλογής που παρουσιάστηκαν προηγουμένως (LOWA και SMART) καθώς και οι πέντε πολιτικές επίλυσης.



Σχήμα 9-2. SAN με κόμβο αφηρημένης ενέργειας

Στο Σχήμα 9-2 δίνεται η γραφική αναπαράσταση ενός SAN που περιλαμβάνει έναν κόμβο αφηρημένης ενέργειας, ενώ στο Σχήμα 9-3 απεικονίζεται ο επεξεργαστής SANs στο σημείο όπου ο σχεδιαστής μπορεί να ρυθμίσει την πολιτική επίλυσης. Οι πέντε επιλογές που παρέχονται παρουσιάστηκαν σε προηγούμενη ενότητα (9.4.3).

Parameter	Value
Id	_Abstract_Action
Name	Abstract Action
Type	Abstract
Action Pool	Action Pool Demo
Sel. Method Configuration	
Resolution Policy	Select
Selection Method	Select
	ONE FIRST (Executes the first)
	ONE RANDOM (Executes a random)
	PARALLEL ANY (Executes all in parallel)
	PARALLEL ALL (Executes all in parallel)

Σχήμα 9-3. Ρυθμίσεις κόμβου αφηρημένης ενέργειας

Με διπλό κλικ σε έναν κόμβο αφηρημένης ενέργειας εμφανίζεται ένα πλαίσιο διαλόγου (Σχήμα 9-4) όπου ο σχεδιαστής μπορεί να ορίσει τα κριτήρια που θα χρησιμοποιηθούν για τη βαθμολόγηση και επιλογή των εναλλακτικών, όταν εκτελεστεί ο κόμβος αφηρημένης ενέργειας. Στο πλαίσιο διαλόγου μπορούν να οριστούν τα ονόματα, τα βάρη και οι τύποι των κριτηρίων. Επίσης για τα αριθμητικά κριτήρια καθορίζεται και η «κατεύθυνση» του κριτηρίου, δηλαδή το αν για μεγαλύτερες τιμές του κριτηρίου βελτιώνεται η βαθμολογία της αντίστοιχης εναλλακτικής ή αντίστροφα.

Edit Action

ID (URI)

Name

Type **Abstract**

Action Pool

Selection Method **LOWA** Resolution Policy **ONE FIRST (Executes the first job in the results list)**

Selection Method Conf. Very Low, Low, Ok, High, Very High

Criteria

Name	Weight	Type	Direction
1st Criterion	0.4	STRING	N/A
2nd Criterion	0.5	NUMBER	INC

Name Weight Type **STRING** Direction **N/A**

Add **Delete** **Edit Mappings** **Normalize** Sum=0.9

Σχήμα 9-4. Εισαγωγή κριτηρίων σε κόμβο αφηρημένης ενέργειας

Στο Σχήμα 9-5 φαίνεται η δυνατότητα αυτόματης κανονικοποίησης των βαρών των κριτηρίων (κουμπί Normalize). Αυτό αυξάνει ή ελαττώνει τα βάρη των κριτηρίων αναλογικά (χωρίς να αλλοιωθούν οι μεταξύ τους σχέσεις) έτσι ώστε να έχουν άθροισμα 1. Επίσης, αν χρησιμοποιηθεί η μέθοδος αναζήτησης και επιλογής με LOWA τότε θα πρέπει να οριστούν και οι αντιστοιχίσεις των τιμών των εναλλακτικών για κάθε κριτήριο με τους γλωσσικούς όρους (Σχήμα 9-6).

Name	Weight	Type	Direction
1st Criterion	0.4444444444444445	STRING	N/A
2nd Criterion	0.5555555555555556	NUMBER	INC

Name Weight Type **STRING** Direction **N/A**

Add **Delete** **Edit Mappings** **Normalize** Sum=1

Normalize Weights

Σχήμα 9-5. Κανονικοποίηση των βαρών των κριτηρίων

Action Pool: Action Pool Demo

Selection Method: **LOWA** Resolution Policy: **ONE FIRST (Executes the first job in the results list)**

Selection Method Conf.:

Criteria:

Value	Mapping
0 - 2	Very Low
3 - 5	Low
6 - 8	Ok
9 - 10	High

Edit Value Mappings for '2nd Criterion' Value: 9 - 10 Mapping: High

Add **Delete** **Back**

Σχήμα 9-6. Αντιστοίχιση των τιμών αριθμητικού κριτηρίου σε γλωσσικούς όρους (LOWA)

Στο Σχήμα 9-7 απεικονίζεται ο επεξεργαστής SANs στο σημείο όπου μπορεί να γίνει ο ορισμός και η επεξεργασία των μετα-δεδομένων μιας δεξαμενής ενεργειών. Ορίζοντας μετα-δεδομένα σε μια δεξαμενή ενεργειών σημαίνει κατά σύμβαση ότι και όλες οι εναλλακτικές που περιέχει θα πρέπει να έχουν μετα-δεδομένα που να αντιστοιχούν με εκείνα της δεξαμενής ενεργειών και να παρέχουν συγκεκριμένες τιμές γι' αυτά.

Action Pool Editor

Metadata

Add meta Selected node: Action Pool Demo

ID	Name	Type	Units	Value	Expression	Range Min	Range Max	Range Step	Action
meta13599923...	Action Pool Dem	MCDM	-	1st Criterion, 2nd Criterion	-	-	-	-	Delete
meta135999247...	1st Criterion	String	-	-	-	-	-	-	Delete
meta135999248...	2nd Criterion	Number	Years	-	-	0	10	1	Delete

Σχήμα 9-7. Ορισμός μετα-δεδομένων σε δεξαμενή ενεργειών

Τελευταίο βήμα στη σχεδίαση ενός SAN που χρησιμοποιεί δεξαμενές ενεργειών είναι η προσθήκη τιμών σε όλες τις εναλλακτικές αυτών και για κάθε τους μετα-δεδομένο (κριτήριο). Στο παράδειγμα του σχήματος (Σχήμα 9-8), η δεξαμενή ενεργειών έχει δύο μετα-δεδομένα ως κριτήρια επιλογής (τα "1st Criterion" και "2nd Criterion"). Για την εναλλακτική "Candidate SAN 1" (που είναι επιλεγμένη αριστερά) ορίζονται οι τιμές για τα δύο αυτά κριτήρια. Το κριτήριο "1st Criterion" δέχεται γλωσσικές τιμές ("Low"). Όσον αφορά το δεύτερο κριτήριο, χρησιμοποιώντας την αντιστοίχιση στο Σχήμα 9-6 βλέπουμε ότι η τιμή του ("8") αντιστοιχεί στον γλωσσικό όρο "Ok".

The screenshot shows a software interface for defining criteria. On the left is a tree view with the following structure:

- Entity1-name
 - Context
 - Sans
 - san1-name
 - Action Pool Demo
 - Sans
 - Candidate SAN 1
 - Goal : goal candidate 1 (highlighted)
 - Candidate SAN 2

On the right, the 'Metadata' section is active, showing a table for 'Selected node: goal candidate 1'.

ID	Name	Type	Units	Value
meta1360172039178	1st Criterion	String		Low
meta1360172098031	2nd Criterion	Number	Years	8

Σχήμα 9-8. Ορισμός των τιμών των κριτηρίων

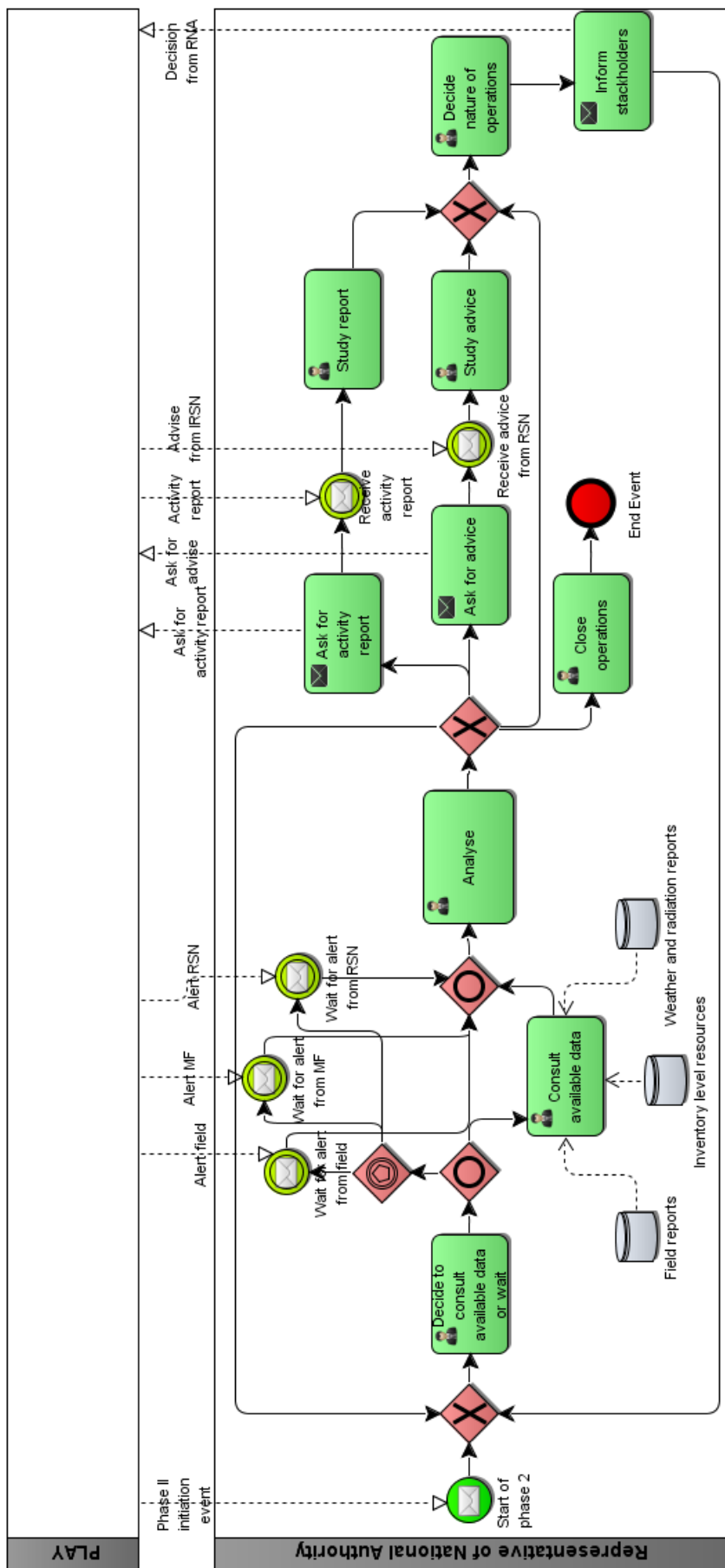
Κλείνοντας υπενθυμίσουμε ότι οι εναλλακτικές που περιέχονται στις δεξαμενές ενεργειών μπορεί να λαμβάνονται ή να ενημερώνονται αυτόματα από κάποια εξωτερική υπηρεσία. Η λήψη των εναλλακτικών της δεξαμενής ενεργειών γίνεται πριν τη χρήση της μεθόδου αναζήτησης και επιλογής.

10 Αξιολόγηση των Δυναμικά Εξελισσόμενων SANs

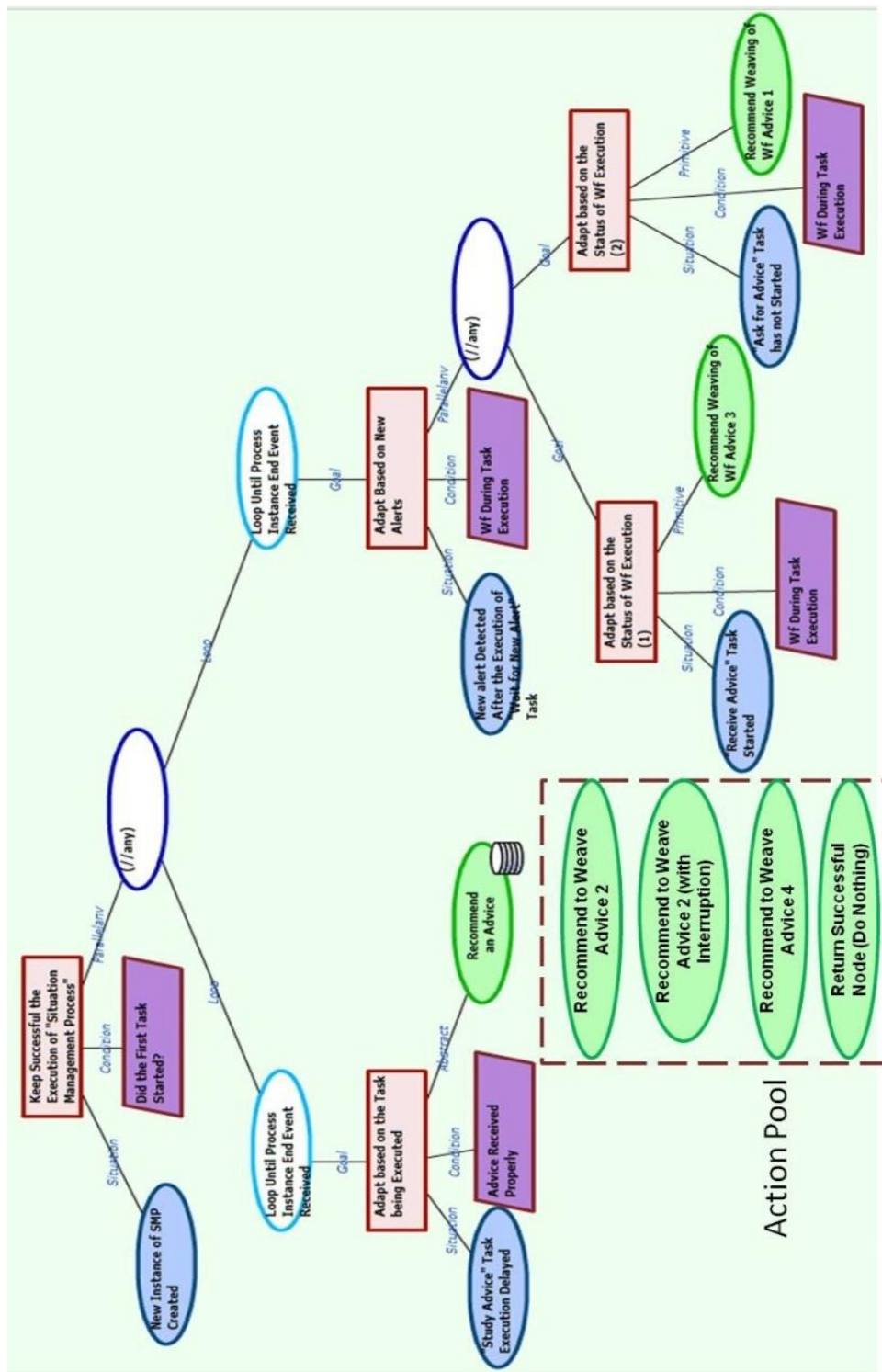
Στο κεφάλαιο αυτό επιχειρείται μια αξιολόγηση της επέκτασης των SANs με τη χρήση κόμβων αφηρημένων ενεργειών και δεξαμενών ενεργειών, μέσω ενός σεναρίου χρήσης. Το σενάριο αυτό είναι κατάλληλο για να καταδείξει την αποτελεσματικότητα της προτεινόμενης επέκτασης. Επίσης επιχειρείται μια ποιοτική σύγκριση των SANs με κόμβους αφηρημένων ενεργειών και SANs χωρίς τέτοιους, ως προς την εκφραστικότητα και την ευκολία μοντελοποίησης που παρέχει η κάθε προσέγγιση. Παρουσιάζεται ακόμη μια πειραματική σύγκριση των δύο προσεγγίσεων ως προς το μέσο χρόνο επεξεργασίας των γεγονότων και τη χρήση πόρων του συστήματος (threads). Περιγράφονται τα πειράματα που περιλαμβάνονται και παρουσιάζονται οι μετρήσεις και τα ευρήματα από την εκτέλεσή τους. Το κεφάλαιο κλείνει με σχολιασμό των αποτελεσμάτων.

10.1 Παράδειγμα χρήσης κόμβου αφηρημένων ενεργειών για προτάσεις προσαρμογών

Ακολούθως δίνεται μια παραλλαγή του παραδείγματος της ενότητας «8.5 Παράδειγμα χρήσης της πλατφόρμας ARP» για προσαρμογή της διαδικασίας διαχείρισης κρίσεων σε πυρηνικό σταθμό παραγωγής ηλεκτρικής ενέργειας (Σχήμα 10-1). Εδώ, αντί να μοντελοποιηθούν εκ των προτέρων όλες οι πιθανές προτάσεις προσαρμογών που αφορούν την περίπτωση της “aspect 2” γίνεται χρήση ενός κόμβου αφηρημένης ενέργειας και μιας αντίστοιχης δεξαμενής ενεργειών. Για την επιλογή της καταλληλότερης πρότασης προσαρμογής κατά την εκτέλεση, αξιοποιούνται πληροφορίες του πλαισίου. Για την αναζήτηση στη δεξαμενή ενεργειών χρησιμοποιείται η πολυκριτηριακή μέθοδος αναζήτησης και επιλογής που βασίζεται στην μέθοδο LOWA. Το παράδειγμα αυτό θα βοηθήσει στην καλύτερη και πληρέστερη κατανόηση των δυνατοτήτων που προσθέτει στα SANs η εισαγωγή των κόμβων αφηρημένων ενεργειών και των δεξαμενών ενεργειών και θα δείξει την ελάττωση της σχεδιαστικής προσπάθειας (modeling effort) που συνεπάγεται η χρήση τους.



Σχήμα 10-1. Διαδικασία χειρισμού κρίσεων



Σχήμα 10-2. SAN για προτάσεις προσαρμογών στη Διαδικασία χειρισμού κρίσεων (με χρήση δεξαμενής ενεργειών)

Το SAN όπως διαμορφώνεται στο παρόν παράδειγμα (Σχήμα 10-2) εκμεταλλεύεται τις ιδιότητες των κόμβων αφηρημένων ενεργειών και των δεξαμενών ενεργειών. Έτσι αποφεύγει τον εκ των προτέρων (δηλαδή κατά το σχεδιασμό) προσδιορισμό της πρότασης προσαρμογής που αφορά την “aspect 2”, δηλαδή την περίπτωση που η εργασία “Study advice” καθυστερεί να ολοκληρωθεί. Αντίθετα, χρησιμοποιείται ο κόμβος αφηρημένης κατάστασης “Recommend an advice”, ο οποίος δεν προσδιορίζει κάποια συγκεκριμένη πρόταση προσαρμογής (ενέργεια) αλλά παρέχει τα μέσα για την επιλογή αυτής όταν η εκτέλεση του SAN φθάσει σε αυτόν. Ο κόμβος αφηρημένης ενέργειας χρησιμοποιεί την δεξαμενή ενεργειών που επίσης φαίνεται στο Σχήμα 10-2 και η οποία περιέχει τέσσερις εναλλακτικές προτάσεις προσαρμογής για την περίπτωση “aspect 2”. Εδώ αξίζει να αναφέρουμε ότι τις εναλλακτικές αυτές θα μπορούσε να τις δημιουργεί δυναμικά ένα άλλο SAN ή και κάποιο τρίτο λογισμικό και να τις προσθέτει στη δεξαμενή ενεργειών.

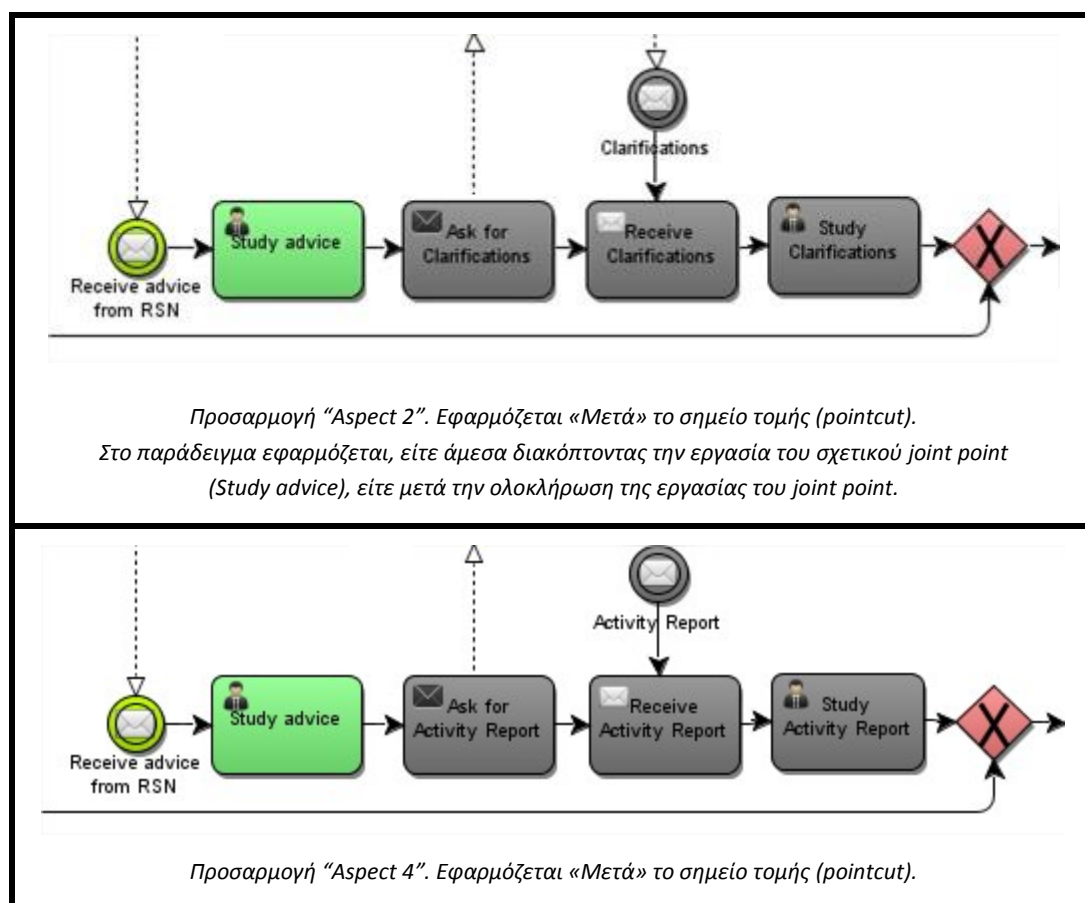
Στο παράδειγμα ως κριτήρια επιλογής των προτάσεων προσαρμογών χρησιμοποιούνται δύο πληροφορίες του πλαισίου, οι οποίες εξάγονται από γεγονότα που προέρχονται από το περιβάλλον. Αυτές είναι: (α) το επίπεδο ραδιενέργειας (Radiation Level) και (β) οι καιρικές συνθήκες (Weather Conditions). Η μέθοδος αναζήτησης και επιλογής με LOWA χρησιμοποιεί τις πληροφορίες αυτές για να επιλέγει την εκάστοτε καλύτερη πρόταση προσαρμογής από τις εναλλακτικές που παρέχει η δεξαμενή ενεργειών. Κάθε εναλλακτική διαθέτει μεταδεδομένα που περιγράφουν για ποιο επίπεδο ραδιενέργειας και ποιες καιρικές συνθήκες συνιστάται η χρήση της. Εκφράζονται δε ως γλωσσικοί όροι με τις εξής αποδεκτές τιμές: Κανονική κατάσταση (Normal), Χρίζει προσοχής (Notify), Ανησυχητική κατάσταση (Alert), Επικίνδυνη κατάσταση (Dangerous) και Κρίσιμη κατάσταση (Critical). Τα δύο κριτήρια αρχικοποιούνται με γλωσσικές τιμές που προέρχονται από το περιβάλλον και συγκεκριμένα από ροές γεγονότων με μετεωρολογικά δεδομένα και με μετρήσεις πεδίου. Η εναλλακτική της δεξαμενής ενεργειών, της οποίας τα μεταδεδομένα είναι πιο κοντά στις δύο αυτές τιμές, επιλέγεται και προτείνεται ως η καταλληλότερη. Προφανώς οι πιθανοί συνδυασμοί καταστάσεων του επιπέδου ραδιενέργειας και των καιρικών συνθηκών (25 τον αριθμό) είναι πολύ περισσότεροι από τους τέσσερις συνδυασμούς που περιγράφουν τις τέσσερις εναλλακτικές. Αυτό δεν αποτελεί πρόβλημα για τη μέθοδο αναζήτησης και επιλογής με LOWA αφού βρίσκει την εναλλακτική της οποίας τα μεταδεδομένα είναι πιο κοντά στον εκάστοτε συνδυασμό των τιμών των δύο κριτηρίων. Συνεπώς δε χρειάζεται να υπάρχει εναλλακτική για κάθε πιθανή περίπτωση. Αυτό ακριβώς καταδεικνύει την ευελιξία που παρέχει η χρήση των κόμβων αφηρημένων ενεργειών και των δεξαμενών ενεργειών. Τα δύο κριτήρια επιλογής δεν είναι απαραίτητως ίδιας σημαντικότητας. Ο σχεδιαστής του SAN μπορεί να ορίσει βάρη γι’ αυτά, για παράδειγμα 70% για το επίπεδο ραδιενέργειας (περισσότερο σημαντικό) και 30% για τις καιρικές συνθήκες (λιγότερο σημαντικές).

Υπενθυμίζουμε ότι ο κόμβος αφηρημένης ενέργειας εκτελείται όταν η εργασία “Study advice” καθυστερεί να ολοκληρωθεί (όπως και στο αρχικό παράδειγμα), ενώ υπάρχει ένδειξη για αυξημένη ραδιενέργεια (γι’ αυτό άλλωστε εκτελείται η διαδικασία διαχείρισης κρίσεων). Στα επόμενα, οι αναφορές στο θέμα “Aspect 2” αφορούν την προσαρμογή που περιγράφηκε στο αρχικό παράδειγμα (ενότητα 8.5) αλλά με ορισμένες τροποποιήσεις.

Η δεξαμενή ενεργειών του παραδείγματος περιέχει τις εναλλακτικές ενέργειες παραγωγής προτάσεων προσαρμογών που περιγράφονται ακολούθως.

1. Προσαρμογή “Aspect 2” χωρίς διακοπή. Μεταδεδομένα: Επίπεδο ραδιενέργειας: ALERT, Καιρικές συνθήκες: ALERT. Η προσαρμογή αυτή είναι ακριβώς ίδια με εκείνη του αρχικού παραδείγματος και αφορά την προσθήκη ορισμένων επιπλέον εργασιών μετά την εργασία “Study advice”. Αυτές είναι η αποστολή αιτήματος παροχής διευκρινήσεων (clarifications) επί της συμβουλής του Ινστιτούτου Ραδιοπροστασίας που επεξεργάστηκε στην “Study advice”. Μετά τη λήψη των διευκρινίσεων ακολουθεί η μελέτη τους στην εργασία “Study clarifications”. Η εναλλακτική αυτή προτείνεται (με βάση τα μεταδεδομένα της) για «λίγο επικίνδυνες» καταστάσεις, οπότε η προσαρμογή της διαδικασίας μπορεί να γίνει αφού ολοκληρωθεί η εργασία “Study advice” που καθυστερεί. Τότε μόνο ζητούνται οι διευκρινίσεις και όταν αυτές ληφθούν και ολοκληρωθεί η εργασία “Study clarifications” (που έπεται), η εκτέλεση της διαδικασίας επανέρχεται στην κανονική της ροή.
2. Προσαρμογή “Aspect 2” με διακοπή. Μεταδεδομένα: Επίπεδο ραδιενέργειας: CRITICAL, Καιρικές συνθήκες: DANGEROUS. Η εναλλακτική αφορά επικίνδυνες και κρίσιμες καταστάσεις οπότε προτείνεται η άμεση διακοπή της εργασίας “Study advice” και η αποστολή αιτήματος για διευκρινίσεις. Ουσιαστικά πρόκειται για την ίδια προσαρμογή με παραπάνω (aspect 2) αλλά με τη διαφορά ότι προτείνεται η άμεση εφαρμογή της.
3. Προσαρμογή “Aspect 4”. Μεταδεδομένα: Επίπεδο ραδιενέργειας: DANGEROUS, Καιρικές συνθήκες: NORMAL. Η προσαρμογή αυτή αφορά την αποστολή αιτήματος για αναλυτική αναφορά των ενεργειών που έχουν γίνει. Έτσι θα καταστεί δυνατή η ενδεδειγμένη μελέτη της τρέχουσας κατάστασης και ο σχεδιασμός της αντίδρασης στην κρίση. Με την παραλαβή της αναλυτικής αναφοράς ακολουθεί η μελέτη αυτής στην εργασία “Study report”. Μετά την ολοκλήρωση της μελέτης η εκτέλεση της διαδικασίας επανέρχεται στην κανονική της ροή. Η προσαρμογή αυτή αφορά επίσης επικίνδυνες καταστάσεις, με βάση το επίπεδο ραδιενέργειας, αλλά όπου όμως υπάρχει κάποιο χρονικό περιθώριο λόγω των ήπιων καιρικών συνθηκών. Έτσι δίνεται η δυνατότητα για πιο προσεκτική εξέταση των ενεργειών που έχουν ήδη γίνει καθώς και για τον σχεδιασμό των επόμενων. Η συγκεκριμένη προσαρμογή (aspect 4) προτείνεται για εκτέλεση μετά την ολοκλήρωση της εργασίας “Study advice” που καθυστερεί.
4. Κενή Προσαρμογή. Μεταδεδομένα: Επίπεδο ραδιενέργειας: NOTIFY, Καιρικές συνθήκες: NORMAL. Δεν προτείνεται καμία προσαρμογή αλλά η συνέχιση της κανονικής ροής της διαδικασίας. Εδώ παρότι έχει διαπιστωθεί καθυστέρηση στην ολοκλήρωση της εργασίας “Study advice” δεν προτείνεται κάποια προσαρμογή καθότι επικρατούν κανονικές καιρικές συνθήκες ενώ οι όποιες σχετικά αυξημένες μετρήσεις ραδιενέργειας (επίπεδο NOTIFY) δεν προκαλούν ανησυχία (αφού αν υπάρχει διαρροή ραδιενέργειας αυτή δεν θα εξαπλωθεί γρήγορα λόγω του ανέμου). Συνεπώς υπάρχει χρόνος για λήψη των σωστών αποφάσεων και για έγκαιρη αντίδραση.

Οι εναλλακτικές προσαρμογές που χρησιμοποιεί το παράδειγμα (ήτοι aspect 2 και aspect 4) παρουσιάζονται στο επόμενο Σχήμα 10-3.



Σχήμα 10-3. Προσαρμογές του παραδείγματος εφαρμοσμένες στη Διαδικασία χειρισμού κρίσεων

10.1.1 Επεξεργασία του κόμβου αφηρημένης ενέργειας του παραδείγματος

Ο κόμβος αφηρημένης ενέργειας "Recommend an advice" (Σχήμα 10-2) ρυθμίζεται ως εξής:

- Χρησιμοποιεί μια δεξαμενή ενεργειών με τις εναλλακτικές που περιγράφηκαν παραπάνω καθώς και με τα σχετικά μεταδεδομένα.
- Χρησιμοποιεί την πολυκριτηριακή μέθοδο αναζήτησης και επιλογής εναλλακτικών με LOWA.
- Χρησιμοποιεί την πολιτική επίλυσης ONE-FIRST, δηλαδή επιλέγεται η πρώτη εναλλακτική (ranked 1st) στην ταξινομημένη λίστα αποτελεσμάτων.

Στον επεξεργαστή των SANs (SAN editor) κάνοντας διπλό κλικ στον κόμβο αφηρημένης ενέργειας ανοίγει πλαίσιο διαλόγου (dialog box) που εμφανίζει τα κριτήρια που χρησιμοποιεί η LOWA καθώς και τα βάρη τους (Σχήμα 10-4). Οι τιμές των κριτηρίων θα ληφθούν από το πλαίσιο (context) και θα πρέπει να είναι γλωσσικοί όροι (normal, notify, alert, dangerous, critical), οπότε δεν χρειάζεται να καθοριστεί κάποια αντιστοίχιση μεταξύ αριθμητικών τιμών των μετρήσεων και των γλωσσικών όρων.

Edit Action

ID (URI)

Name

Type **Abstract**

Action Pool

Selection Method **LOWA** Resolution Policy **ONE FIRST (Executes the first job in the results list)**

Selection Method Conf: Normal, Notify, Alert, Dangerous, Critical

Criteria	Name	Weight	Type	Direction
	Weather	0.3	string	N/A
	Radioactivity	0.7	string	N/A

Name Weight Type **STRING** Direction **N/A**

Add **Delete** **Edit Mappings** **Normalize** Sum=0.3

Σχήμα 10-4. Επεξεργασία παραμέτρων της μεθόδου LOWA

10.1.2 Εκτέλεση του παραδείγματος για την προσαρμογή “Aspect 4”

Τα δύο επόμενα σχήματα απεικονίζουν δύο γεγονότα προτάσεων προσαρμογών που παρήγαγε και απέστειλε στη ροή γεγονότων “Process Adaptation Recommendations”, το SAN του παραδείγματος.

```

Process Adaptation Recommendations Unsubscribe
ADAPTATION_RECOMMENDATION_EVENT
<http://events.event-processing.org/ids/arp-1360591395692999936#event>
  a :ADAPTATION_RECOMMENDATION_EVENT ;
  :endTime "2013-02-11T14:03:15Z"^^xsd:dateTime ;
  :stream <http://streams.event-processing.org/ids/AsyncAdaptationRecommendations#stream> ;
  <http://imu.ntua.gr/play/flexibpnm2/adaptation-events/1#ask-user>
    "no" ;
  <http://imu.ntua.gr/play/flexibpnm2/adaptation-events/1#engine-id>
    "flexibpnm2-engine-1360591302024" ;
  <http://imu.ntua.gr/play/flexibpnm2/adaptation-events/1#event-type>
    "IS_RECOMMENDATION" ;
  <http://imu.ntua.gr/play/flexibpnm2/adaptation-events/1#id>
    "arp-1360591395692999936" ;
  <http://imu.ntua.gr/play/flexibpnm2/adaptation-events/1#message>
    "" ;
  <http://imu.ntua.gr/play/flexibpnm2/adaptation-events/1#session-id>
    "flexibpnm2-session-9467461f_9c8c_4474_a32b_b5e8d1973424" ;
  <http://imu.ntua.gr/play/flexibpnm2/adaptation-recommendation-events/1#action>
    "activate-aspect" ;
  <http://imu.ntua.gr/play/flexibpnm2/adaptation-recommendation-events/1#aspects>
    "http://147.102.23.45:8080/adapt/advise-2-d35fb1e3-38a3-420d-ab13-7426a1efddb9.xml" .
ADAPTATION_RECOMMENDATION_EVENT
    
```

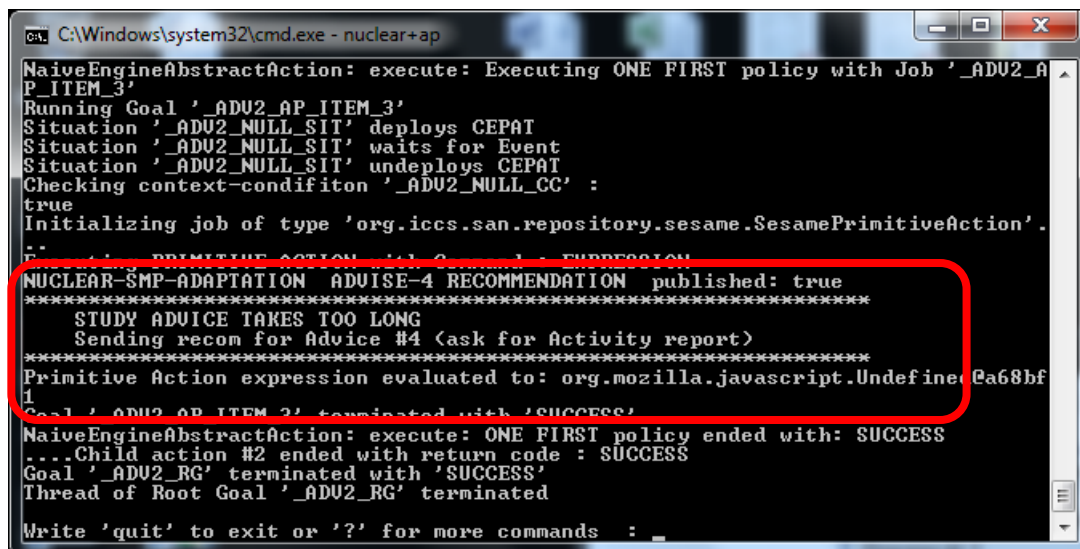
Σχήμα 10-5. Γεγονός πρότασης εφαρμογής της Προσαρμογής “Aspect 2”

```

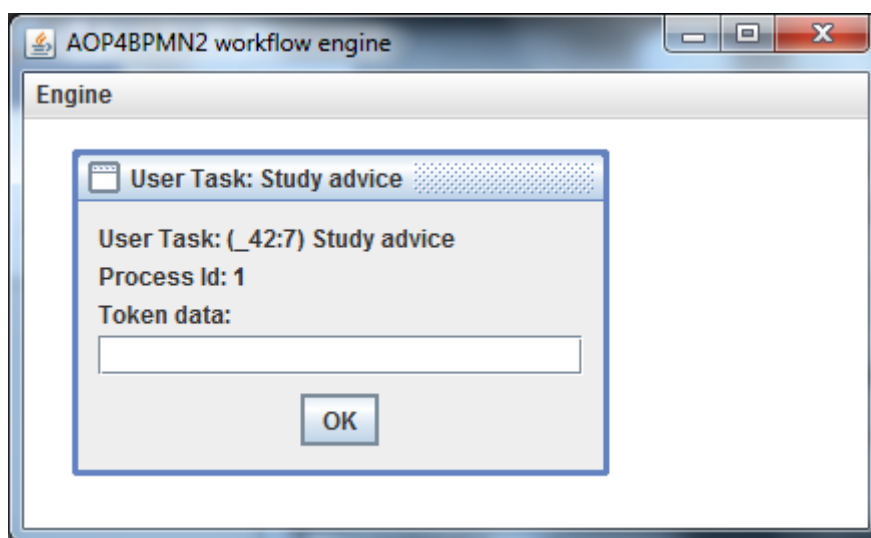
Service Adaptation Recommendations Unsubscribe
ADAPTATION_RECOMMENDATION_EVENT
<http://events.event-processing.org/ids/arp-1360591459407000064#event>
  a :ADAPTATION_RECOMMENDATION_EVENT ;
  :endTime "2013-02-11T14:04:19Z"^^xsd:dateTime ;
  :stream <http://streams.event-processing.org/ids/AsyncAdaptationRecommendations#stream> ;
  <http://imu.ntua.gr/play/flexibpnm2/adaptation-events/1#ask-user>
    "no" ;
  <http://imu.ntua.gr/play/flexibpnm2/adaptation-events/1#engine-id>
    "flexibpnm2-engine-1360591302024" ;
  <http://imu.ntua.gr/play/flexibpnm2/adaptation-events/1#event-type>
    "IS_RECOMMENDATION" ;
  <http://imu.ntua.gr/play/flexibpnm2/adaptation-events/1#id>
    "arp-1360591459407000064" ;
  <http://imu.ntua.gr/play/flexibpnm2/adaptation-events/1#message>
    "" ;
  <http://imu.ntua.gr/play/flexibpnm2/adaptation-events/1#session-id>
    "flexibpnm2-session-9467461f_9c8c_4474_a32b_b5e8d1973424" ;
  <http://imu.ntua.gr/play/flexibpnm2/adaptation-recommendation-events/1#action>
    "activate-aspect" ;
  <http://imu.ntua.gr/play/flexibpnm2/adaptation-recommendation-events/1#aspects>
    "http://147.102.23.45:8080/adapt/advise-4-35b3cde6-da1b-47e5-a8c3-f78a9b684d98.xml" .
ADAPTATION_RECOMMENDATION_EVENT
    
```

Σχήμα 10-6. Γεγονός πρότασης εφαρμογής της Προσαρμογής “Aspect 4”

Στην προσαρμογή “Aspect 4” μεταβάλλεται η κανονική ροή της διαδικασίας διαχείρισης κρίσεων και η εκτέλεση (μετά την “Study advice”) συνεχίζει στην κλάδο αίτησης, λήψης και μελέτης της αναφοράς ενεργειών (request/receive/study report), όπως αυτή απεικονίζεται στο Σχήμα 10-1. Τα επόμενα σχήματα δείχνουν αυτήν ακριβώς την προσαρμογή.

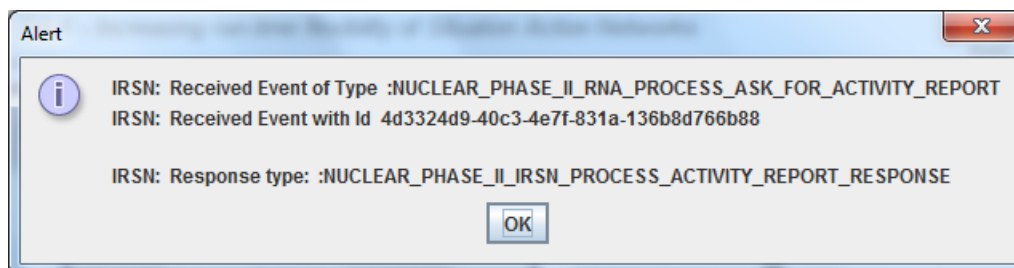


Σχήμα 10-7. Μηνύματα του μηχανισμού των SANs καθώς αποστέλλει την πρόταση εφαρμογής της προσαρμογής “Aspect 4”



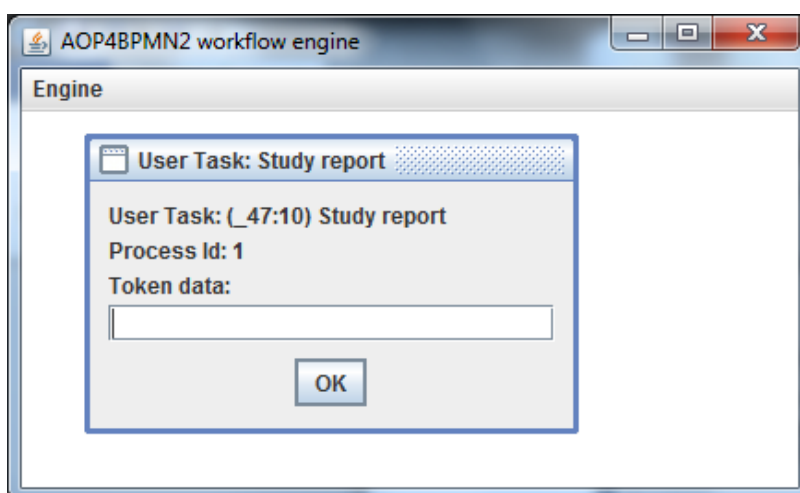
Σχήμα 10-8. Η εργασία “Study advice” (πριν την εφαρμογή της προσαρμογής)

Υποθέτοντας ότι η πρόταση προσαρμογής γίνεται δεκτή, όταν ολοκληρωθεί η εργασία “Study advice” ενεργοποιείται η προσαρμογή “Aspect 4” και μεταφέρει τη ροή εκτέλεσης στον κλάδο της διαδικασίας διαχείρισης κρίσεων που αφορά την αίτηση, λήψη και μελέτη αναφοράς ενεργειών (activity report). Η πρώτη εργασία (task) του κλάδου αυτού είναι να αποστείλει την αίτηση για αναφορά ενεργειών στο ινστιτούτο ραδιοπροστασίας (IRSN). Για την εκτέλεση του παραδείγματος χρησιμοποιήθηκε μια διαδικασία που προσομοιώνει το IRSN και η οποία εμφανίζει το ακόλουθο μήνυμα όταν λάβει την αίτηση για αναφορά ενεργειών.



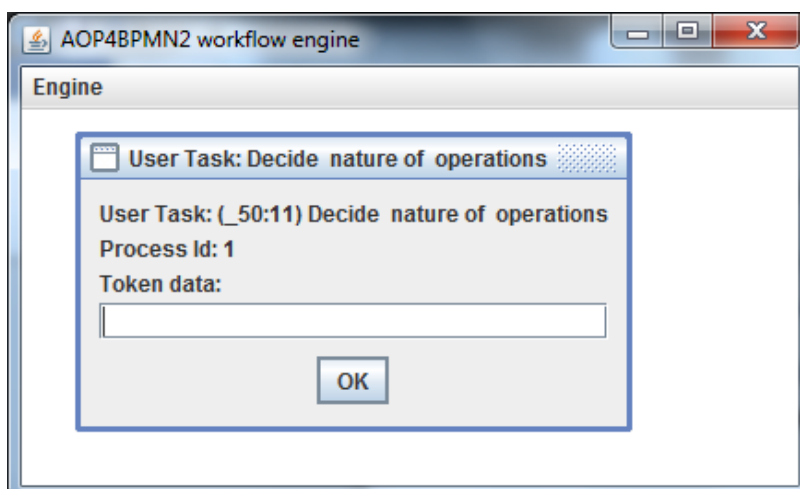
Σχήμα 10-9. Μήνυμα της διαδικασίας του IRSN – Αίτηση αναφοράς ενεργειών (Activity report)

Πατώντας το κουμπί OK προσομοιώνεται η απάντηση του ινστιτούτου ραδιοπροστασίας με την αποστολή μιας αναφοράς ενεργειών στην (ήδη) προσαρμοσμένη διαδικασία διαχείρισης κρίσεων. Οπότε με τη λήψη της αναφοράς ξεκινά την εργασία “Study report” που εμφανίζει το αντίστοιχο μήνυμα (Σχήμα 10-10).



Σχήμα 10-10. Λήψη αναφοράς ενεργειών και εκκίνηση της εργασίας “Study report”

Μετά την ολοκλήρωση της εργασίας “Study report” ολοκληρώνεται και η προσαρμογή, και έτσι η εκτέλεση της διαδικασίας διαχείρισης κρίσεων επανέρχεται στην κανονική ροή της, δηλαδή στην εργασία “Decide nature of operation” (Σχήμα 10-11).



Σχήμα 10-11. Η εργασία της κανονικής ροής, μετά την ολοκλήρωση της προσαρμογής

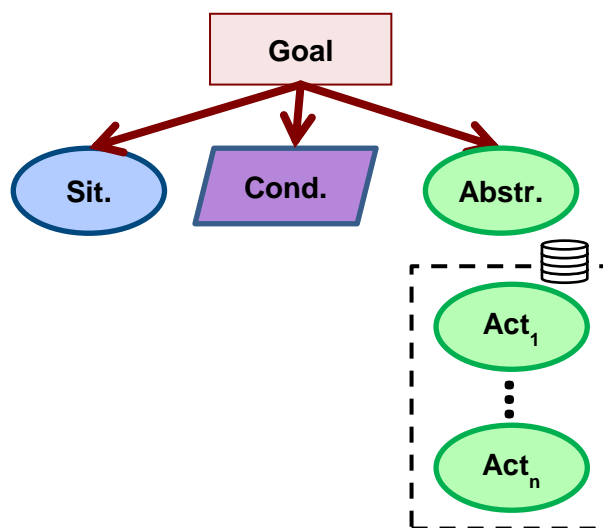
10.2 Αξιολόγηση των Κόμβων Αφηρημένων Ενεργειών

Στη συνέχεια του κεφαλαίου επιχειρείται μια αξιολόγηση των κόμβων αφηρημένων ενεργειών και των δεξαμενών ενεργειών με βάση τις επεκτάσεις που επέφεραν στην σημειογραφία και τη σημασιολογία των SANs, στον αλγόριθμο διάσχισής τους καθώς και στην υλοποίηση του μηχανισμού των SANs. Ιδίως εστιάζουμε στα πλεονεκτήματα αυτών των επεκτάσεων όσον αφορά τις δυνατότητες μοντελοποίησης και την εκφραστικότητα που παρέχουν. Ακόμη συζητείται ένας τρόπος πειραματικής σύγκρισης των επιδόσεων των SANs με χρήση κόμβων αφηρημένων ενεργειών και χωρίς αυτούς. Τέλος παρουσιάζονται και σχολιάζονται τα ευρήματα από την εκτέλεση των πειραμάτων.

10.2.1 Βελτίωση εκφραστικότητας με χρήση κόμβων αφηρημένων ενεργειών

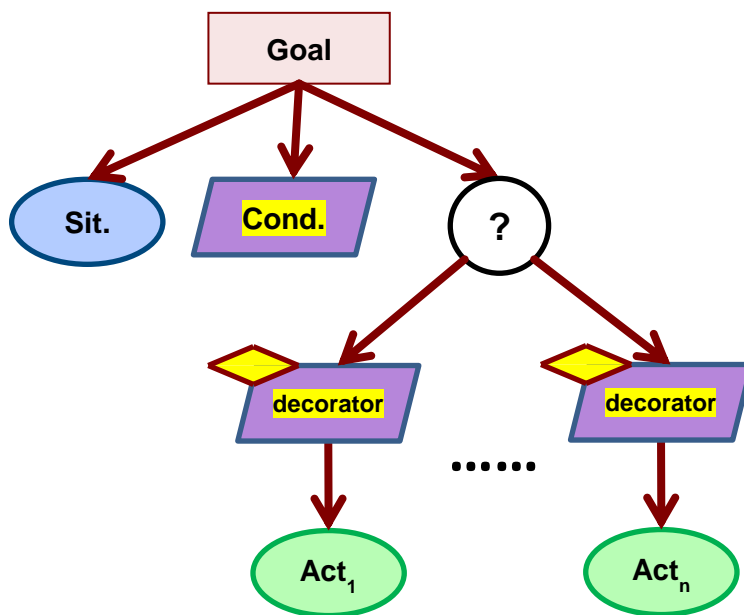
Η σημειογραφία των SANs ενσωματώνει διάφορους τύπους σύνθετων ενεργειών που μπορούν να συνδέσουν ένα στόχο με πολλαπλές ενέργειες όπως είναι οι κόμβοι κατ' επιλογή εκτέλεσης (Selector), οι κόμβοι σειριακής εκτέλεσης (Sequence) και οι κόμβοι παράλληλης εκτέλεσης (parallel-any, parallel-all, parallel-timeout). Προκειμένου να μελετήσουμε και τεκμηριώσουμε τη βελτίωση της εκφραστικότητας των SANs συγκρίνουμε τους κόμβους αυτούς με τους κόμβους αφηρημένων ενεργειών, χρησιμοποιώντας απλά αλλά αρκετά γενικά παραδείγματα, και έπειτα θα σχολιάσουμε τις διαφορές τους.

Έστω ένα SAN που αποτελείται από έναν στόχο Goal, ο οποίος αναλύεται στους κόμβους Situation, Context Condition και Abstract Action. Ο τελευταίος συνδέεται με κατάλληλη δεξαμενή ενεργειών που περιέχει τις εναλλακτικές Action-1 έως Action-N. Το αντίστοιχο SAN απεικονίζεται στο Σχήμα 10-12.

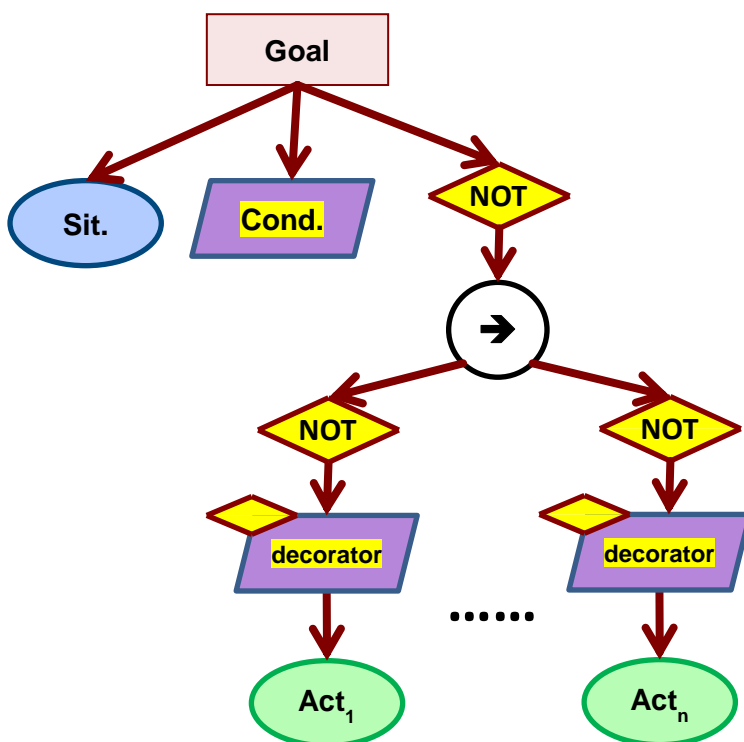


Σχήμα 10-12. SAN με δεξαμενή ενεργειών

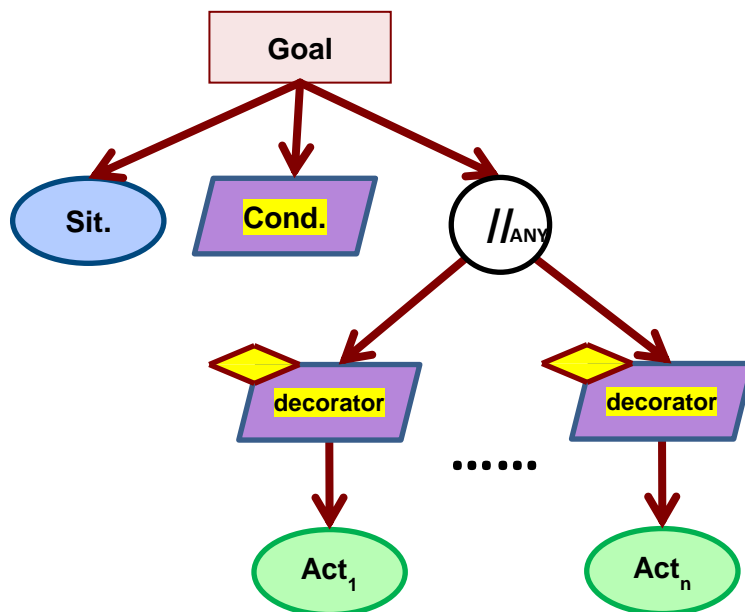
Αν επιχειρήσουμε να υλοποιήσουμε το ίδιο SAN χωρίς τη χρήση κόμβου αφηρημένης ενέργειας και δεξαμενής ενεργειών καταλήγουμε σε κάποιο από τα ακόλουθα SANs (Σχήμα 10-13, Σχήμα 10-14, Σχήμα 10-15 και Σχήμα 10-16). Το κύριο χαρακτηριστικό τους είναι η χρήση decorators συνθήκης, που λειτουργούν ουσιαστικά ως δομές ελέγχου της ροής εκτέλεσης, τύπου *if-then*. Είναι ξεκάθαρο ότι τα SANs χωρίς τον κόμβο αφηρημένης ενέργειας είναι αρκετά πολυπλοκότερα.



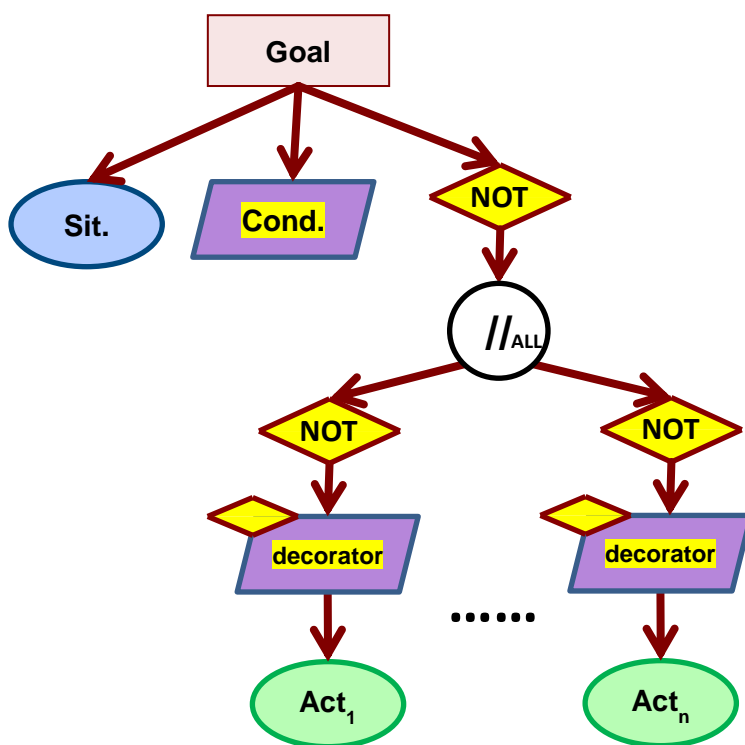
Σχήμα 10-13. SAN με κόμβο κατ' επιλογή εκτέλεσης (Selector)



Σχήμα 10-14. SAN με κόμβο σειριακής εκτέλεσης (Sequence)

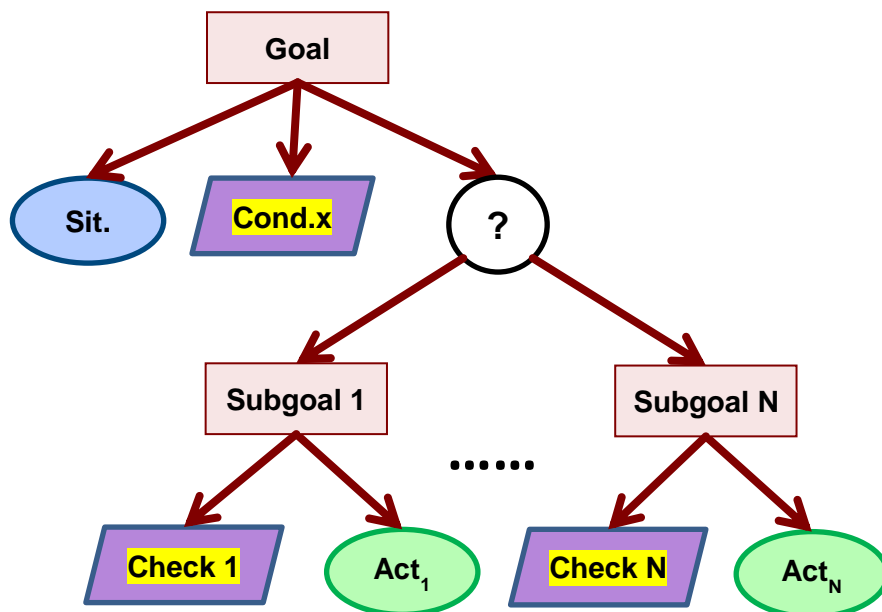


Σχήμα 10-15. SAN με κόμβο παράλληλης εκτέλεσης τύπου ANY



Σχήμα 10-16. SAN με κόμβο παράλληλης εκτέλεσης τύπου ALL

Να συμπληρώσουμε ακόμη ότι είναι δυνατή η μοντελοποίηση του ίδιου SAN χωρίς decorators συνθήκης αλλά με χρήση υποστόχων που θα έχουν μια εναλλακτική ως ενέργειά τους. Οι υποστόχοι περιλαμβάνουν κόμβο συνθήκης πλαισίου εκτέλεσης και έτσι μπορούν να ελέγξουν την εκτέλεση ή μη της εναλλακτικής. Ο τρόπος αυτός είναι ισοδύναμος με τη χρήση decorator συνθήκης αλλά οδηγεί σε ακόμη συνθετότερα μοντέλα.



Σχήμα 10-17. SAN με κόμβο κατ' επιλογή εκτέλεσης με χρήση υποστόχων για επιλογή

Συγκρίνοντας τα SANs με κόμβο αφηρημένης ενέργειας και τα SANs με κόμβους σύνθετων ενεργειών διαπιστώνουμε τα ακόλουθα:

- (α) Κατ' αρχήν η προσθήκη μιας νέας εναλλακτικής, όταν δε χρησιμοποιείται κόμβος αφηρημένης ενέργειας (με δεξαμενή ενεργειών), απαιτεί την αλλαγή του μοντέλου του SAN, δηλαδή τον επανασχεδιασμό του. Αν όμως χρησιμοποιηθεί δεξαμενή ενεργειών τότε η προσθήκη μπορεί να γίνει πολύ εύκολα χωρίς να απαιτείται κάποια αλλαγή στο μοντέλο του SAN. Η εισαγωγή μπορεί να γίνει είτε κατά το σχεδιασμό είτε κατά την εκτέλεση, μέχρι και τη στιγμή προτού ξεκινήσει η μέθοδος αναζήτησης και επιλογής να σαρώνει και να εξετάζει τα περιεχόμενα της δεξαμενής ενεργειών.
- (β) Με χρήση κόμβων κατ' επιλογή ή σειριακής εκτέλεσης δεν μπορεί να αλλάξει η σειρά εξέτασης των ενεργειών κατά το χρόνο εκτέλεσης. Αντίθετα στους κόμβους αφηρημένης ενέργειας η σειρά αποφασίζεται σε κάθε εκτέλεση από τη μέθοδο αναζήτησης και επιλογής.
- (γ) Η χρήση decorators συνθήκης για την επιλογή εναλλακτικών δεν εξασφαλίζει τον αμοιβαίο αποκλεισμό των υπόλοιπων εναλλακτικών όταν πρέπει να επιλέγεται μόνο μία. Αυτό είναι ένα σοβαρό πρόβλημα απ' το οποίο πάσχουν κυρίως οι κόμβοι παράλληλων ενεργειών. Στους κόμβους αφηρημένης ενέργειας απ' την άλλη, μπορεί να επιλεγεί η καλύτερη εναλλακτική που προκύπτει από την ταξινομημένη σειρά των εναλλακτικών (για παράδειγμα με μια πολιτική επίλυσης ONE-FIRST).
- (δ) Οι κόμβοι σύνθετης ενέργειας δεν μπορούν να αποκλείσουν, κατά την εκτέλεση, εναλλακτικές που είναι ενδεχομένως ακατάλληλες σε μια δεδομένη κατάσταση. Οι κόμβοι αφηρημένων ενεργειών όμως μπορούν να φιλτράρουν τις εναλλακτικές με βάση τα μεταδεδομένα τους και τις πληροφορίες του πλαισίου.
- (ε) Οι κόμβοι σύνθετης ενέργειας δεν μπορούν να ταξινομήσουν τις εναλλακτικές με βάση κάποια βαθμολογία ή κάποιο άλλο κριτήριο και να επιλέξουν κατόπιν την

καλύτερη ή τις καλύτερες. Αντίθετα επιλέγουν την πρώτη που θα θεωρήσουν ως κατάλληλη. Οι κόμβοι αφηρημένων ενεργειών απ' την άλλη, πρώτα βαθμολογούν και ταξινομούν τις εναλλακτικές και κατόπιν επιλέγουν μία ή περισσότερες.

- (στ) Για την επιλογή μιας εναλλακτικής, οι κόμβοι σύνθετης ενέργειας εξετάζουν πληροφορίες που περιέχονται στο πλαίσιο εκτέλεσης, οι οποίες είτε προέρχονται από γεγονότα είτε υπολογίζονται κατά την εκτέλεση του SAN. Δεν είναι όμως εύκολο να εξετάσουν πληροφορίες που αφορούν την εκάστοτε εναλλακτική ξεχωριστά. Στους κόμβους αφηρημένης ενέργειας εκτός απ' τις πληροφορίες του πλαισίου εκτέλεσης συνεκτιμώνται και τα μεταδεδομένα της κάθε εναλλακτικής.
- (ζ) Οι κόμβοι κατ' επιλογή εκτέλεσης και σειριακής εκτέλεσης εξετάζουν διαδοχικά τις εναλλακτικές μέχρι να επιλεγεί κάποια ενώ οι επόμενες απλά αγνοούνται. Έτσι εξετάζεται ένας απροσδιόριστος αριθμός εναλλακτικών προτού επιλεγεί κάποια. Αντίθετα, οι κόμβοι αφηρημένης ενέργειας προτού επιλέξουν εξετάζουν όλες τις εναλλακτικές (συνεκτιμώντας τα μεταδεδομένα τους). Το γεγονός αυτό καθιστά τη μέθοδο αυτή δικαιότερη και πιο ακριβή σε σχέση με την προσέγγιση με χρήση μόνο κόμβων σύνθετης ενέργειας.
- (η) Οι κόμβοι παράλληλης εκτέλεσης μπορεί να δημιουργήσουν σημαντικό φόρτο (έστω μικρής χρονικής διάρκειας) στο σύστημα, αφού θα επιχειρήσουν να εκτελέσουν όλα τους τα κλαδιά ταυτόχρονα δημιουργώντας έτσι πολλά νήματα εκτέλεσης. Φυσικά τα περισσότερα θα τερματίσουν σύντομα, όταν ο decorator συνθήκης που τα ελέγχει αποτιμηθεί σε ψευδής. Στους κόμβους αφηρημένης κατάστασης η συμπεριφορά αυτή εξαρτάται από την υλοποίηση της μεθόδου αναζήτησης και επιλογής. Είναι όμως λογικό οι απαιτούμενες εργασίες να γίνουν με χρήση ενός νήματος εκτέλεσης (thread), όπως συμβαίνει στη δική μας υλοποίηση, ή για λόγους βελτιστοποίησης με χρήση ενός προκαθορισμένου αριθμού νημάτων, τα οποία θα εξετάσουν παράλληλα τις εναλλακτικές χωρίς όμως να δημιουργούν ιδιαίτερα υψηλό φόρτο στο σύστημα. Υπάρχει επίσης η δυνατότητα τα νήματα αυτά να επαναχρησιμοποιούνται.

Από τα παραπάνω παραδείγματα γίνεται φανερό ότι η εισαγωγή των κόμβων αφηρημένων ενεργειών, των δεξαμενών ενεργειών και η χρήση των πολυκριτηριακών μεθόδων λήψης αποφάσεων για επιλογή ενεργειών, βελτίωσαν σημαντικά την εκφραστικότητα των SANs.

10.2.2 Πειραματική αξιολόγηση

Σε αυτή την ενότητα εστιάζουμε στην αξιολόγηση της εκτέλεσης των SANs με κόμβους αφηρημένων ενεργειών και δεξαμενές ενεργειών σε σχέση με ισοδύναμα SANs που δε χρησιμοποιούν κόμβους αφηρημένων ενεργειών. Σημειώνουμε ότι η αξιολόγηση αυτή αφορά περιπτώσεις όπου μπορούν να σχεδιαστούν ισοδύναμα SANs χωρίς κόμβους αφηρημένων ενεργειών.

Στο πλαίσιο αυτό σχεδιάστηκαν και εκτελέστηκαν σειρές πειραμάτων με σκοπό να καταδείξουν τις διαφορές στις επιδόσεις και την αποδοτικότητα της χρήσης των SANs με τη χρήση και χωρίς τη χρήση κόμβων αφηρημένων ενεργειών. Σε κάθε σειρά πειραμάτων

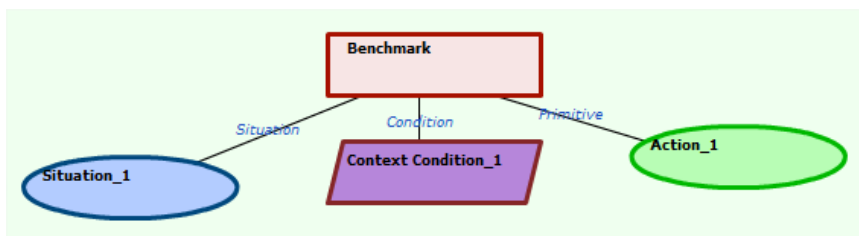
χρησιμοποιήθηκε ένα συγκεκριμένο (απλό) SAN και δοκιμάστηκε σε διάφορους ρυθμούς αποστολής γεγονότων. Συγκεκριμένα πειραματιστήκαμε με τα ακόλουθα SANs.

- SAN-A-baseline: SAN ενός επιπέδου, του οποίου οι επιδόσεις χρησιμοποιούνται ως βάση αναφοράς (baseline) (Σχήμα 10-18)
- SAN-B-Selector-1-choice: SAN ενός επιπέδου με χρήση κόμβου κατ' επιλογή εκτέλεσης (Selector), ο οποίος με βάση τις συνθήκες πλαισίου επιλέγει μία (μόνο) ενέργεια για εκτέλεση
- SAN-B-Selector-10-choices: SAN ενός επιπέδου με χρήση κόμβου κατ' επιλογή εκτέλεσης (Selector), ο οποίος με βάση τις συνθήκες πλαισίου επιλέγει για εκτέλεση μεταξύ 10 ενεργειών (Σχήμα 10-20)
- SAN-B-Selector-100-choices: SAN ενός επιπέδου με χρήση κόμβου κατ' επιλογή εκτέλεσης (Selector), ο οποίος με βάση τις συνθήκες πλαισίου επιλέγει για εκτέλεση μεταξύ 100 ενεργειών
- SAN-B-Selector-1000-choices: SAN ενός επιπέδου με χρήση κόμβου κατ' επιλογή εκτέλεσης (Selector), ο οποίος με βάση τις συνθήκες πλαισίου επιλέγει για εκτέλεση μεταξύ 1000 ενεργειών
- SAN-C-Action-Pool-1-choice: SAN ενός επιπέδου με χρήση κόμβου αφηρημένης ενέργειας και δεξαμενής ενεργειών με μία ενέργεια και ένα κριτήριο επιλογής
- SAN-C-Action-Pool-10-choices: SAN ενός επιπέδου με χρήση κόμβου αφηρημένης ενέργειας και δεξαμενής ενεργειών με 10 εναλλακτικές ενέργειες (Σχήμα 10-19) και ένα κριτήριο επιλογής
- SAN-C-Action-Pool-100-choices: SAN ενός επιπέδου με χρήση κόμβου αφηρημένης ενέργειας και δεξαμενής ενεργειών με 100 εναλλακτικές και ένα κριτήριο επιλογής
- SAN-C-Action-Pool-1000-choices: SAN ενός επιπέδου με χρήση κόμβου αφηρημένης ενέργειας και δεξαμενής ενεργειών με 1000 εναλλακτικές και ένα κριτήριο επιλογής
- SAN-D-Action-Pool-10-criteria: SAN ενός επιπέδου με χρήση κόμβου αφηρημένης ενέργειας και δεξαμενής ενεργειών με μία εναλλακτική και 10 κριτήρια επιλογής
- SAN-D-Action-Pool-50-criteria: SAN ενός επιπέδου με χρήση κόμβου αφηρημένης ενέργειας και δεξαμενής ενεργειών με μία εναλλακτική και 50 κριτήρια επιλογής
- SAN-D-Action-Pool-100-criteria: SAN ενός επιπέδου με χρήση κόμβου αφηρημένης ενέργειας και δεξαμενής ενεργειών με μία εναλλακτική και 100 κριτήρια επιλογής

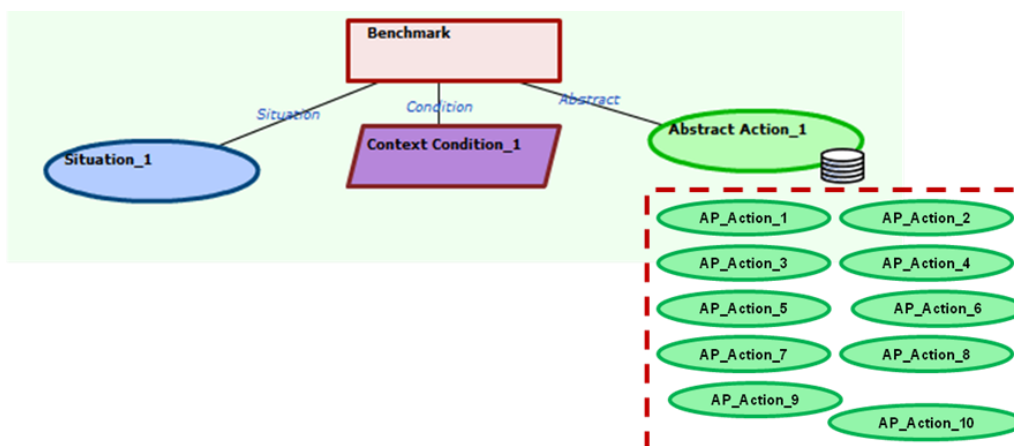
Τα παραπάνω SANs χρησιμοποιήθηκαν σε σειρές πειραμάτων όπου από εκτέλεση σε εκτέλεση μεταβαλλόταν ο ρυθμός αποστολής των γεγονότων. Φυσικά στη διάρκεια μιας εκτέλεσης ο ρυθμός παρέμεινε σταθερός. Για τις ανάγκες της μελέτης και προκειμένου να εξαλειφθούν οι καθυστερήσεις που οφείλονται στον δίαυλο γεγονότων ή στο δίκτυο, συνδέσαμε την πηγή των γεγονότων απευθείας με τον μηχανισμό των SANs. Έτσι ο ρυθμός αποστολής γεγονότων της πηγής συνέπιπτε με το ρυθμό λήψης γεγονότων από τον μηχανισμό των SANs. Έτσι κατέστη δυνατή η μέτρηση του καθαρού χρόνου επεξεργασίας ενός γεγονότος από τον μηχανισμό των SANs. Η πηγή των γεγονότων είναι λογισμικό που διαβάζει τα γεγονότα από ένα αρχείο και τα στέλνει με συγκεκριμένο σταθερό ρυθμό (ανά εκτέλεση).

Ειδικότερα τώρα, τα τέσσερα SAN-B-Selector SANs πρόκειται για επεκτάσεις του SAN-A-baseline (Σχήμα 10-18) που χρησιμοποιείται ως βάση αναφοράς, αλλά αντί για μια απλή ενέργεια (primitive action) έχουν στη θέση της έναν κόμβο κατ' επιλογή εκτέλεσης (Selector node) που επιλέγει μία από τις ενέργειες – επιλογές (choices) που έχει ως παιδιά του. Τα τέσσερα SAN-B-Selector έχουν 1, 10 (Σχήμα 10-20), 100 και 1000 επιλογές. Για λόγους καλύτερης σύγκρισης οι κόμβοι κατ' επιλογή εκτέλεσης ρυθμίστηκαν ώστε να επιλέγουν πάντα την τελευταία εναλλακτική αφού και η μέθοδος αναζήτησης και επιλογής των κόμβων αφηρημένων ενεργειών των SAN-C-Action-Pool SANs σαρώνει άλλες τις εναλλακτικές. Όλα τα SANs-C-Action-Pool SANs αντιστοιχούν σε κάποιο SAN-B-Selector, αλλά σχεδιάστηκαν με χρήση κόμβων αφηρημένων ενεργειών που χρησιμοποιούν τη μέθοδο αναζήτησης και επιλογής με LOWA και ένα κριτήριο επιλογής. Οι δεξαμενές ενεργειών περιέχουν αντίστοιχα 1, 10 (Σχήμα 10-19), 100 και 1000 εναλλακτικές ενέργειες. Κατ' αυτόν τον τρόπο είναι δυνατή η σύγκριση των δύο σχεδιαστικών προσεγγίσεων για διάφορα πλήθη εναλλακτικών.

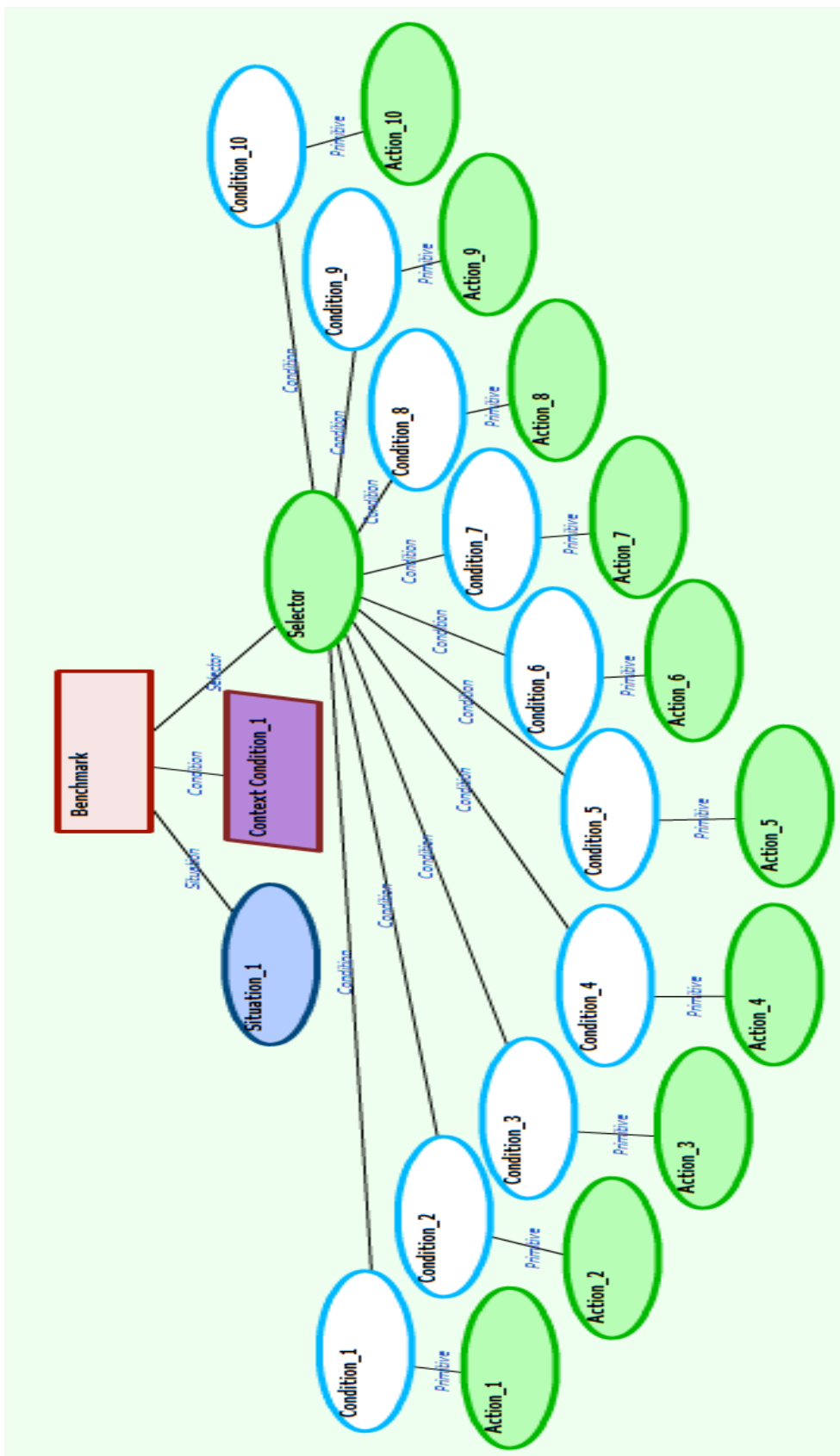
Τέλος, τα SAN-D-Action-Pool SANs αφορούν τη μελέτη της συμπεριφοράς των δεξαμενών ενεργειών καθώς αυξάνεται ο αριθμός των κριτηρίων επιλογής που χρησιμοποιείται. Ως βάση αναφοράς θα χρησιμοποιηθεί το SAN-C-Action-Pool-1-choice που περιέχει μία εναλλακτική ενέργεια και ένα κριτήριο επιλογής. Οι μετρήσεις των πειραμάτων αυτών δεν είναι συγκρίσιμες με εκείνες των προηγούμενων πειραμάτων (ειδικά των SAN-B-Selector) επειδή μεταβάλλεται το πλήθος των κριτηρίων επιλογής (10, 50, 100), το οποίο ήταν σταθερό (και ίσο με 1) στα πειράματα SAN-B-Selector και SAN-C-Action-Pool. Για το λόγο αυτό τα ευρήματα και οι παρατηρήσεις μας γι' αυτά θα παρουσιαστούν ξεχωριστά.



Σχήμα 10-18. Το SAN του πειράματος αναφοράς (baseline)



Σχήμα 10-19. SAN με δεξαμενή ενεργειών με 10 επιλογές



Σχήμα 10-20. SAN με κόμβο κατ' επιλογή εκτέλεσης με 10 επιλογές

Τα πειράματα εκτελέστηκαν σε έναν υπολογιστή με τα ακόλουθα χαρακτηριστικά:

Επεξεργαστής	Intel® Core™2 Quad CPU Q8300 @ 2.50GHz
Μνήμη (RAM)	4GB (3.5GB usable)
Λειτουργικό Σύστημα	32-bit OS, Windows 7 Professional SP1
Αποθηκευτικός χώρος	1.4TB, HDD ATA

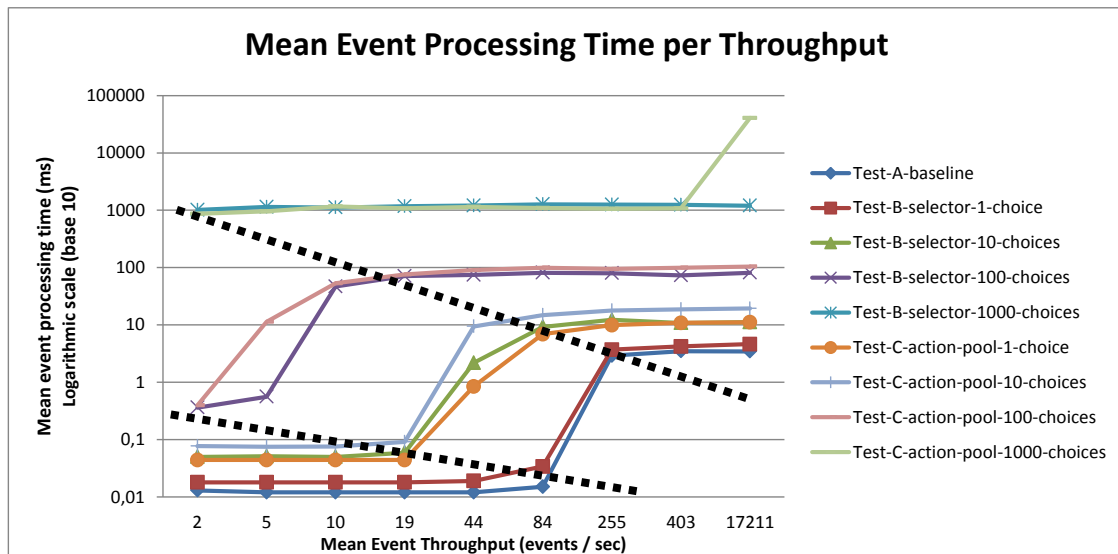
Για καθένα από τα παραπάνω πειράματα μετρήθηκαν τα ακόλουθα μεγέθη σε σχέση με το ρυθμό αποστολής των γεγονότων (event throughput), δηλαδή τον αριθμό των γεγονότων που αποστέλλονται στον μηχανισμό των SANs ανά δευτερόλεπτο.

- Μέσος χρόνος επεξεργασίας γεγονότος (Mean Event Processing Time). Ο χρόνος επεξεργασίας γεγονότος είναι το χρονικό διάστημα μεταξύ της λήψης ενός γεγονότος από τον μηχανισμό των SANs και την ολοκλήρωση και της τελευταίας ενέργειας του SAN. Οι εργασίες που εκτελούνται κατά την επεξεργασία ενός γεγονότος περιλαμβάνουν την τοποθέτησή του αρχικά στην ουρά εισόδου (incoming events queue), φιλτράρισμα των γεγονότων της ουράς εισόδου, δημιουργία ενός νέου τοπικού πλαισίου εκτέλεσης, δημιουργία ενός νέου στιγμιότυπου του SAN (για την επεξεργασία του γεγονότος) καθώς και αντίστοιχου νήματος εκτέλεσης, εξαγωγή των περιεχομένων του γεγονότος, contextualization των περιεχομένων του γεγονότος στο τοπικό πλαίσιο εκτέλεσης, έλεγχος των συνθηκών πλαισίου, εκτέλεση του κόμβου επιλογής ενέργειας, ήτοι κόμβου κατ' επιλογή εκτέλεσης (selector) ή κόμβου αφηρημένης ενέργειας και εκτέλεση της μεθόδου αναζήτησης και επιλογής (LOWA σε αυτά τα πειράματα), και τέλος εκτέλεση της ενέργειας που θα επιλεγεί.
- Μέγιστος αριθμός ενεργών νημάτων εκτέλεσης (Maximum Active Threads). Στην υλοποίηση του μηχανισμού των SANs, κάθε στιγμιότυπο SAN συσχετίζεται με ένα νήμα εκτέλεσης προκειμένου να εκτελεστεί. Ένα νήμα που έχει αρχίσει η εκτέλεσή του αλλά δεν έχει ολοκληρωθεί καλείται ενεργό νήμα (active thread). Ο αριθμός των ενεργών νημάτων, ήτοι ο αριθμός των νημάτων που εκτελούνται ταυτόχρονα, αναμένεται να έχει επίπτωση στην επίδοση του μηχανισμού των SANs. Εξάλλου τα νήματα είναι πόρος του συστήματος που καταναλώνουν κύκλους επεξεργασίας από τη CPU καθώς και μνήμη. Συνεπώς αυξημένος αριθμός ενεργών νημάτων εκτέλεσης στερεί πολύτιμους πόρους του συστήματος και υποδεικνύει την ύπαρξη συνθηκών ανταγωνισμού μεταξύ τους που μπορεί να οδηγήσουν σε καθυστερήσεις και σε μείωση των επιδόσεων. Αυτή η κατάσταση μπορεί να συμβεί με λιγότερα ή περισσότερα ενεργά νήματα, αναλόγως με τα χαρακτηριστικά και τις δυνατότητες του συστήματος. Σε τυπικούς υπολογιστές ευρείας κατανάλωσης αυτό συμβαίνει όταν εκτελούνται ταυτόχρονα αρκετές δεκάδες ή λίγες εκατοντάδες νήματα. Ο μέγιστος αριθμός ενεργών νημάτων εκτέλεσης είναι μια ένδειξη των «χειρότερων» συνθηκών εκτέλεσης που αντιμετώπισε ο μηχανισμός των SANs κατά τη διάρκεια ενός πειράματος.

Για την εκτέλεση των πειραμάτων χρησιμοποιήθηκε το ίδιο σύνολο γεγονότων, τα οποία εστάλησαν με διάφορους ρυθμούς (throughputs) με σκοπό να δοκιμάσουν τον μηχανισμό των SANs σε διάφορες συνθήκες φόρτου. Οι ρυθμοί που επετεύχθησαν κυμάνθηκαν από 2 έως 17.000 γεγονότα ανά δευτερόλεπτο. Ο στιγμιαίος ρυθμός αποστολής γεγονότων στη διάρκεια ενός πειράματος (instant event throughput) δεν ήταν σταθερός. Γι' αυτό στα διαγράμματα που ακολουθούν χρησιμοποιούνται οι μέσοι ρυθμοί αποστολής γεγονότων.

10.2.2.1 Μέσος χρόνος επεξεργασίας γεγονότος (Mean event processing time)

Το επόμενο διάγραμμα απεικονίζει τον μέσο χρόνο επεξεργασίας γεγονότος για κάθε πείραμα, καθώς ο ρυθμός αποστολής αυξάνεται από 2 σε περίπου 17.000 γεγονότα ανά δευτερόλεπτο.



Σχήμα 10-21. Μέσος χρόνος επεξεργασίας γεγονότων ανά ρυθμό αποστολής

Παρατηρώντας το Σχήμα 10-21 διαπιστώνεται η ύπαρξη τριών «ζωνών επίδοσης», οι οποίες σημειώνονται στο σχήμα με τις δύο διακεκομμένες γραμμές. Σε κάθε ζώνη ο μέσος χρόνος επεξεργασίας των γεγονότων παρουσιάζει διαφορετική συμπεριφορά, καθώς ο ρυθμός αποστολής γεγονότων αυξάνεται. Ειδικότερα, στη χαμηλή ζώνη, ο μέσος χρόνος επεξεργασίας γεγονότων παραμένει μικρός και σχεδόν σταθερός, ανεξάρτητα από την αύξηση του ρυθμού αποστολής γεγονότων. Στην μεσαία ζώνη, ο μέσος χρόνος αυξάνεται καθώς αυξάνεται και ο ρυθμός αποστολής γεγονότων. Σημειώνεται ότι στο διάγραμμα ο κατακόρυφος άξονας χρησιμοποιεί λογαριθμική κλίμακα με βάση 10. Άρα οι μεταβολές είναι στην πραγματικότητα εκθετικές κι όχι γραμμικές όπως φαίνονται γραφικά. Τέλος στην υψηλή ζώνη ο μέσος χρόνος επεξεργασίας γεγονότων είναι ξανά σχεδόν σταθερός. Στα περισσότερα πειράματα οι αντίστοιχες γραμμές επίδοσης προσομοιάζουν σε σιγμοειδείς καμπύλες (sigmoidal curves).

Στη χαμηλή ζώνη, όπου οι ρυθμοί αποστολής είναι χαμηλοί, τα γεγονότα λαμβάνονται με μια σχετικά μεγάλη χρονική απόσταση μεταξύ τους. Η απόσταση αυτή επαρκεί για την ολοκλήρωση της επεξεργασίας του γεγονότος προτού ληφθεί το επόμενο γεγονός. Συνεπώς τα γεγονότα δεν παραμένουν στην ουρά αναμονής αλλά η επεξεργασία τους αρχίζει

αμέσως με την παραλαβή τους. Επιπλέον, μόνο ελάχιστα SANs εκτελούνται ταυτόχρονα (άρα και νήματα εκτέλεσης) και έτσι δεν υπάρχουν συνθήκες ανταγωνισμού μεταξύ τους για χρήση κοινών πόρων. Αυτό έχει ως αποτέλεσμα τη γρήγορη ολοκλήρωση της επεξεργασίας των γεγονότων.

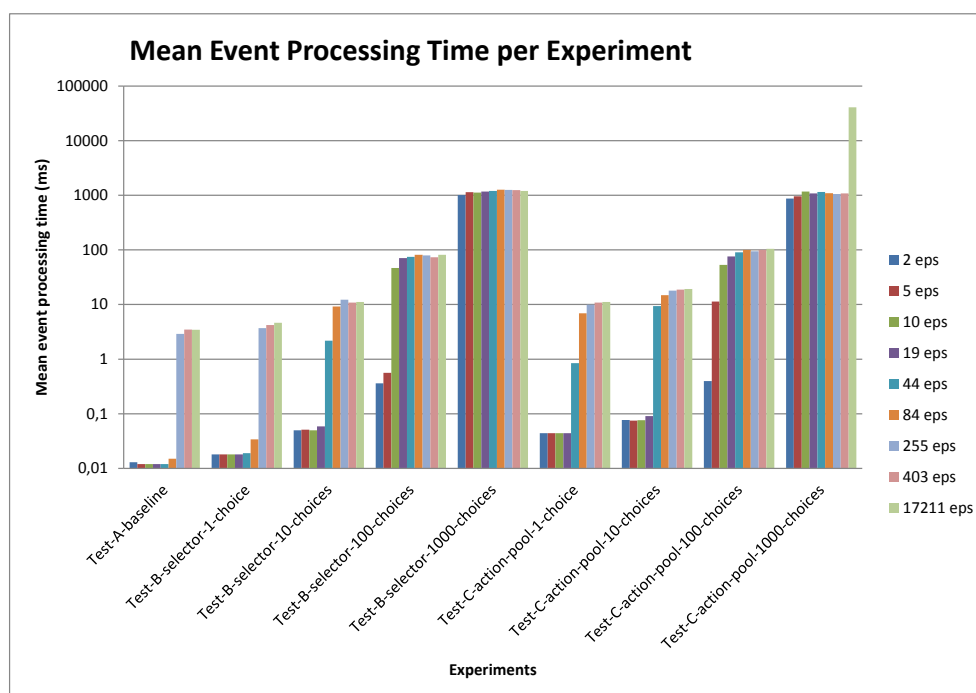
Καθώς όμως ο ρυθμός αποστολής αυξάνεται, στην μεσαία ζώνη, τα γεγονότα λαμβάνονται πιο γρήγορα από το ρυθμό δημιουργίας νέων στιγμιότυπων SANs (άρα και νημάτων εκτέλεσης) του μηχανισμού SANs. Αυτό έχει ως αποτέλεσμα κάποια γεγονότα να πρέπει να παραμείνουν στην ουρά αναμονής για κάποιο διάστημα. Σε αυτή τη ζώνη ο αριθμός των ενεργών SANs (και νημάτων εκτέλεσης) είναι σημαντικός και έχει ως αποτέλεσμα τη δημιουργία συνθηκών ανταγωνισμού (race conditions) μεταξύ τους και την αύξηση του μέσου χρόνου επεξεργασίας. Καθώς ο ρυθμός αποστολής γεγονότων αυξάνεται ο μέσος χρόνος επεξεργασίας αυξάνεται εκθετικά. Αυτό συμβαίνει για ρυθμούς αποστολής μεταξύ 20 και 100 γεγονότων το δευτερόλεπτο ωστόσο σε μερικά πειράματα (αυτά με 100 επιλογές) η αύξηση του χρόνου επεξεργασίας αρχίζει και σε πολύ μικρότερους ρυθμούς λόγω του αυξημένου αριθμού επιλογών που θα πρέπει να εξεταστούν για κάθε γεγονός. Τα πειράματα με 1000 επιλογές είναι εκτός της συγκεκριμένης ζώνης.

Στην υψηλή ζώνη, ο μέσος χρόνος επεξεργασίας γεγονότων παραμένει σχεδόν σταθερός και με υψηλές τιμές. Αυτό αποτελεί ένδειξη του ότι πολλά ενεργά στιγμιότυπα SANs (άρα και νήματα εκτέλεσης) ανταγωνίζονται για τη χρήση κοινών πόρων του συστήματος (κυρίως κύκλους CPU) με αποτέλεσμα να καθυστερεί το ένα το άλλο. Συνεπώς όλα τους εκτελούνται πιο αργά ενώ συσσωρεύονται νέα γεγονότα στην ουρά αναμονής. Τέλος σημειώνουμε ότι τα όρια των τριών ζωνών χαμηλώνουν καθώς αυξάνεται ο αριθμός των επιλογών.

Οι μετρήσεις των πειραμάτων με τις 1000 επιλογές εμφανίζονται αποκλειστικά στην υψηλή ζώνη πράγμα που σημαίνει ότι η επεξεργασία των γεγονότων απαιτεί σημαντικό χρόνο, και άρα οδηγεί σε καθυστερήσεις, ακόμη για χαμηλούς ρυθμούς αποστολής. Οι μετρήσεις των πειραμάτων με τις 100 επιλογές εμφανίζονται από τη μεσαία ζώνη και επεκτείνονται στην υψηλή, ενώ τα πειράματα με λιγότερες επιλογές έχουν μετρήσεις που καλύπτουν και τις τρεις ζώνες. Τα όρια των ζωνών εκτείνονται από 20 έως 90 περίπου γεγονότα το δευτερόλεπτο μεταξύ της χαμηλής και μεσαίας ζώνης, και από 50 έως 350 περίπου γεγονότα το δευτερόλεπτο μεταξύ μεσαίας και υψηλής ζώνης (εξαιρούνται τα πειράματα με 100 ή περισσότερες επιλογές).

Ακολούθως δίνεται το ίδιο διάγραμμα αλλά τώρα στον οριζόντιο άξονα εμφανίζονται τα πειράματα αντί για τους ρυθμούς αποστολής (Σχήμα 10-22). Ο κατακόρυφος άξονας είναι και πάλι σε λογαριθμική κλίμακα με βάση 10. Ο μέσος χρόνος επεξεργασίας των γεγονότων σε κάθε πείραμα αυξάνεται σημαντικά καθώς αυξάνεται ο ρυθμός αποστολής. Για τα πειράματα με λίγες επιλογές η αύξηση αυτή συμβαίνει σε σχετικά υψηλούς ρυθμούς. Ωστόσο σε πειράματα με πολλές επιλογές (100 ή 1000) η αύξηση ξεκινά από πολύ χαμηλούς ρυθμούς. Ακόμη παρατηρούμε ότι ο μέσος χρόνος επεξεργασίας γεγονότων στα πειράματα με τις 1000 επιλογές είναι περίπου 10 φορές παραπάνω από τον αντίστοιχο χρόνο των πειραμάτων με 100 επιλογές. Η ίδια περίπου σχέση (λίγο μικρότερη από 10 προς 1) ισχύει και μεταξύ των πειραμάτων με 100 και 10 επιλογές. Ωστόσο δε συμβαίνει το ίδιο

μεταξύ των πειραμάτων με 10 και μία επιλογή, όπου η σχέση είναι αρκετά μικρότερη. Ακόμη, ο χρόνος επεξεργασίας του πειράματος με χρήση δεξαμενών ενεργειών με 1000 επιλογές, σε υψηλούς ρυθμούς αποστολής (ακραία περίπτωση) είναι περίπου 34 φορές υψηλότερος από τον αντίστοιχο χρόνο του πειράματος με κόμβο κατ' επιλογή εκτέλεσης (Selector) και 1000 επιλογές. Φυσικά ο χειρότερος χρόνος επεξεργασίας αντισταθμίζεται από τα πλεονεκτήματα και την ευελιξία των δεξαμενών ενεργειών που συζητήθηκαν σε προηγούμενες ενότητες.



Σχήμα 10-22. Μέσος χρόνος επεξεργασίας γεγονότων ανά πείραμα

Συγκρίνοντας τις δύο σχεδιαστικές προσεγγίσεις για ίδιους αριθμούς επιλογών έχουμε τους ακόλουθους μέσους χρόνους επεξεργασίας γεγονότων.

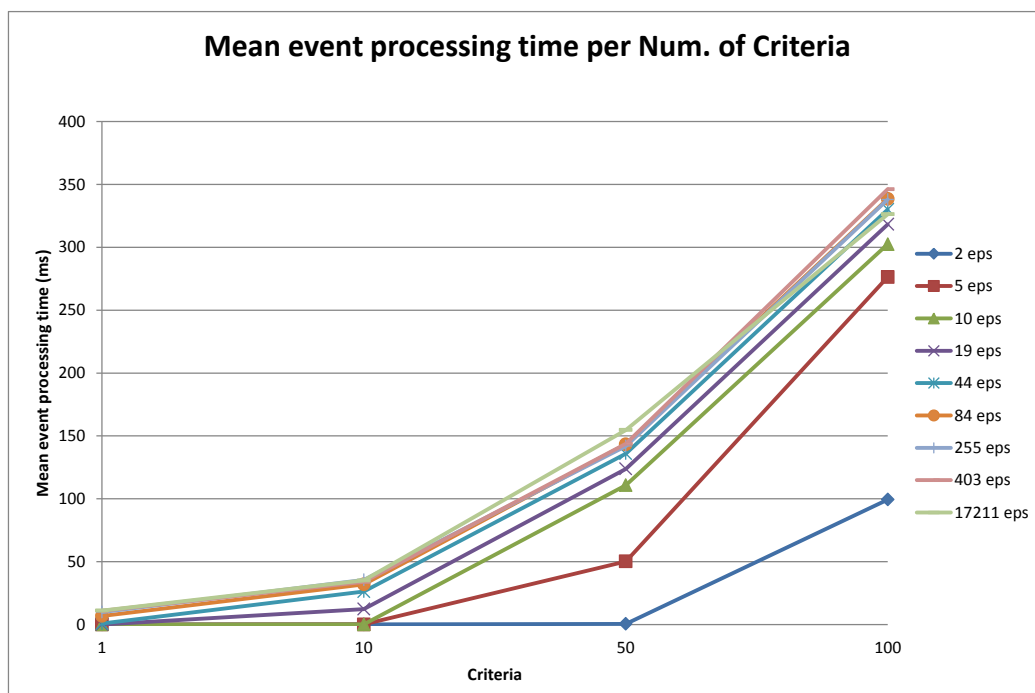
Πίνακας 10-1. Σύγκριση των μέσων χρόνων επεξεργασίας γεγονότων των δύο μεθόδων, ανά ρυθμό αποστολής και πλήθος επιλογών

Ρυθμός αποστ. (e/s)	Base-line (s)	1 επιλογή		10 επιλογές		100 επιλογές		1000 επιλογές	
		Selector	Act.Pool	Selector	Act.Pool	Selector	Act.Pool	Selector	Act.Pool
2	0.01	0.02	0.04	0.05	0.08	0.36	0.40	1013.36	869.34
5	0.01	0.02	0.04	0.05	0.08	0.56	11.33	1141.37	954.87
10	0.01	0.02	0.04	0.05	0.08	46.74	53.14	1116.98	1168.02
19	0.01	0.02	0.04	0.06	0.09	71.11	76.07	1172.39	1078.51
44	0.01	0.02	0.84	2.18	9.41	74.48	90.32	1204.32	1144.80
84	0.02	0.03	6.88	9.22	14.81	81.06	99.84	1269.64	1093.50
255	2.90	3.71	9.91	12.29	17.91	79.59	94.65	1253.21	1057.08
403	3.48	4.22	10.89	10.81	18.72	73.05	99.86	1239.43	1081.03
17.211	3.46	4.62	11.12	11.16	19.33	81.37	104.57	1199.13	40836.49

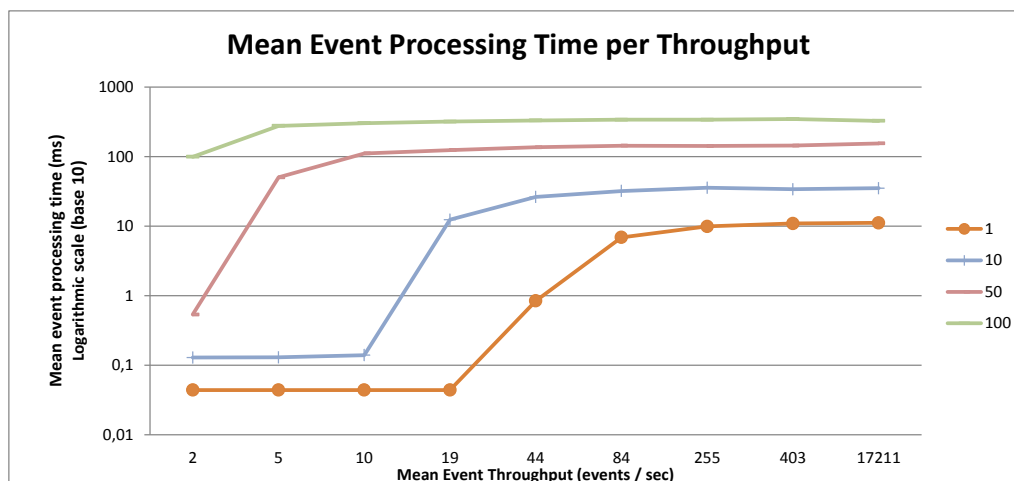
Παρατηρούμε ότι η χρήση των κόμβων κατ' επιλογή εκτέλεσης (Selectors) οδηγεί σε καλύτερες επιδόσεις απ' ότι η χρήση κόμβων αφηρημένων ενεργειών και δεξαμενών ενεργειών. Τα πειράματα με κόμβους κατ' επιλογή εκτέλεσης φαίνεται να είναι από 20% έως 150% ταχύτερα απ' ότι τα αντίστοιχα με κόμβους αφηρημένων ενεργειών. Εξαιρέση αποτελούν τα πειράματα με 1000 επιλογές όπου εκεί η χρήση κόμβων αφηρημένων ενεργειών οδηγεί σε 0% έως 15% καλύτερα αποτελέσματα. Επίσης παρατηρούμε ότι ο χρόνος επεξεργασίας των πειραμάτων με κόμβους αφηρημένων ενεργειών αρχίζει να αυξάνει ενωρίτερα απ' ότι στα πειράματα με κόμβους κατ' επιλογή εκτέλεση (μεσαία ζώνη στο Σχήμα 10-21). Τέλος, όταν ο ρυθμός αποστολής γεγονότων είναι υψηλός τότε δημιουργούνται πολλά νήματα εκτέλεσης που εκτελούνται ταυτόχρονα με αποτέλεσμα την ελάττωση των επιδόσεων.

Η χρήση των κόμβων αφηρημένων ενεργειών οδήγησε σε χαμηλότερες επιδόσεις σε σχέση με τη χρήση κόμβων κατ' επιλογή εκτέλεσης εξαιτίας της σημαντικά πολυπλοκότερης και πιο ευέλικτης μεθόδου που χρησιμοποιούν. Όπως έχει προαναφερθεί στα πειράματα με κόμβους αφηρημένων ενεργειών χρησιμοποιήθηκε η μέθοδος αναζήτησης και επιλογής LOWA, η οποία περιλαμβάνει αρκετές εργασίες όπως αντιστοίχιση των πραγματικών τιμών σε γλωσσικούς όρους, βαθμολόγηση, ταξινόμηση και φιλτράρισμα. Από την άλλη οι κόμβοι κατ' επιλογή εκτέλεσης χρησιμοποιούν απλές δομές δοκιμής και λάθους (trial and error).

Τα επόμενα διαγράμματα (Σχήμα 10-23 και Σχήμα 10-24) αφορούν τα πειράματα που χρησιμοποιούν κόμβους αφηρημένων ενεργειών και δεξαμενών ενεργειών με διαφορετικό όμως αριθμό κριτηρίων επιλογής. Απεικονίζουν το μέσο χρόνο επεξεργασίας των γεγονότων καθώς ο αριθμός των κριτηρίων επιλογής αυξάνει, για διάφορους ρυθμούς αποστολής.



Σχήμα 10-23. Μέσος χρόνος επεξεργασίας γεγονότων ανά αριθμό κριτηρίων

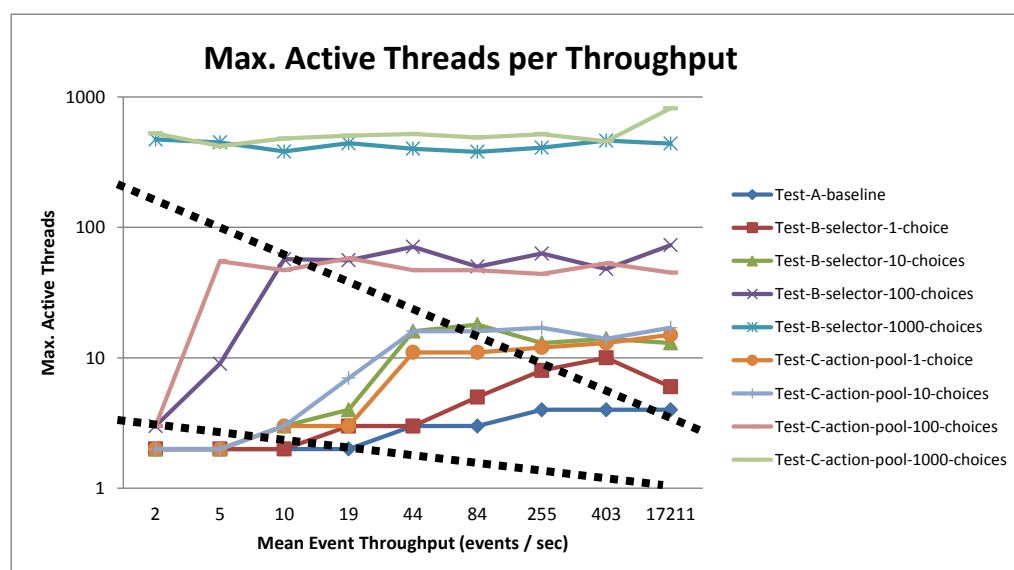


Σχήμα 10-24. Μέσος χρόνος επεξεργασίας γεγονότων ανά ρυθμό αποστολής

Παρατηρούμε μια αύξηση του μέσου χρόνου επεξεργασίας των γεγονότων καθώς αυξάνει ο αριθμός των κριτηρίων επιλογής. Ειδικότερα φαίνεται να υπάρχει μια σχέση 10 προς 1 (κατά προσέγγιση) μεταξύ των μέσων χρόνων επεξεργασίας των πειραμάτων με 100 κριτήρια επιλογής σε σχέση με αυτά με 10 κριτήρια επιλογής. Μια λίγο μικρότερη σχέση παρατηρείται μεταξύ των πειραμάτων με 10 κριτήρια επιλογής σε σχέση με αυτά με ένα μόνο κριτήριο. Η διαφορά αυτή οφείλεται στην σταθερή επιβάρυνση (overhead) εκκίνησης μιας αναζήτησης σε δεξαμενή ενεργειών. Συνεπώς συμπεραίνουμε ότι υπάρχει μια (σχεδόν) γραμμική σχέση μεταξύ του μέσου χρόνου επεξεργασίας των γεγονότων και του αριθμού των κριτηρίων επιλογής.

10.2.2.2 Μέγιστος αριθμός ενεργών στιγμιοτύπων SANs και νημάτων

Στο επόμενο διάγραμμα (Σχήμα 10-25) απεικονίζεται ο μέγιστος αριθμός στιγμιοτύπων SANs και νημάτων εκτέλεσης (στα πειράματά μας αυτά συμπίπτουν) καθώς αυξάνει ο ρυθμός αποστολής γεγονότων.

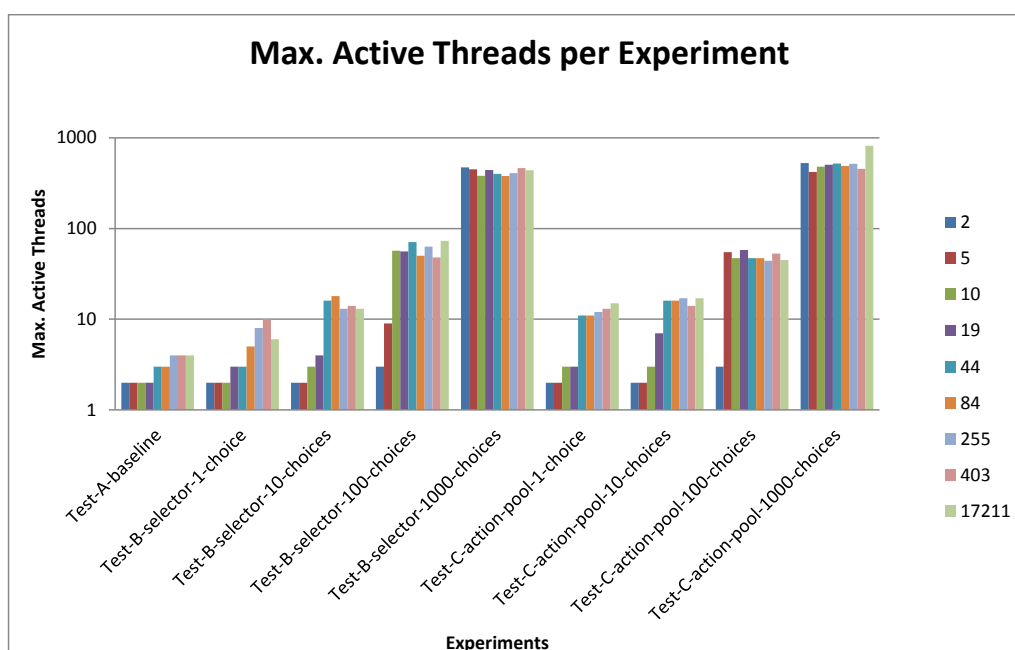


Σχήμα 10-25. Μέγιστο πλήθος ενεργών στιγμιοτύπων SANs/νημάτων ανά ρυθμό αποστολής

Το διάγραμμα δείχνει (αν και όχι τόσο ξεκάθαρα όπως στην περίπτωση του μέσου χρόνου επεξεργασίας γεγονότων) ότι οι τρεις ζώνες επιδόσεων υπάρχουν. Και πάλι στη χαμηλή ζώνη ο μέγιστος αριθμός ενεργών στιγμιότυπων/νημάτων είναι σταθερός και μικρός με τιμές από 2 έως 5. Στην μεσαία ζώνη ο αριθμός αυτός αυξάνει με εκθετικό τρόπο, καθώς αυξάνει ο ρυθμός αποστολής γεγονότων. Οι τιμές του μέγιστου αριθμού ενεργών στιγμιότυπων/νημάτων κυμαίνεται μεταξύ 10 και 100. Τέλος στην υψηλή ζώνη ο αριθμός αυτός παραμένει σχεδόν σταθερός. Οι λόγοι έχουν εξηγηθεί παραπάνω, στον αντίστοιχο σχολιασμό για τα αποτελέσματα του μέσου χρόνου επεξεργασίας γεγονότων.

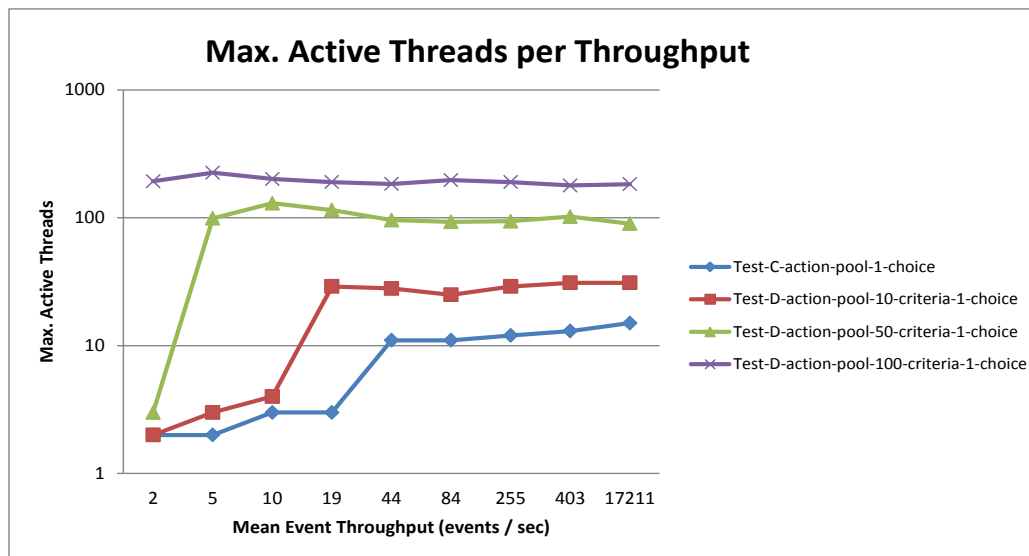
Στη συνέχεια δίνεται το ίδιο διάγραμμα αλλά τώρα στον οριζόντιο άξονα εμφανίζονται τα πειράματα (Σχήμα 10-26) αντί του ρυθμού αποστολής γεγονότων όπως στο Σχήμα 10-25. Η κλίμακα του κατακόρυφου άξονα είναι και πάλι λογαριθμική.

Ο μέγιστος αριθμός ενεργών στιγμιότυπων SANs/νημάτων φαίνεται να παραμένει χαμηλός για πειράματα με λίγες επιλογές ενώ αυξάνεται σημαντικά για τα πειράματα με 100 ή 1000 επιλογές. Αυτή η συμπεριφορά είναι αναμενόμενη διότι ο ρυθμός αποστολής γεγονότων είναι υψηλότερος από τον ρυθμό με τον οποίο ο μηχανισμός των SANs μπορεί να εκτελεί τα στιγμιότυπα των SANs (για το συγκεκριμένο υπολογιστικό σύστημα). Η ίδια συμπεριφορά διαπιστώνεται και στις δύο προσεγγίσεις (με χρήση κόμβων κατ' επιλογή εκτέλεσης και με χρήση κόμβων αφηρημένων ενεργειών).

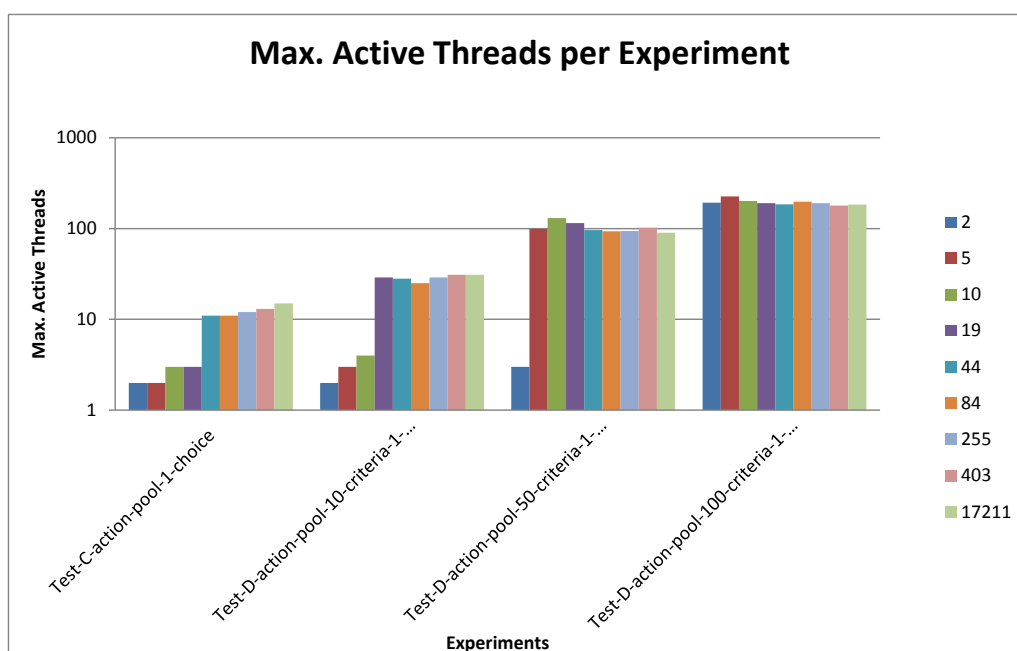


Σχήμα 10-26. Μέγιστο πλήθος ενεργών στιγμιότυπων SANs/νημάτων ανά πείραμα

Όσον αφορά τα πειράματα SAN-D-Action-Pool, δηλαδή αυτά με τον μεταβλητό αριθμό κριτηρίων επιλογής, φαίνεται ότι ο μέγιστος αριθμός ενεργών στιγμιότυπων SANs/νημάτων αυξάνεται με την αύξηση του αριθμού των κριτηρίων. Τα τρία πειράματα αντιστοιχούν σε 10, 50 και 100 κριτήρια. Για λόγους σύγκρισης στα επόμενα διαγράμματα (Σχήμα 10-27 και Σχήμα 10-28) έχουν συμπεριληφθεί και οι μετρήσεις των πειραμάτων SAN-C-Action-Pool-1-choice (χρήση κόμβων αφηρημένων ενεργειών με μία επιλογή και ένα κριτήριο).



Σχήμα 10-27. Μέγιστο πλήθος ενεργών στιγμιοτύπων SANs/νημάτων ανά ρυθμό αποστολής και ανά αριθμό κριτηρίων



Σχήμα 10-28. Μέγιστο πλήθος ενεργών στιγμιοτύπων SANs/νημάτων ανά ρυθμό αποστολής και ανά αριθμό κριτηρίων

10.2.2.3 Σχολιασμός των αποτελεσμάτων

Ανακεφαλαιώνοντας, παρατηρούμε ότι ο μέσος χρόνος επεξεργασίας των γεγονότων αυξάνεται καθώς αυξάνεται ο ρυθμός αποστολής γεγονότων ή και ο αριθμός των κριτηρίων επιλογής. Ως συνάρτηση του ρυθμού αποστολής, ο μέσος χρόνος επεξεργασίας γεγονότων φαίνεται να ακολουθεί ένα μοτίβο σιγμοειδούς καμπύλης για όλα τα πειράματα. Το σημείο καμπής (της καμπύλης) εμφανίζεται σε όλο και μικρότερους ρυθμούς αποστολής, καθώς μεγαλώνει ο αριθμός των επιλογών. Στα πειράματα με 1000 επιλογές παρατηρήθηκε ουσιαστικά το «άνω πλατό» μιας σιγμοειδούς καμπύλης που μακροσκοπικά προσομοιάζει

μια γραμμικά ανερχόμενη ευθεία. Επίσης, τα SANs με κόμβους κατ' επιλογή εκτέλεσης χρειάζονται λιγότερο χρόνο για να ολοκληρώσουν την επεξεργασία ενός γεγονότος σε σχέση με ισοδύναμα SANs που χρησιμοποιούν κόμβους αφηρημένων ενεργειών με πολυκριτηριακή μέθοδο αναζήτησης και επιλογής όπως η LOWA.

Όσον αφορά τον μέγιστο αριθμό ενεργών στιγμιότυπων SANs ή νημάτων εκτέλεσης παρατηρούμε ότι αυτός αυξάνεται με τον ρυθμό αποστολής γεγονότων και τον αριθμό των κριτηρίων επιλογής. Ο μέγιστος αριθμός ενεργών στιγμιότυπων ή νημάτων αντικατοπτρίζει τον μεγαλύτερο επεξεργαστικό φόρτο που υφίσταται ο μηχανισμός των SANs και κατ' επέκταση το σύστημα στη διάρκεια ενός πειράματος.

Σε υψηλούς ρυθμούς αποστολής γεγονότων (χιλιάδες γεγονότα ανά δευτερόλεπτο) δημιουργούνται και εκτελούνται ταυτόχρονα πολλά νήματα εκτέλεσης. Αυτό οδηγεί σε συνθήκες έντονου ανταγωνισμού μεταξύ τους για τη χρήση κοινών πόρων του συστήματος με αποτέλεσμα τη σημαντική μείωση των επιδόσεων του μηχανισμού των SANs. Ακόμη και εναλλακτικές υλοποιήσεις του μηχανισμού των SANs (με βελτιστοποιήσεις), αν εκτελούνταν σε υπολογιστικό σύστημα με έναν επεξεργαστή, θα αντιμετώπιζαν μείωση των επιδόσεων. Με χρήση όμως στρατηγικών κατανεμημένης εκτέλεσης μπορεί να αποσοβηθεί το πρόβλημα της μείωσης της επίδοσης (δηλαδή η αύξηση του μέσου χρόνου επεξεργασίας) διότι ο επεξεργαστικός φόρτος θα κατανέμεται σε ένα δίκτυο κόμβων επεξεργασίας. Ωστόσο σε αυτήν την περίπτωση η χρήση των πρωτοκόλλων επικοινωνίας και ο φόρτος του δικτύου ενδεχομένως να εισήγαγαν καθυστερήσεις στην κατανομή των γεγονότων στους κόμβους επεξεργασίας ή στο συγχρονισμό μεταξύ τους.

Συμπερασματικά, οι κόμβοι κατ' επιλογή εκτέλεσης (Selector nodes) επιτυγχάνουν καλύτερες επιδόσεις, σε όρους χρόνου επεξεργασίας και μέγιστου αριθμού ενεργών νημάτων, σε σχέση με τους κόμβους αφηρημένων ενεργειών με χρήση δεξαμενών ενεργειών και πολυκριτηριακών μεθόδων αναζήτησης και επιλογής (όπως η LOWA). Συγκεκριμένα, για μικρούς ή μεσαίους ρυθμούς αποστολής γεγονότων (50 – 100 γεγονότα ανά δευτερόλεπτο) η διαφορά τους είναι μικρή ή και μη παρατηρήσιμη. Ωστόσο σε μεγάλους ρυθμούς αποστολής η διαφορά αυτή γίνεται εμφανής και σε αρκετές περιπτώσεις σημαντικά μεγάλη. Οι κόμβοι κατ' επιλογή εκτέλεσης είναι απλές δομές με απλή λειτουργική σημασιολογία (execution semantics). Αυτό οδηγεί σε βελτιωμένες επιδόσεις σε σχέση με τους κόμβους αφηρημένων ενεργειών. Επιπλέον η μοντελοποίηση των επιλογών (choice selection) είναι σχετικά εύκολη και απλή εργασία. Μολαταύτα η μοντελοποίηση με κόμβους κατ' επιλογή εκτέλεσης προϋποθέτει ότι όλες οι εναλλακτικές ενέργειες είναι γνωστές εκ των προτέρων κατά τον σχεδιασμό των SANs. Ωστόσο αυτό δεν είναι πάντα εφικτό (δείτε παράδειγμα 10.1) ενώ εξάλλου μπορεί να απαιτείται ευελιξία κατά το χρόνο εκτέλεσης. Επιπλέον, είναι προφανής η μεγαλύτερη σχεδιαστική προσπάθεια που απαιτείται για τη μοντελοποίηση SANs με πολλές (π.χ. 1000) επιλογές και χρήση ενός κόμβου κατ' επιλογή εκτέλεσης σε σχέση με την σχεδιαστική προσπάθεια που απαιτείται με χρήση ενός κόμβου αφηρημένης ενέργειας και της αντίστοιχης δεξαμενής ενεργειών. Οι κόμβοι αφηρημένων ενεργειών χρησιμοποιούν πιο προηγμένους τρόπους και στρατηγικές αναζήτησης και επιλογής απ' ό,τι οι κόμβοι κατ' επιλογή εκτέλεσης, παρέχοντας έτσι ένα ευέλικτο και προσαρμόσιμο εργαλείο λήψης αποφάσεων. Οι κόμβοι κατ' επιλογή εκτέλεσης

μπορεί να αποτελέσουν και αιτίες αποτυχίας αν καμιά από τις επιλογές δεν ταιριάζει. Αντίθετα οι μέθοδοι αναζήτησης και επιλογής των κόμβων αφηρημένων ενεργειών μπορεί να ρυθμιστούν έτσι ώστε να δίνουν πάντοτε μια επιλογή (έστω και αν δεν ταιριάζει εντελώς) κι έτσι να αποφεύγονται οι αποτυχίες. Κλείνοντας αυτήν την αξιολόγηση σημειώνουμε ότι σε περιπτώσεις που είναι εκ των προτέρων γνωστές όλες οι επιθυμητές επιλογές, είναι ευθύνη του σχεδιαστή των SANs να εκτιμήσει τα πλεονεκτήματα και τα μειονεκτήματα της κάθε προσέγγισης προκειμένου να χρησιμοποιήσει την πλέον κατάλληλη. Όταν όμως οι επιλογές δεν είναι γνωστές κατά τον σχεδιασμό ή δεν μπορούν να απαριθμηθούν ή υπάρχει ανάγκη για επικαιροποίησή τους κατά την εκτέλεση, τότε η χρήση κόμβων αφηρημένων ενεργειών και δεξαμενών ενεργειών είναι επιβεβλημένη.

11 Συμπεράσματα και Περαιτέρω Έρευνα

Στο κεφάλαιο αυτό συνοψίζονται τα συμπεράσματα και η συνεισφορά της διατριβής. Επίσης, επισημαίνονται περιορισμοί της έρευνας που πραγματοποιήθηκε καθώς και πιθανές επεκτάσεις αυτής. Τέλος κλείνουμε κάνοντας μια αναφορά σε κατευθύνσεις για μελλοντική έρευνα.

11.1 Συμπεράσματα

Στην παρούσα διατριβή παρουσιάστηκε μια νέα προσέγγιση για οδηγούμενη από γεγονότα προσαρμογή επιχειρησιακών διαδικασιών (event-driven business process adaptation), η οποία απαντά στα ερευνητικά ερωτήματα που τέθηκαν στο κεφάλαιο 4. Η προσέγγιση αυτή καλύπτει το μεγαλύτερο μέρος του κύκλου ζωής προσαρμογών, από την ανίχνευση της ανάγκης για προσαρμογή, την επιλογή και πρόταση μιας προσαρμογής και τέλος την εφαρμογή αυτής σε διαδικασίες αποτυπωμένες σε BPMN 2.

Από τη βιβλιογραφική μελέτη στα πεδία της προσαρμογής επιχειρησιακών διαδικασιών, επίγνωσης κατάστασης και επεξεργασίας γεγονότων, αλλά και από την εξέταση διαφόρων μεθόδων και τεχνολογιών που κρίθηκαν σχετικές, αντλήθηκαν ιδέες για τη σχεδίαση και ανάπτυξη ενός πλαισίου εργασίας για αντιμετώπιση του προβλήματος. Ειδικότερα, το ενδιαφέρον της διατριβής εστίασε στην ανίχνευση με τη βοήθεια γεγονότων, καταστάσεων που δημιουργούν την ανάγκη προσαρμογής των διαδικασιών, την επιλογή και πρόταση κατάλληλων προσαρμογών, διατηρώντας τους στόχους των διαδικασιών και τέλος στην εφαρμογή των προτεινόμενων προσαρμογών σε διαδικασίες σε BPMN 2 με τρόπο που δεν απαιτεί την προηγούμενη προετοιμασία τους. Ακόμη, στο πλαίσιο της διατριβής αναπτύχθηκαν και δοκιμάστηκαν εργαλεία λογισμικού που υλοποιούν το παραπάνω πλαίσιο εργασίας και τα επιμέρους στοιχεία του.

Η συνεισφορά της διατριβής εντοπίζεται και στα τρία βασικά μέρη του προτεινόμενου πλαισίου εργασίας, τα οποία αντιστοιχούν στα τρία ερευνητικά ερωτήματα της διατριβής. Συγκεκριμένα, παρουσιάστηκαν τα Δίκτυα Κατάστασης – Ενέργειας (SANs), μια νέα μέθοδος αποτύπωσης της συσχέτισης των στόχων και λειτουργιών των διαδικασιών με τις καταστάσεις που τα επηρεάζουν, καθώς και αποτύπωσης του τρόπου ανταπόκρισης σε αυτές. Τα SANs μοντελοποιούν την ανάλυση των στόχων σε υποστόχους αφού κι αυτοί

επηρεάζονται από την εκάστοτε επικρατούσα κατάσταση και συμμετέχουν φυσικά στην επίτευξη των βασικών στόχων των διαδικασιών. Τα SANs απαντούν στο πρώτο ερευνητικό ερώτημα, δηλαδή στην ανίχνευση της ανάγκης για προσαρμογή, με μοντελοποίηση του τρόπου απόκτησης επίγνωσης της κατάστασης από γεγονότα. Επίσης απαντούν εν μέρει και στο δεύτερο ερευνητικό ερώτημα, δηλαδή την επιλογή κατάλληλης προσαρμογής με βάση την αποκτηθείσα επίγνωση κατάστασης. Η προσαρμογή εκφράζεται ως ένα θέμα προσαρμογής (adaptation aspect) το οποίο προτείνεται στον υπεύθυνο της διαδικασίας.

Επιπλέον προτάθηκε μια επέκταση των SANs που επιτρέπει την αφαιρετική αποτύπωση ενεργειών ή υποστόχων, γεγονός που παρέχει ευελιξία κατά τον σχεδιασμό (αφού δεν απαιτείται αποτύπωση όλων των λεπτομερειών που ενδεχομένως να μην είναι γνωστές). Κατά την εκτέλεση οι αφηρημένες ενέργειες και υποστόχοι συγκεκριμενοποιούνται με κάποια μέθοδο που έχει προσδιοριστεί από τον σχεδιασμό. Στη διατριβή αυτή προτείνεται η χρήση δεξαμενών ενεργειών για επιλογή συγκεκριμένων υλοποιήσεων των αφηρημένων ενεργειών από ένα σύνολο εναλλακτικών. Η επέκταση αυτή συμπληρώνει την απάντηση στο δεύτερο ερευνητικό ερώτημα, προσθέτοντας τη δυνατότητα δυναμικής επιλογής των προτάσεων προσαρμογών.

Παρουσιάστηκε ακόμη η μέθοδος FlexiBPMN2.0 για θεματοστρεφή προσαρμογή διαδικασιών σε BPMN 2.0. Η μέθοδος δεν προϋποθέτει την προετοιμασία των διαδικασιών προκειμένου αυτές να δεχτούν τις προσαρμογές παρά μόνο έναν κατάλληλο μηχανισμό εκτέλεσης. Η μέθοδος αυτή απαντά στο τρίτο ερευνητικό ερώτημα, δηλαδή στο πώς εφαρμόζονται οι προτεινόμενες προσαρμογές σε διαδικασίες BPMN 2.

Τα παραπάνω συνοψίζονται ακολούθως (Πίνακας 11-1), όπου τα ερευνητικά ερωτήματα και οι παράμετροί τους παρατίθενται με τις αντίστοιχες προτάσεις της διατριβής.

Πίνακας 11-1. Ερευνητικά ερωτήματα, παράμετροι και προτάσεις της διατριβής

EE-1. Ανίχνευση της ανάγκης προσαρμογής με χρήση γεγονότων	
<i>Επιμέρους Παράμετροι</i>	<i>Πρόταση της Διατριβής</i>
<ul style="list-style-type: none"> • Πώς αξιοποιούνται τα γεγονότα; • Πώς επιτυγχάνεται η επίγνωση κατάστασης; • Πώς μοντελοποιείται η επίγνωση της κατάστασης και της ανάγκης προσαρμογής; • Πότε υπάρχει ανάγκη προσαρμογής; 	<ul style="list-style-type: none"> • Στους κόμβους κατάστασης των SANs, σηματοδοτώντας αλλαγές καταστάσεων • Με γεγονότα, contextualization σε συνδυασμό με την ανάλυση σε στόχους • Με SANs που καταγράφουν τον τρόπο επίγνωσης κατάστασης ως ιεραρχίες στόχων σε σχέση με καταστάσεις • Όταν τα SANs ανιχνεύσουν την αντίστοιχη κατάσταση. <p>Τα SANs είναι το μέσο μοντελοποίησης της γνώσης των εμπειρογνομόνων τόσο για την επίγνωση της κατάστασης όσο και της διαπίστωσης της ανάγκης προσαρμογής</p>

EE-2. Προσαρμογή επιχειρησιακών διαδικασιών οδηγούμενη από την επίγνωση κατάστασης	
<i>Επιμέρους Παράμετροι</i>	<i>Πρόταση της Διατριβής</i>
<ul style="list-style-type: none"> • Πώς γίνεται η επιλογή προσαρμογών με βάση μια δεδομένη ανάγκη ή κατάσταση; 	<ul style="list-style-type: none"> • Είτε με πλήρως καθορισμένα και γνωστά (απ' τον σχεδιασμό), "στατικά" SANS, • είτε με μερικώς προσδιορισμένα (κατά το σχεδιασμό) SANS, με χρήση κόμβων αφηρημένων ενεργειών και δεξαμενών ενεργειών. Αυτά συμπληρώνονται δυναμικά κατά την εκτέλεση
<ul style="list-style-type: none"> • Πώς μοντελοποιείται η διαδικασία αυτή; 	<ul style="list-style-type: none"> • Ως SANS με ή χωρίς κόμβους αφηρημένων ενεργειών και δεξαμενές ενεργειών, καθώς και ως θέμα προσαρμογών.
EE-3. Εφαρμογή προσαρμογών	
<i>Επιμέρους Παράμετροι</i>	<i>Πρόταση της Διατριβής</i>
<ul style="list-style-type: none"> • Πώς εφαρμόζονται οι προσαρμογές στις διαδικασίες; • Απαιτείται a priori προετοιμασία των διαδικασιών; • Πώς καταγράφονται οι προσαρμογές; 	<ul style="list-style-type: none"> • Πρώτα προτείνονται στον υπεύθυνο της διαδικασίας. Αν γίνει δεκτή η πρόταση εφαρμόζονται με την τεχνική FlexiBPMN2.0, μια θεματοστρεφή τεχνική • Όχι. Μόνο ένας μηχανισμός προσαρμοστικής εκτέλεσης (APES) • Ως μέρη διαδικασιών σε BPMN 2 (συμβουλές των θεμάτων προσαρμογής)

Κλείνοντας μπορούμε να πούμε ότι το προτεινόμενο πλαίσιο εργασίας και τα μεθοδολογικά εργαλεία που περιλαμβάνει απαντούν στο πρόβλημα της οδηγούμενης από γεγονότα προσαρμογής διαδικασιών όπως αυτό καταγράφεται στον κύκλο ζωής προσαρμογών.

11.2 Περιορισμοί και πιθανές επεκτάσεις

Κατά την αντιμετώπιση των ερευνητικών ερωτημάτων που τέθηκαν στην διατριβή ήταν αναγκαία η χρήση μιας σειράς παραδοχών και συμβάσεων που οδήγησαν σε αντίστοιχους περιορισμούς. Οι περιορισμοί αυτοί επέτρεψαν την εστίαση και εμβάθυνση της έρευνας σε συγκεκριμένες μεθόδους ή τεχνικές αλλά και στην υλοποίηση των αντίστοιχων συστημάτων λογισμικού.

Η επίγνωση κατάστασης και κατ' επέκταση η ανίχνευση της ανάγκης προσαρμογής γίνεται μόνο με επεξεργασία γεγονότων και με χρήση μιας τεχνικής CEP που βασίζεται σε πρότυπα σύνθετων γεγονότων (CEPats). Υπάρχει όμως η δυνατότητα επέκτασης των κόμβων κατάστασης των SANS έτσι ώστε να μπορούν να χρησιμοποιούν και άλλες εναλλακτικές μεθόδους επεξεργασίας γεγονότων. Περαιτέρω, θα μπορούσαν να χρησιμοποιηθούν και μέθοδοι επίγνωσης κατάστασης που δε βασίζονται στη χρήση γεγονότων (για παράδειγμα χρήση κάποιου μηχανισμού παρακολούθησης, ανάλυση αρχείων καταγραφών και άλλα).

Οι προσαρμογές που προτείνει το πλαίσιο εργασίας επιλέγονται από ένα σύνολο (προδιαγεγραμμένων) εναλλακτικών με χρήση δύο πολυκριτηριακών τεχνικών ταξινόμησης καθώς και τη χρήση των μεταδεδομένων κάθε εναλλακτικής. Ωστόσο το πλαίσιο επιτρέπει την εισαγωγή επιπλέον μεθόδων προς τρεις μάλιστα κατευθύνσεις. Πρώτον τη χρήση εναλλακτικών τεχνικών πολυκριτηριακής ταξινόμησης (για παράδειγμα AHP ή Topsis). Δεύτερον την εισαγωγή τεχνικών επιλογής από ένα σύνολο εναλλακτικών που δεν βασίζεται όμως στην πολυκριτηριακή ταξινόμηση (για παράδειγμα χρήση κάποια μεθόδου υπεροχής όπως οι μέθοδοι της οικογένειας Electre). Τρίτον τη χρήση μεθόδων σύνθεσης νέων προσαρμογών ή την αναζήτηση και επαναχρησιμοποίηση προηγούμενων προσαρμογών με βάση την ομοιότητα των καταστάσεων και διαδικασιών ή και άλλες.

Ακόμη η έρευνά μας εστίασε σε διορθωτικές προσαρμογές που θεραπεύουν προβλήματα που έχουν ήδη εμφανιστεί ή επείκονται να εμφανιστούν. Ωστόσο προσαρμογές μπορεί να γίνουν και για βελτίωση των διαδικασιών, όταν φυσικά δοθούν οι κατάλληλες ευκαιρίες. Κάτι τέτοιο όμως θα απαιτούσε διαφορετικές τεχνικές αναζήτησης και επιλογής, ικανές να λαμβάνουν υπόψη και τυχόν νέες ιδιότητες ή χαρακτηριστικά που ενδεχομένως εισάγουν οι εκάστοτε ευκαιρίες για βελτίωση.

Όσον αφορά την εφαρμογή των προσαρμογών, στη διατριβή προτείνεται η μέθοδος FlexiBPMN2.0, η οποία βασίζεται στο θεματοστρεφή προγραμματισμό για την τροποποίηση των διαδικασιών χωρίς να απαιτείται η εκ των προτέρων προετοιμασία αυτών. Όμως το προτεινόμενο πλαίσιο εργασίας επιτρέπει τη χρήση και οποιασδήποτε άλλης ισοδύναμης τεχνικής, πραγματοποιώντας φυσικά τις αντίστοιχες αλλαγές στις μεθόδους επιλογής ή παραγωγής των προσαρμογών. Η μέθοδος FlexiBPMN2.0 που προτείνεται στη διατριβή αφορά διαδικασίες αποτυπωμένες σε BPMN 2.0. Ωστόσο εύκολα θα μπορούσε να τροποποιηθεί έτσι ώστε να υποστηρίζει διαδικασίες σε BPEL ή σε άλλες γλώσσες και σημειογραφίες.

Αναφέρουμε τέλος ότι επεκτάσεις όπως αυτές που παρουσιάστηκαν παραπάνω, απαιτούν την αντίστοιχη τροποποίηση και επέκταση των συστημάτων λογισμικού που υλοποιούν το προτεινόμενο πλαίσιο εργασίας.

11.3 Περαιτέρω έρευνα

Στην ενότητα αυτή παρουσιάζονται ορισμένες προτάσεις για περαιτέρω έρευνα βασισμένη σε αυτή τη διατριβή, τις οποίες θεωρούμε ενδιαφέρουσες. Κατ' αρχήν η βασική συνεισφορά της διατριβής, τα Δίκτυα Κατάστασης – Ενέργειας, θα μπορούσαν να επεκταθούν και εμπλουτιστούν με πολλούς τρόπους, τόσο σε μεθοδολογικό επίπεδο, όσο και στη σημειογραφία τους αλλά και στην υλοποίησή τους. Ενδεικτικά αναφέρουμε τα ακόλουθα:

- Προσθήκη διαφόρων τύπων στόχων (προς επίτευξη, προς αποφυγή) και τρόπου τυπικού καθορισμού των παραμέτρων που τους επηρεάζουν.
- Προσθήκη μηχανισμού ανατροφοδότησης των SANs ώστε να υπάρχει επίγνωση και για το αν μια προσαρμογή ήταν επιτυχής και σε ποιο βαθμό.

- Προσθήκη κόμβων «σύνθετων» καταστάσεων και «αφηρημένων» καταστάσεων έτσι ώστε να είναι δυνατός ο συνδυασμός απλούστερων καταστάσεων σε πιο σύνθετες καθώς και να είναι δυνατή η αποτύπωση μη επαρκώς γνωστών καταστάσεων κατά το σχεδιασμό.
- Ανάπτυξη ενός τρόπου τυπικής (formal) αποτύπωσης του πλαισίου εκτέλεσης των SANs.
- Εισαγωγή νέων κόμβων ενεργειών και decorators που αφενός θα εμπλουτίσουν τη σημειογραφία με πιο εξειδικευμένες δυνατότητες και αφετέρου θα απλοποιήσουν ορισμένα συχνά απαντώμενα μοτίβα χρήσης.

Μια άλλη κατεύθυνση θα μπορούσε να αφορά τη διερεύνηση της δυνατότητας μετατροπής των SANs σε άλλες σημειογραφίες ή γλώσσες (όπως τα HTNs, η BPMN ή άλλες) καθώς και αντίστροφα από αυτές σε SANs. Ακόμη, ανάπτυξη και χρήση εναλλακτικών τρόπων απεικόνισης και οπτικοποίησης των SANs πέραν της δενδρικής απεικόνισης.

Όσον αφορά τους κόμβους αφηρημένων ενεργειών, δηλαδή τη δεύτερη καινοτομία της διατριβής. Εδώ θα μπορούσαν να εισαχθούν νέες μέθοδοι συγκεκριμενοποίησής τους κατά την εκτέλεση, πέραν της χρήσης των δεξαμενών ενεργειών. Για παράδειγμα:

- Χρήση νέων αλγορίθμων για αναζήτηση και επιλογή προσαρμογών με τεχνικές πέραν της πολυκριτηριακής ταξινόμησης που προτάθηκαν (όπως ταξινόμηση με μεθόδους υπεροχής, επιλογή με μεθόδους ελέγχου ομοιότητας ή άλλες).
- Εισαγωγή αλγορίθμων για σύνθεση προτάσεων προσαρμογών (για παράδειγμα βασισμένων σε τεχνικές σχεδιασμού και ιεραρχικών δικτύων εργασιών).

Όσον αφορά την υλοποίηση των SANs και του προτεινόμενου πλαισίου, θα μπορούσαν να πραγματοποιηθούν ορισμένες βελτιώσεις ώστε αυτή να καταστεί πιο εύχρηστη, ευρύτερης εφαρμογής και πιο ελκυστική ως προϊόν. Φερ' ειπείν θα μπορούσε να διερευνηθεί η υλοποίηση της πλατφόρμας για κινητά τηλέφωνα (απαιτούν έναν ελαφρύ και οικονομικό SAN engine), ή για λειτουργία σε περιβάλλον υπολογιστικής νέφους⁶⁸ (Cloud Computing). Η μεν πρώτη θα επιτρέψει τη χρήση των SANs σε μια ευρεία βάση χρηστών και τη συλλογή γεγονότων και καταστάσεων απευθείας από τους χρήστες. Η δεύτερη μπορεί να καταστήσει τα SANs μια φθηνή, γρήγορη και ελκυστική λύση.

Μια άλλη ενδιαφέρουσα βελτίωση είναι η αλλαγή της αρχιτεκτονικής της πλατφόρμας προς μια κατακευματισμένη λογική ή λογική ομότιμων δικτύων (Peer-to-Peer networks). Αυτή μπορεί να αφορά είτε μοιραζόμενα (σε πολλούς κόμβους ενός δικτύου) αποθετήρια (repositories) SANs, πλαίσια εκτέλεσης ή άλλους πόρους, είτε την κατανομή των προς εκτέλεση SANs σε πολλούς κόμβους (load balancing), είτε ακόμη την πλήρως κατακευματισμένη εκτέλεση των στιγμιότυπων των SANs (δηλαδή ένα στιγμιότυπο μπορεί να εκτελείται ταυτόχρονα σε πολλούς κόμβους ή να μεταφέρεται η εκτέλεσή του από κόμβο σε κόμβο).

⁶⁸ Η υπολογιστική νέφος (cloud computing) είναι ένα μοντέλο επεξεργασίας που επιτρέπει την ευέλικτη, κατ' απαίτηση, (δικτυακή) πρόσβαση σε ένα κοινόχρηστο σύνολο παραμετροποιήσιμων υπολογιστικών πόρων όπως δίκτυα, εξυπηρετητές, αποθηκευτικοί χώροι, εφαρμογές και υπηρεσίες. Αυτοί μπορούν να τροφοδοτηθούν γρήγορα και να διατεθούν με ελάχιστη προσπάθεια διαχείρισης ή αλληλεπίδρασης με τον πάροχο της υπηρεσίας. [Πηγή: <http://www.nist.gov/itl/cloud/upload/cloud-def-v15.pdf>]

Άλλη βελτίωση αφορά την υλοποίηση της δυνατότητας εξυπηρέτησης πολλαπλών χρηστών ή εφαρμογών από την ίδια εγκατάσταση της πλατφόρμας (multi-tenancy), γεγονός που θα επέτρεπε στα SANs να λειτουργούν και σε περιβάλλον παρόχου εφαρμογών (application provider). Επίσης βελτιώσεις μπορούν να γίνουν και στη βασική υλοποίηση του μηχανισμού εκτέλεσης SANs, με χρήση ενός πιο έξυπνου μοντέλου πολυεπεξεργασίας (multithreading model), προσθήκη επιπλέον αποτιμητών (evaluators) και contextualizers, εμπλουτισμό της μονάδας λειτουργικών ευκολιών με χρήσιμες λειτουργίες όπως για πρόσβαση σε Βάσεις Δεδομένων. Βελτιώσεις θα μπορούσαν να γίνουν και στο περιβάλλον επεξεργασίας των SANs προκειμένου να γίνει πιο εύχρηστο και ελκυστικότερο, ή ακόμη να αναπτυχθεί ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) βασισμένο πιθανώς στο Eclipse. Ακόμη, ένα καλύτερο, γραφικό περιβάλλον διαχείρισης και εκτέλεσης των SANs, με χρήση ίσως μιας web-based διεπαφής, επίσης θα βελτίωνε την εμπειρία χρήσης τους.

11.4 Επίλογος

Στην παρούσα διατριβή εξετάστηκε η οδηγούμενη από γεγονότα προσαρμογή επιχειρησιακών διαδικασιών (event-driven business process adaptation). Αναπτύχθηκε ένα πλαίσιο εργασίας που βασίζεται στα SANs, για την ανίχνευση των αναγκών προσαρμογής διαδικασιών, την επιλογή και πρόταση προσαρμογών από δεξαμενές ενεργειών καθώς και για την εφαρμογή αυτών σε διαδικασίες σε BPMN 2.0 με χρήση μιας θεματοστρεφούς μεθόδου (FlexiBPMN2.0).

Το πλαίσιο αυτό δοκιμάστηκε με μια σειρά από περιπτώσεις χρήσης όπου επαληθεύτηκε η καταλληλότητα και εφαρμοσιμότητά του σε πραγματικά προβλήματα, ενώ αποδείχθηκε και η οικονομία που επιτυγχάνει όσων αφορά τη διακίνηση και «κατανάλωση» γεγονότων σε σχέση με προϋπάρχουσες τεχνολογίες. Επιπλέον πραγματοποιήθηκαν πειράματα τα οποία επέτρεψαν τη μελέτη της υλοποίησης του πλαισίου.

Κλείνοντας την παρούσα διατριβή υπογραμμίζουμε ότι η σημασία των επιχειρησιακών διαδικασιών στις σύγχρονες επιχειρήσεις και οργανισμούς γίνεται ολοένα και πιο ξεκάθαρη ενώ και η αντίστοιχη βιομηχανία παρουσιάζει σημαντική ανάπτυξη. Έτσι μέθοδοι και συστήματα για σχεδιασμό, διαχείριση και εκτέλεση των επιχειρησιακών διαδικασιών θα χρησιμοποιούνται συχνότερα στην πράξη και θα δοκιμάζονται σε πραγματικές καταστάσεις και προκλήσεις. Το γεγονός αυτό μας επιτρέπει να πούμε ότι στο μέλλον θα γίνει επιτακτική η ανάγκη για εξεύρεση ολοκληρωμένων τρόπων διαχείρισης των διαδικασιών (που θα περιλαμβάνουν και την προσαρμογή), ικανών να επιλύουν συνολικά τα εκάστοτε προβλήματα που ανακύπτουν κι όχι συγκεκριμένα μόνο μέρη τους. Η παρούσα διατριβή είναι ένα μικρό βήμα προς την κατεύθυνση αυτή.

Δημοσιεύσεις και Ανακοινώσεις

Ο Ι. Πατινωτάκης έχει πραγματοποιήσει 14 δημοσιεύσεις σε επιστημονικά περιοδικά και ανακοινώσεις σε διεθνή επιστημονικά συνέδρια, οι οποίες έχουν λάβει 48 αναφορές εκ των οποίων οι 37 είναι ετεροαναφορές. Ο δείκτης h-index του Ι. Πατινωτάκη είναι 5 και έχει υπολογιστεί με την εφαρμογή Publish or Perish (21/05/2015).

Άρθρα σε Περιοδικά

- [J1] Patiniotakis, I., N. Papageorgiou, Y. Verginadis, D. Apostolou, and G. Mentzas. 2013a. "Dynamic Event Subscriptions in Distributed Event Based Architectures." *Expert Systems with Applications* 40 (6): 1935–1946.
doi: <http://dx.doi.org/10.1016/j.eswa.2012.10.005> (5-Year Impact Factor: 2.254)
- [J2] Patiniotakis, I., D. Apostolou, and G. Mentzas. 2012. "Unified Collaborative Innovation Framework." *International Journal of Learning and Intellectual Capital* 9 (3): 260–275. doi: <http://dx.doi.org/10.1504/IJLIC.2012.047287>.
- [J3] Patiniotakis, I., D. Apostolou, and G. Mentzas. 2011. "Fuzzy UTASTAR: A Method for Discovering Utility Functions from Fuzzy Data." *Expert Systems with Applications* 38 (12): 15463–15474. doi: <http://dx.doi.org/10.1016/j.eswa.2011.06.014>.

Άρθρα σε Πρακτικά Συνεδρίων

- [C1] Verginadis, Y., I. Patiniotakis, N. Papageorgiou, D. Apostolou, G. Mentzas, and N. Stojanovic. 2015. "Context Management in Event Marketplaces." In *The Semantic Web: ESWC 2012 Satellite Events*, edited by E. Simperl, B. Norton, D. Mladenic, E. Della Valle, I. Fundulaki, A. Passant and R. Troncy, LNCS 7540: 313–326. Springer Berlin Heidelberg. doi: http://dx.doi.org/10.1007/978-3-662-46641-4_24.
- [C2] Patiniotakis, I., Y. Verginadis, and G. Mentzas. 2014. "Preference-based cloud service recommendation as a brokerage service." In *Proceedings of the 2nd International Workshop on CrossCloud Systems (CCB 2014) (at 15th International Middleware Conference)*, 1–5. Bordeaux, France: ACM New York, NY, USA. ISBN: 978-1-4503-3233-0. doi: <http://dx.doi.org/10.1145/2676662.2676677>.

- [C3] Veloudis, S., I. Paraskakis, A. Friesen, Y. Verginadis, I. Patiniotakis, and A. Rossini. 2014. "Continuous Quality Assurance and Optimisation in Cloud-Based Virtual Enterprises." In *Collaborative Systems for Smart Networked Environments*, edited by L.M. Camarinha-Matos, and H. Afsarmanesh, 434:621–632. Amsterdam, Netherlands: Springer Berlin Heidelberg. doi: http://dx.doi.org/10.1007/978-3-662-44745-1_61.
- [C4] Veloudis, S., A. Friesen, I. Paraskakis, Y. Verginadis, and I. Patiniotakis. 2014. "Underpinning a Cloud Brokerage Service Framework for Quality Assurance and Optimization." In *Proceedings of IEEE 6th International Conference on Cloud Computing Technology and Science (CloudCom 2014)*, 660–663. Singapore: IEEE. doi: <http://dx.doi.org/10.1109/CloudCom.2014.146>.
- [C5] Patiniotakis, I., N. Papageorgiou, Y. Verginadis, D. Apostolou, and G. Mentzas. 2013b. "A Framework for Situation-Aware Adaptation of Service-Based Applications." In *Adaptive Web Services for Modular and Reusable Software Development: Tactics and Solutions*, edited by G. Ortiz and J. Cubo, 253–262. IGI Global. doi: <http://dx.doi.org/10.4018/978-1-4666-2089-6.ch010>.
- [C6] Patiniotakis, I., N. Papageorgiou, D. Apostolou, Y. Verginadis, and G. Mentzas. 2013. "Collaborative Process Flexibility Using Multi-Criteria Decision Making." In *Proceedings of: 14th IFIP Working Conference on Virtual Enterprise – Special Session on Event-Driven Collaborative Network*. IFIP Advances in Information and Communication Technology. 408:691–698. Dresden, Germany. doi: http://dx.doi.org/10.1007/978-3-642-40543-3_72
- [C7] Patiniotakis, I., N. Papageorgiou, Y. Verginadis, D. Apostolou, and G. Mentzas. 2013c. "An Aspect Oriented Approach for Implementing Situational Driven Adaptation of BPMN2.0 Workflows." In *Business Process Management Workshops 2012*, edited by M. La Rosa and P. Soffer, LNBI 132:414–425. Tallinn, Estonia: Springer Berlin Heidelberg. doi: http://dx.doi.org/10.1007/978-3-642-36285-9_44.
- [C8] Patiniotakis, I., S. Rizou, Y. Verginadis, and G. Mentzas. 2013. "Managing Imprecise Criteria in Cloud Service Ranking with a Fuzzy Multi-Criteria Decision Making Method." In *Service-Oriented and Cloud Computing*, edited by K.K. Lau, W. Lamersdorf, and E. Pimentel, LNCS 8135:34–48. Malaga, Spain: Springer Berlin Heidelberg. doi: http://dx.doi.org/10.1007/978-3-642-40651-5_4.
- [C9] Verginadis, Y., N. Papageorgiou, I. Patiniotakis, D. Apostolou, and G. Mentzas. 2012. "A Goal Driven Dynamic Event Subscription Approach." In *Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems (DEBS 2012)*, edited by F. Bry, A. Paschke, P.T. Eugster, C. Fetzer, and A. Behrend, 81–84. Berlin, Germany: ACM. doi: <http://doi.acm.org/10.1145/2335484.2335493>.

- [C10] Verginadis, Y., I. Patiniotakis, N. Papageorgiou, and R. Stühmer. 2011. “Service Adaptation Recommender in the Event Marketplace: Conceptual View.” In *The Semantic Web: ESWC 2011 Workshops*, edited by R. Garcia-Castro, D. Fensel, and G. Antoniou, LNCS 7117:194–201. Heraklion, Crete, Greece: Springer Berlin Heidelberg. doi: http://dx.doi.org/10.1007/978-3-642-25953-1_16.

Ανακοινώσεις σε Συνέδρια

- [A1] Patiniotakis, I., D. Apostolou, and G. Mentzas. 2009. “Applying a Linguistic Operator for Aggregating Movie Preferences.” In Corfu, Greece.

Μετάφραση Όρων

Adaptation = Προσαρμογή

Advice = Συμβουλή

Aspect = Θέμα

Aspect-Oriented Adaptation = Θεματοστρεφής Προσαρμογή

Broker = Μεσάζων

Business Process = Επιχειρησιακή Διαδικασία

Business Process Execution Language (BPEL) ή Web Services Business Process Execution Language (WS-BPEL) = Γλώσσα Εκτέλεσης Επιχειρησιακών Διαδικασιών

CEP Engine = Μηχανισμός CEP ή Μηχανισμός επεξεργασίας σύνθετων γεγονότων

CEP Pattern ή CEPat = Πρότυπο Σύνθετων Γεγονότων

Choreography = Χορογραφία

Complex Event = Σύνθετο Γεγονός

Complex Event Processing (CEP) = Επεξεργασία Σύνθετων Γεγονότων

Component = (Λειτουργική) Μονάδα

Concern = Αρμοδιότητα

Context = Πλαίσιο

Contextualization = Πλαισιοποίηση

ECA ή Event-Condition-Action = ECA (αμετάφραστο) ή Γεγονός-Συνθήκη-Ενέργεια

ECA rules = Κανόνες ECA ή Κανόνες Γεγονότος-Συνθήκης-Ενέργειας

Engine = Μηχανισμός

Event = Γεγονός ή συμβάν

Event-Driven Architecture ή EDA = Αρχιτεκτονική Οδηγούμενη από Γεγονότα

Event Consumer = Παραλήπτης ή Καταναλωτής Γεγονότων

Event Source = Πηγή ή Παραγωγός Γεγονότων

Event Stream ή Flow = Ροή γεγονότων

Framework = Πλαίσιο εργασίας

Hierarchical Decomposition = Ιεραρχική Ανάλυση

Inference Engine = Επαγωγικός Μηχανισμός ή Μηχανισμός Εξαγωγής Συμπερασμάτων

Joint Point = Σημείο σύνδεσης

Middleware = Ενδιάμεσο λογισμικό

Orchestration = Ενορχήστρωση

Plan = Σχέδιο (Εκτέλεσης)

Planning = Σχεδιασμός

Planner = Σχεδιαστής

Plug-in = Επέκταση, στοιχείο επέκτασης, πρόσθετο

Pointcut = Σημείο τομής

Proactive = Προληπτικός, δρων εκ των προτέρων

Publish / Subscribe (Pub/Sub) = Αποστολή / Εγγραφή

Reactive = «Εκ των υστέρων», δρων εκ των υστέρων ως αποτέλεσμα κάποιας ενέργειας ή κατάστασης, αυτός που αντιδρά σε ένα ερέθισμα

Reasoning = Επαγωγή, πραγματοποίηση συλλογισμών, εξαγωγή συμπερασμάτων

SAN = σύντμηση του Situation – Action Network = Δίκτυο Κατάστασης – Ενέργειας

Service = Υπηρεσία

Service-Based Application (SBA) = Εφαρμογή Βασισμένη σε Υπηρεσίες

Service Composition = Σύνθεση Υπηρεσιών

Service-Oriented Architecture ή SOA = Αρχιτεκτονική Προσανατολισμένη σε Υπηρεσίες ή Υπηρεσιοστρεφής Αρχιτεκτονικής

Situation – Action Network (SAN) = Δίκτυο Κατάστασης – Ενέργειας

Trigger = Έναυσμα

Weaving = Ενσωμάτωση (κυριολεκτικά, πλέξη)

Web Service = Υπηρεσία Ιστού

Web Services Description Language (WSDL) = Γλώσσα Περιγραφής Υπηρεσιών Ιστού

Wrapping = Πλαισίωση

ΒΙΒΛΙΟΓΡΑΦΙΑ – ΑΝΑΦΟΡΕΣ – ΠΑΡΑΠΟΜΠΕΣ

- Abdelsalam, W., & Ebrahim, Y. (2004). A Bayesian-networks-based approach for managing uncertainty in location-tracking applications. In *Canadian Conference on Electrical and Computer Engineering* (pp. 2205–2208). IEEE. doi: 10.1109/CCECE.2004.1347682
- Adams, M., Edmond, D., & ter Hofstede, A. H. M. (2003). The application of activity theory to dynamic workflow adaptation issues. In *Proceedings of the 7th Pacific Asia Conference on Information Systems* (pp. 1836–1852).
- Adi, A., & Etzion, O. (2004). Amit - the situation manager. *VLDB J.*, 13(2), 177–203.
- Adomavicius, G., & Tuzhilin, A. (2010). Context-Aware Recommender Systems. In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.), *Recommender Systems Handbook: A Complete Guide for Research Scientists and Practitioners* (pp. 217–253). Springer. Retrieved from <http://ids.csom.umn.edu/faculty/gedas/nsfcareer/CARS-chapter-2010.pdf>
- Almeida, T., Vieira, S. C., & Casanova, M. A. (2004). Flexible workflow execution through an ontology-based approach. In *Proceedings of the Workshop on Ontologies as Software Engineering Artifacts (OOPSLA 2004)*.
- Anagnostopoulos, C., Ntirladimas, Y., & Hadjiefthymiades, S. (2007). Situational computing: An innovative architecture with imprecise reasoning. *Systems and Software*, 80(12), 1993–2014.
- Anicic, D., Fodor, P., Rudolph, S., & Stojanovic, N. (2011). EP-SPARQL: a unified language for event processing and stream reasoning. In *Proceedings of the 20th international conference on World wide web* (pp. 635–644). New York, NY, USA: ACM.
- Anicic, D., Rudolph, S., Fodor, P., & Stojanovic, N. (2012a). Real-Time Complex Event Recognition and Reasoning-a Logic Programming Approach. *Applied Artificial Intelligence*, 26(1-2), 6–57.
- Anicic, D., Rudolph, S., Fodor, P., & Stojanovic, N. (2012b). Stream Reasoning and Complex Event Processing in ETALIS, 3(4), 397–407.
- Appian. (2015). Business Process Definition. Retrieved from <http://www.appian.com/about-bpm/definition-of-a-business-process/>

- Atallah, L., & Yang, G.-Z. (2009). The use of pervasive sensing for behaviour profiling — a survey. *Pervasive and Mobile Computing*, 5(5), 447–464. doi: 10.1016/j.pmcj.2009.06.009
- Aydın, O., Kesim Cicekli, N., & Cicekli, I. (2008). Automated web services composition with the event calculus. In A. Artikis, G. M. P. O’Hare, K. Stathis, & G. Vouros (Eds.), *Engineering Societies in the Agents World VIII* (pp. 142–157).
- Bai, X., Chen, Y., & Shao, Z. (2007). Adaptive web services testing. In *Proceedings of the 31st Annual International Computer Software and Applications Conference (COMPSAC)* (pp. 233–236).
- Baldan, P., Corradini, A., Ehrig, H., & Heckel, R. (2001). Compositional Modeling of Reactive Systems Using Open Nets. In *Concurrency Theory (CONCUR 2001)* (Vol. 2154, pp. 502–518). Aarhus, Denmark: Springer-Verlag. doi: 10.1007/3-540-44685-0_34
- Baldauf, M., Dustdar, S., & Rosenberg, F. (2007). A survey on context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computing*, 2(4), 263. doi: 10.1504/IJAHUC.2007.014070
- Bana e Costa, C. A., & Vansnick, J.-C. (1997). Applications of the MACBETH Approach in the Framework of an Additive Aggregation Model. *Journal of Multi-Criteria Decision Analysis*, 6(2), 107–114. doi: 10.1002/(SICI)1099-1360(199703)6:2<107::AID-MCDA147>3.0.CO;2-1
- Bao, L., & Intille, S. S. (2004). Activity Recognition from User-Annotated Acceleration Data. In A. Ferscha & F. Mattern (Eds.), *Pervasive Computing* (Vol. 3001, pp. 1–17). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from http://link.springer.com/10.1007/978-3-540-24646-6_1
- Baresi, L., Ghezzi, C., & Guinea, S. (2004). Towards Self-healing Service Compositions. In *Proceedings of the First Conference on the Principles of Software Engineering (PRISE’04)* (pp. 11–20).
- Baresi, L., Guinea, S., & Pasquale, L. (2007). Self-healing BPEL processes with Dynamo and the JBoss rule engine. In *Proceedings of the International workshop on Engineering of software services for pervasive environments* (pp. 11–20).
- Baresi, L., Guinea, S., & Plebani, P. (2007). Policies and Aspects for the Supervision of BPEL Processes. *Advanced Information Systems Engineering*, 4495, 340–354.
- Bastida, L., Nieto, F. J., & Tola, R. (2008). Context-aware service composition: a methodology and a case study. In *Proceedings of the 2nd international workshop on Systems development in SOA environments (SDSOA) 2008* (p. 19). ACM Press. doi: 10.1145/1370916.1370921
- Bauer, C. (2012). A Comparison and Validation of 13 Context Meta-Models. In *ECIS 2012*. AIS Electronic Library. Retrieved from <http://aisel.aisnet.org/ecis2012/17>
- Baumgartner, N., Gottesheim, W., Mitsch, S., Retschitzegger, W., & Schwinger, W. (2010). BeAware!—Situation awareness, the ontology-driven way. *Data & Knowledge Engineering*, 69, 1181–1193.
- Baumgartner, N., & Retschitzegger, W. (2007). Towards a Situation Awareness Framework Based on Primitive Relations. In *Information, Decision and Control* (pp. 291–295). IEEE. doi: 10.1109/IDC.2007.374565
- Behavior Tree Group. (2007). *Behavior Tree Notation v1.0*. Retrieved from <http://www.behaviorengineering.org/docs/Behavior-Tree-Notation-1.0.pdf>

- Bellman, R. E., & Zadeh, L. A. (1970). Decision-Making in a Fuzzy Environment. *Management Science*, 17(4), B-141–B-164.
- Ben-Arieh, D., & Chen, Z. (2006). Linguistic-labels aggregation and consensus measure for autocratic decision making using group recommendations. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 36(3), 558–568. doi: 10.1109/TSMCA.2005.853488
- Benayoun, R., Roy, B., & Sussman, N. (1966). *Manual de Reference du Programme Electre*. Paris, France: Direction Scientifique SEMA.
- Berkovsky, S., & Eytani, Y. (2005). Semantic Platform for Context-Aware Publish/Subscribe M-Commerce. In *Proceedings of the SAINT Workshops* (pp. 188–191). Trento Italy.
- Bettini, C., Brdiczka, O., Henriksen, K., Indulska, J., Nicklas, D., Ranganathan, A., & Riboni, D. (2010). A survey of context modelling and reasoning techniques. *Pervasive and Mobile Computing*, 6(2), 161–180.
- Blake, M. B., & Nowlan, M. F. (2007). A web service recommender system using enhanced syntactical matching. In *Proceedings of the 5th International Conference on Web Services (ICWS 2007)*.
- Bolchini, C., Curino, C. A., Quintarelli, E., Schreiber, F. A., & Tanca, L. (2007). A data-oriented survey of context models. *ACM SIGMOD Record*, 36(4), 19. doi: 10.1145/1361348.1361353
- Bonnet, F., Decker, G., Dugan, L., Kurz, M., Misiak, Z., & Ringuette, S. (2014). Making BPMN a True lingua franca. Retrieved from <http://www.bptrends.com/making-bpmn-a-true-lingua-franca/>
- Boukadi, K., Ghedira, C., Maamar, Z., Benslimane, D., & Vincent, L. (2009). Context-Aware Data and IT Services Collaboration in E-Business. In *Proceedings of the Transactions on Large-Scale Data-and Knowledge-Centered Systems I* (pp. 91–115).
- Boukadi, K., Ghedira, C., & Vincent, L. (2008). An Aspect Oriented Approach for Context-Aware Service Domain Adapted to E-Business. In Z. Bellahsene & M. Léonard (Eds.), *CAiSE 2008* (Vol. 5074, pp. 64–78).
- Brans, J. P., Mareschal, B., & Vincke, P. (1984). PROMETHEE: A new family of outranking methods in multicriteria analysis. In J. P. Brans (Ed.), *Operational Research* (pp. 477–490). Amsterdam, Netherlands: North-Holland Publishing Company.
- Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., & Mylopoulos, J. (2004). Tropos: An agentoriented software development methodology. *Autonomous Agents and Multi-Agent Systems*, 8(3), 203–236.
- Brézillon, P. (1999). Context in problem solving: a survey. *The Knowledge Engineering Review*, 14(1), 47–80. doi: 10.1017/S0269888999141018
- Bruneton, E., Coupaye, T., Leclercq, M., Quema, V., & Stefani, J. (2006). The FRACTAL Component Model and its Support in Java. *Software Practice and Experience, Special Issue on Experiences with Auto-adaptive and Reconfigurable Systems*, 36(11-12), 1257–1284.
- Bucchiarone, A., Cappiello, C., di Nitto, E., Kazhamiakin, R., Mazza, V., & Pistore, M. (2010). Design for Adaptation of Service-Based Applications: Main Issues and Requirements. In A. Dan, F. Gittler, & F. Toumani (Eds.), *Proceedings of the ICSOC/ServiceWave Workshops 2009* (Vol. 6275, pp. 467–476). doi: 10.1007/978-3-642-16132-2_44

- Bui, H. H., Venkatesh, S., & West, G. (2002). Policy Recognition in the Abstract Hidden Markov Model. *Journal of Artificial Intelligence Research*, 17(1), 451–499.
- businessdictionary.com. (2015). Business Process. Retrieved from <http://www.businessdictionary.com/definition/business-process.html>
- Canfora, G., Di Penta, M., Esposito, R., & Villani, M. L. (2008). A framework for QoS-aware binding and re-binding of composite web services. *Journal of Systems and Software*, 81(10), 1754–1769. doi: 10.1016/j.jss.2007.12.792
- Cardell-Oliver, R., & Liu, W. (2010). Representation and recognition of situations in sensor networks. *IEEE Communications Magazine*, 112–117.
- Carlsen, S., Krogstie, J., Sjølvberg, A., & Lindland, O. I. (1997). Evaluating Flexible Workflow Systems. In *Proceedings of the Thirtieth Hawaii International Conference on System Sciences (HICSS-30)*. Maui, Hawaii: IEEE Computer Society Press.
- Castro, J., Kolp, M., & Mylopoulos, J. (2002). Towards requirements-driven information systems engineering: The tropos project. *Information Systems*, 27(6), 365–389.
- Catarci, T., de Leoni, M., Marrella, A., Mecella, M., Salvatore, B., Vetere, G., ... Manzoor, A. (2008). Pervasive software environments for supporting disaster responses. *IEEE Internet Computing*, 26–37.
- Chan, W., Cheung, S., & Leung, K. (2007). A metamorphic testing approach for online testing of service-oriented software applications. *International Journal of Web Services Research*, 4(2), 61–81.
- Charfi, A., Dinkelaker, T., & Mezini, M. (2009). A plug-in architecture for self-adaptive web service compositions. In *Proceedings of the 2009 IEEE International Conference on Web Services (ICWS '09)*, IEEE Computer Society (pp. 35–42).
- Charfi, A., & Mezini, M. (2006). Aspect-Oriented Workflow Languages. In R. Meersman & Z. Tari (Eds.), *On the Move to Meaningful Internet Systems 2006: OTM Confederated International Conferences, CoopIS, DOA, GADA, and ODBASE 2006* (Vol. 4275, pp. 183–200). Heidelberg: Springer.
- Charfi, A., & Mezini, M. (2007). Ao4bpel: An aspect-oriented extension to bpel. *World Wide Web Journal: Recent Advances on Web Services (special Issue)*, 10(3), 309–344.
- Charfi, A., Mueller, H., & Mezini, M. (2010). Aspect-Oriented Business Process Modeling with AO4BPMN. In T. Kuehne et al. (Ed.), *Proceedings of 6th European Conference on Modelling Foundations and Applications (ECMFA 2010)* (Vol. 6138, pp. 48–61). Berlin, Heidelberg: Springer-Verlag.
- Chen, G., & Kotz, D. (2000). *A Survey of Context-Aware Mobile Computing Research*. Hanover, NH, USA: Dartmouth College.
- Chen, H., Finin, T., & Joshi, A. (2003). An ontology for context-aware pervasive computing environments. *The Knowledge Engineering Review*, 18(3), 197–207. doi: 10.1017/S0269888904000025
- Chen, L., Nugent, C., Mulvenna, M., Finlay, D., & Hong, X. (2009). Semantic Smart Homes: Towards Knowledge Rich Assisted Living Environments. In S. McClean, P. Millard, E. El-Darzi, & C. Nugent (Eds.), *Intelligent Patient Management* (Vol. 189, pp. 279–296). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from http://link.springer.com/10.1007/978-3-642-00179-6_17

- Chen, T. Y., Cheung, S. C., & Yiu, S. M. (1998). *Metamorphic testing: a new approach for generating next test cases* (No. Technical Report HKUST-CS98-01). Hong Kong: Department of Computer Science, Hong Kong University of Science and Technology.
- Chen, X., Liu, X., Huang, Z., & Sun, H. (2010). RegionKNN: A Scalable Hybrid Collaborative Filtering Algorithm for Personalized Web Service Recommendation. In *Proceedings of the 2010 IEEE International Conference on Web Services* (pp. 9–16).
- Cheung, K. S. K., & Vogel, D. (2005). Complexity Reduction in Lattice-Based Information Retrieval. *Information Retrieval*, 8(2), 285–299. doi: 10.1007/s10791-005-5663-y
- Choujaa, D., & Dulay, N. (2009). Activity Inference through Sequence Alignment. In *Location and Context Awareness* (Vol. 5561, pp. 19–36). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from http://link.springer.com/10.1007/978-3-642-01721-6_2
- Clarkson, B., & Pentland, A. (1999). Unsupervised clustering of ambulatory audio and video. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP99)* (pp. 3037–3040 vol.6). IEEE. doi: 10.1109/ICASSP.1999.757481
- Colombo, M., Di Nitto, E., & Mauri, M. (2006). SCENE: A Service Composition Execution Environment Supporting Dynamic Changes Disciplined Through Rules. In Asit Dan & W. Lamersdorf (Eds.), *Service-Oriented Computing – ICSOC 2006* (Vol. 4294, pp. 191–202). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from http://link.springer.com/10.1007/11948148_16
- Colvin, R., Grunske, L., & Winter, K. (2008). Timed Behavior Trees for Failure Mode and Effects Analysis of Time-Critical Systems. *Journal of Systems and Software*.
- Colvin, R., & Hayes, I. J. (2006). *A Semantics for Behavior Trees, Technical report*. Retrieved from http://www.accs.uq.edu.au/documents/TechnicalReports/ACCS_TR_07_01.pdf
- Cook, D. J., Augusto, J. C., & Jakkula, V. R. (2009). Ambient intelligence: Technologies, applications, and opportunities. *Pervasive and Mobile Computing*, 5(4), 277–298. doi: 10.1016/j.pmcj.2009.04.001
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2001). *Introduction to Algorithms; Section 22.3: Depth-first search* (2nd ed.). MIT Press and McGraw-Hill.
- Costa, P., Guizzardi, G., Almeida, J., Pires, L., & Sinderen, M. (2006). Situations in Conceptual Modeling of Context. In *Proceedings of the 10th IEEE International Enterprise Distributed Object Computing Conference Workshops (EDOCW), 2006* (p. 6). IEEE. doi: 10.1109/EDOCW.2006.62
- Courbis, C., & Finkelstein, A. (2005). Towards aspect weaving applications. In *Proceedings of the 27th international conference on Software engineering (ICSE) 2005* (pp. 69–77). ACM Press. doi: 10.1145/1062455.1062484
- Da Cunha Mattos, T., Santoro, F. M., Revoredo, K., & Nunes, V. T. (2014). A formal representation for context-aware business processes. *Computers in Industry*, 65(8), 1193–1214. doi: 10.1016/j.compind.2014.07.005
- Dai, W., & Liu, J. (2010). Context aware adaptive services framework. In *Proceedings of the 7th International Conference on Service Systems and Service Management (ICSSSM)* (pp. 1–6). Tokyo: IEEE. doi: 10.1109/ICSSSM.2010.5530202

- David, P., & Ledoux, T. (2005). Wildcat: a generic framework for context-aware applications. In *Proceedings of the 3rd international workshop on Middleware for pervasive and ad-hoc computing (MPAC)* (pp. 1–7).
- David, P., & Ledoux, T. (2006). An Aspect-Oriented Approach for Developing Self-Adaptive Fractal Components. In *Proceedings of the 5th Int. Symposium on Software Composition* (Vol. 4089, pp. 82–99). Vienna, Austria.
- Del Carmen Rodríguez-Hernández, M., & Ilarri, S. (2014). Towards a Context-Aware Mobile Recommendation Architecture. In I. Awan, M. Younas, X. Franch, & C. Quer (Eds.), *Mobile Web Information Systems* (Vol. 8640, pp. 56–70). Cham: Springer International Publishing. Retrieved from http://link.springer.com/10.1007/978-3-319-10359-4_5
- Delgado, M., Verdegay, J. L., & Vila, M. A. (1993). On aggregation operations of linguistic labels. *International Journal of Intelligent Systems*, 8(3), 351–370. doi: 10.1002/int.4550080303
- Delir Haghighi, P., Krishnaswamy, S., Zaslavsky, A., & Gaber, M. M. (2008). Reasoning about Context in Uncertain Pervasive Computing Environments. In D. Roggen, C. Lombriser, G. Tröster, G. Kortuem, & P. Havinga (Eds.), *Smart Sensing and Context* (Vol. 5279, pp. 112–125). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from http://link.springer.com/10.1007/978-3-540-88793-5_9
- Deussen, P., Din, G., & Schieferdecker, I. (2003). A TTCN-3 based online test and validation platform for Internet services. In *Proceedings of the 6th International Symposium on Autonomous Decentralized Systems (ISADS)* (pp. 177–184).
- Dey, A. (2001). Understanding and using context. *Personal and Ubiquitous Computing*, 5(1), 4–7. doi: doi:10.1007/s007790170019
- Dey, A., & Abowd, G. (2000). Towards a Better Understanding of Context and Context-Awareness. In *Proceedings of the PrCHI 2000 Workshop on the What, Who, Where, When, and How of Context-Awareness* (pp. 304–307).
- Di Penta, M., Bruno, M., Esposito, G., Mazza, V., & Canfora, G. (2007). Web Services Regression Testing. In *Test and Analysis of Web Services* (pp. 205–234). Springer.
- Dicken, C., & Gould, K. (2008). Process as an Asset. Retrieved from <http://www.bptrends.com/publicationfiles/07-08-ART-Process%20as%20an%20Asset-Dicken%20and%20Gould-final.doc.pdf>
- Ding, Z., & Peng, Y. (2004). A probabilistic extension to ontology language OWL. In *37th Annual Hawaii International Conference on System Sciences* (p. 10 pp.). IEEE. doi: 10.1109/HICSS.2004.1265290
- Dittrich, K. R., Gatzju, S., & Geppert, A. (1995). The Active Database Management System Manifesto: A Rulebase of ADBMS Features (Vol. 985, pp. 3–20). Springer.
- Dorn, C., Burkhart, T., Werth, D., & Dustdar, S. (2010). Self-adjusting Recommendations for People-Driven Ad-Hoc Processes. In *Proceedings of the 8th international conference on Business process management (BPM 2010)* (pp. 327–342). Springer.
- Drean, O. (2010). Are Business Processes an Asset? Retrieved from <http://www.apriso.com/blog/2010/12/are-business-processes-an-asset/>
- Dromey, R. G. (2001). Genetic Software Engineering - Simplifying Design Using Requirements Integration. In *IEEE Working Conference on Complex and Dynamic Systems Architecture*. Brisbane, Australia.

- Dromey, R. G. (2003). From Requirements to Design: Formalizing the Key Steps. In *Proceedings of the 1st IEEE International Conference on Software Engineering and Formal Methods (SEFM-2003), Invited Keynote Address* (pp. 2–11). Brisbane, Australia.
- Dromey, R. G. (2005). Genetic Design: Amplifying Our Ability to Deal With Requirements Complexity. In S. Leue & T. J. Systra (Eds.), *Scenarios* (Vol. 3466, pp. 95–108).
- Dromey, R. G. (2006). Formalizing the Transition from Requirements to Design. In *Mathematical Frameworks for Component Software - Models for Analysis and Synthesis* (pp. 156–187). World Scientific Publishing Co, USA.
- Dumas, Marlon, La Rosa, M., Mendling, J., & Reijers, H. A. (2013). *Fundamentals of Business Process Management*. Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from <http://link.springer.com/10.1007/978-3-642-33143-5>
- Duong, T. V., Bui, H. H., Phung, D. Q., & Venkatesh, S. (2005). Activity Recognition and Abnormality Detection with the Switching Hidden Semi-Markov Model. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* (Vol. 1, pp. 838–845). IEEE. doi: 10.1109/CVPR.2005.61
- Edwards, W. (1977). How to Use Multiattribute Utility Measurement for Social Decisionmaking. *IEEE Transactions on Systems, Man, and Cybernetics*, 7(5), 326–340. doi: 10.1109/TSMC.1977.4309720
- Edwards, W., & Barron, F. H. (1994). SMARTS and SMARTER: Improved Simple Methods for Multiattribute Utility Measurement. *Organizational Behavior and Human Decision Processes*, 60(3), 306–325. doi: 10.1006/obhd.1994.1087
- Eichholz, C., Dittmar, A., & Forbrig, P. (2004). Using task modelling concepts for achieving adaptive workflows. In *Proceedings of the EHCI DS-VIS 2004* (pp. 96–111).
- Endsley, M. R. (1995). Toward a theory of situation awareness in dynamic systems: Situation awareness. *Human Factors*, 37, 32–64.
- Endsley, M. R. (2000). Theoretical underpinnings of situation awareness: a critical review. In *Situation Awareness Analysis and Measurement*. Mahwah, NJ, USA: Lawrence Erlbaum Associates.
- Endsley, M. R., & Connors, E. S. (2008). Situation awareness: State of the art. In *Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century, 2008 IEEE* (pp. 1–4).
- Erradi, A., Maheshwari, P., & Tosic, V. (2006). Policy-Driven Middleware for Self-adaptation of Web Services Compositions. In *Proceedings of the 7th International ACM Middleware Conference* (pp. 62–80).
- Escobar-Toledo, C. E., & López-García, B. (2005). The use of multicriteria decision aid system in the information technology (It) allocation problem. *Operational Research*, 5(2), 223–240. doi: 10.1007/BF02944310
- Etzion, O., & Niblett, P. (2010). *Event-Driven Processing in Action*. Manning Publications.
- Etzion, O., Skarbovsky, I., Magid, Y., Zolotorevsky, N., & Rabinovich, E. (2010). Context Aware Computing and its utilization in event-based systems. In *Proceedings of the Fourth ACM International Conference on Distributed Event-Based Systems, Tutorial presented in DEBS'10*. Cambridge, UK.

- Faggiolani, C. (2011). Perceived Identity: applying Grounded Theory in Libraries. *JLIS.it (University of Florence)*, 2(1). doi: doi:10.4403/jlis.it-4592
- Feng, Y.-H., Teng, T.-H., & Tan, A.-H. (2009). Modelling situation awareness for Context-aware Decision Support. *Expert Systems with Applications*, 36(1), 455–463. doi: 10.1016/j.eswa.2007.09.061
- Fine, S., & Singer, Y. (1998). The Hierarchical Hidden Markov Model: Analysis and Applications. In *Machine Learning* (pp. 41–62).
- Forrester. (2013). *Market Analysis of Multichannel Capture, Business Process Management, and Smart Process Applications, 2013 through 2016. A summary of a commissioned study conducted by Forrester® Consulting on behalf of Kofax*. Retrieved from <http://www.kofax.com/downloads/reports/report-market-analysis-spa.pdf>
- Frost, A. (2015). Knowledge Management. Retrieved from <http://www.knowledge-management-tools.net/>
- Fülöp, J. (2005). *Introduction to Decision Making Methods* (Working paper of the Laboratory of Operations Research and Decision Systems (LORDS) No. WPO5-6). Retrieved from <http://academic.evergreen.edu/projects/bdei/documents/decisionmakingmethods.pdf>
- Ganti, R., Ye, F., & Lei, H. (2011). Mobile crowdsensing: current state and future challenges. *IEEE Communications Magazine*, 49(11), 32–39. doi: 10.1109/MCOM.2011.6069707
- Gartner. (2014). *Hype Cycle for Business Process Management* (No. G00258623).
- Gebauer, J., & Schober, F. (2006). Information System Flexibility and the Cost Efficiency of Business Processes. *Journal of the Association for Information Systems*, 7(3), 122–147.
- Geebelen, K., Kulikowski, E., Truyen, E., & Joosen, W. (2010). A MVC Framework for Policy-Based Adaptation of Workflow Processes: A Case Study on Confidentiality. In *IEEE International Conference on Web Services (ICWS) 2010* (pp. 401–408). IEEE. doi: 10.1109/ICWS.2010.81
- Gehlert, A., Metzger, A., Karastoyanova, D., Kazhamiakin, R., Pohl, K., Leymann, F., & Pistore, M. (2011). Chapter 6: Integrating Perfective and Corrective Adaptation of Service-based Applications. In *Service Engineering: European Research Results Book* (pp. 137–169). Springer.
- Goodrich, M. T. (2001). *Algorithm Design: Foundations, Analysis, and Internet Examples*. Wiley.
- Grassi, V., & Sindico, A. (2007). Towards model driven design of service-based context-aware applications. In *Proceedings of International workshop on Engineering of software services for pervasive environments (ESSPE '07): in conjunction with the 6th ESEC/FSE joint meeting* (pp. 69–74).
- Greene, R. (2007). OODA and You. Retrieved from <http://powerseductionandwar.com/ooda-and-you/>
- Gu, T., Pung, H. K., & Zhang, D. Q. (2004). A Bayesian approach for dealing with uncertain contexts. *Proceedings of the Second International Conference on Pervasive Computing*.

- Gu, T., Wang, X. H., Pung, H. K., & Zhang, D. Q. (2004). An Ontology-based Context Model in Intelligent Environments. In *Proceedings of the Conference on Communication Network and Distributed Systems* (pp. 270–275).
- Gu, Tao, Chen, S., Tao, X., & Lu, J. (2010). An unsupervised approach to activity recognition and segmentation based on object-use fingerprints. *Data & Knowledge Engineering*, 69(6), 533–544. doi: 10.1016/j.datak.2010.01.004
- Gu, Tao, Pung, H. K., & Zhang, D. Q. (2005). A service-oriented middleware for building context-aware services. *Journal of Network and Computer Applications*, 28(1), 1–18. doi: 10.1016/j.jnca.2004.06.002
- Habiba, U., & Asghar, S. (2009). A survey on multi-criteria decision making approaches. In *IEEE International Conference on Emerging Technologies* (pp. 321–325). Islamabad: IEEE. doi: 10.1109/ICET.2009.5353151
- Han, J., & Kamber, M. (2006). *Data mining: concepts and techniques* (2nd ed). Amsterdam ; Boston : San Francisco, CA: Elsevier ; Morgan Kaufmann.
- Harter, A., Hopper, A., Steggles, P., Ward, A., & Webster, P. (1999). The anatomy of a context-aware application. In *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking (MobiCom) 1999* (pp. 59–68). ACM Press. doi: 10.1145/313451.313476
- Hasan, M. K., Rubaiyeat, H. A., Lee, Y.-K., & Lee, S. (2008). A Reconfigurable HMM for Activity Recognition. In *10th International Conference on Advanced Communication Technology* (pp. 843–846). IEEE. doi: 10.1109/ICACT.2008.4493886
- Heckel, R. (2003). Open Petri Nets as Semantic Model for Workflow Integration. In H. Ehrig, W. Reisig, G. Rozenberg, & H. Weber (Eds.), *Petri Net Technology for Communication-Based Systems* (Vol. 2472, pp. 281–294). Springer-Verlag, Berlin Heidelberg. doi: 10.1007/978-3-540-40022-6_14
- Heinl, P., Horn, S., Jablonski, S., Neeb, J., Stein, K., & Teschke, M. (1999). A Comprehensive Approach to Flexibility in Workflow Management Systems. In *Proceedings of the international joint conference on Work activities coordination and collaboration (WACC '99)* (pp. 79–88). New York, NY, USA: ACM.
- Henricksen, K., & Indulska, J. (2006). Developing context-aware pervasive computing applications: Models and approach. *Pervasive and Mobile Computing*, 2(1), 37–64. doi: 10.1016/j.pmcj.2005.07.003
- Hermsillo, G. (2012). *Towards Creating Context-Aware Dynamically-Adaptable Business Processes Using Complex Event Processing*. niversit_e des Sciences et Technologie de Lille - Lille I, France. Retrieved from <https://tel.archives-ouvertes.fr/tel-00709303>
- Hermsillo, G., Seinturier, L., & Duchien, L. (2010). Using Complex Event Processing for Dynamic Business Process Adaptation. In *Proceedings of the 2010 IEEE International Conference on Services Computing (IEEE SCC 2010)* (pp. 466–473).
- Herrera, F., & Herrera-Viedma, E. (1997). Aggregation operators for linguistic weighted information. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 27(5), 646–656. doi: 10.1109/3468.618263
- Herrera, F., Herrera-Viedma, E., & Verdegay, J. L. (1996). Direct approach processes in group decision making using linguistic OWA operators. *Fuzzy Sets and Systems*, 79(2), 175–190. doi: 10.1016/0165-0114(95)00162-X

- Hielscher, J., Kazhamiakin, R., Metzger, A., & Pistore, M. (2008). A framework for proactive self-adaptation of service-based applications based on online testing. In *Proceedings of the ServiceWave 2008* (pp. 122–133).
- Hinze, A., Michel, Y., & Eschner, L. (2009). Event-based Communication for Location-based Service Collaboration. In A. Bouguettaya & X. Lin (Eds.), *Proceedings of 20th Australasian Database Conference (ADC 2009)* (Vol. 92, pp. 127–136). Wellington, New Zealand: ACS.
- Hinze, A., Sachs, K., & Buchmann, A. (2009). Event-Based Applications and Enabling Technologies. In *Proceedings of the 3rd ACM International Conference on Distributed Event-Based Systems*. Nashville, USA.
- Hoang, H., Lee-Urban, S., & Muñoz-Avila, H. (2005). Hierarchical Plan Representations for Encoding Strategic Game AI. In *Proceedings of AIIDE-05*. AAAI Press.
- Hu, S., Muthusamy, V., Li, G., & Jacobsen, H. (2008). Distributed automatic service composition in large-scale systems. In *Proceedings of the 2nd International Conference on Distributed Event-based Systems* (pp. 233–244). Rome, Italy.
- Hwang, C. L., & Masud, A. S. M. (1979). *Multiple objective decision making, methods and applications: a state-of-the-art survey*. Berlin ; New York: Springer-Verlag.
- Hwang, C. L., & Yoon, K. (1981). *Multiple attribute decision making: methods and applications : a state-of-the-art survey* (Vol. 186). Berlin; New York: Springer-Verlag.
- Jacquet-Lagrez, E., & Siskos, J. (1982). Assessing a set of additive utility functions for multicriteria decision-making, the UTA method. *European Journal of Operational Research*, 10(2), 151–164. doi: 10.1016/0377-2217(82)90155-2
- Jakobson, G. (2008). Introduction to cognitive situation management for tactical operations. *IEEE Communications Society, Distinguished Lecture, Goteborg*.
- Jakobson, G., Buford, J., & Lewis, L. (2007). Situation management: Basic concepts and approaches. *Information Fusion and Geographic Information Systems*, 18–33.
- Jayaraman, P., Whittle, J., Elkhodary, A., & Gomaa, H. (2007). Model Composition in Product Lines and Feature Interaction Detection Using Critical Pair Analysis. In *Proceedings of the 10th ACM/IEEE International Conference on Model Driven Engineering Languages and Systems* (pp. 151–165). Nashville, USA.
- JBoss community. (n.d.). Drools - The Business Logic integration Platform. Retrieved from <http://www.jboss.org/drools>
- Jung, J., Park, J., Han, S., & Lee, K. (2007). An ECA-based framework for decentralized coordination of ubiquitous web services. *Information and Software Technology*, 49(11-12), 1141–1161.
- Juric, M. (2010). WSDL and BPEL extensions for Event Driven Architecture. *Information and Software Technology*, 52(10), 1023–1043.
- Kaltz, W. J., Ziegler, J., & Lohmann, S. (2005). Context-aware Web Engineering: Modeling and Applications. *Revue D'intelligence Artificielle*, 19(3), 439–458. doi: 10.3166/ria.19.439-458
- Kanda, T., Glas, D. F., Shiomi, M., Ishiguro, H., & Hagita, N. (2008). Who will be the customer?: a social robot that anticipates people's behavior from their trajectories. In *Proceedings of the 10th international conference on Ubiquitous computing (UbiComp) 2008* (p. 380). ACM Press. doi: 10.1145/1409635.1409686

- Karantonis, D. M., Narayanan, M. R., Mathie, M., Lovell, N. H., & Celler, B. G. (2006). Implementation of a Real-Time Human Movement Classifier Using a Triaxial Accelerometer for Ambulatory Monitoring. *IEEE Transactions on Information Technology in Biomedicine*, 10(1), 156–167. doi: 10.1109/TITB.2005.856864
- Karastoyanova, D., & Leymann, F. (2009). BPEL'n'Aspects: Adapting Service Orchestration Logic. In *Proceedings of the IEEE International Conference on Web Services, ICWS 2009* (pp. 222–229).
- Keeney, R. L. (1976). *Decisions with multiple objectives: Preferences and Value Trade-offs*. New York, NY, USA: Wiley, New York.
- Khalaf, R., Karastoyanova, D., & Leymann, F. (2007). Pluggable Framework for Enabling the Execution of Extended BPEL Behavior. In *Proceedings of WESOA workshop at ICSOC'07* (pp. 376–387). Springer.
- Kiczales, G., Hilsdale, E., Hugunin, J., Kersten, M., Palm, J., & Griswold, W. (2001). An overview of AspectJ. In *Proceedings of the 15th European Conference on Object-Oriented Programming (ECOOP '01)* (pp. 327–353). Springer-Verlag.
- Kiczales, G., Lamping, J., Mendheka, A., Maeda, C., Lopes, C., Loingtier, J., & Irwin, J. (1997). Aspect-Oriented Programming. In S. Gjessing & K. Nygaard (Eds.), *Proceedings of the European Conference on Object-Oriented Programming (ECOOP)* (Vol. 1241, pp. 220–242). Springer.
- KiSSFLOW. (n.d.). Top 5 Reasons why you should automate your business process. Retrieved from https://kissflow.com/process_playbook/5-reasons-why-you-should-automate-your-business-process/
- Klein, M., Schmidt, A., & Lauer, R. (2007). Ontology-Centred Design of an Ambient Middleware for Assisted Living: The Case of SOPRANO. In *Towards Ambient Intelligence: Methods for Cooperating Ensembles in Ubiquitous Environments (AIM-CU)*.
- Knuth, D. E. (1997). *The Art Of Computer Programming* (3rd ed., Vol. 1). Boston, USA: Addison-Wesley.
- Ko, R. K. L. (2009). A computer scientist's introductory guide to business process management (BPM). *Crossroads*, 15(4), 11–18. doi: 10.1145/1558897.1558901
- Kokar, M. M., Matheus, C. J., & Baclawski, K. (2009). Ontology-based situation awareness. *J. Information Fusion*, 10, 83–98.
- Kongdenfha, W., Saint-Paul, R., Benatallah, B., & Casati, F. (2006). An Aspect-Oriented Framework for Service Adaptation. In *Proceedings of the 4th International Conference in Service-Oriented Computing - ICSOC 2006* (pp. 15–26). Chicago, IL, USA: Springer.
- Kumar, R. L., & Narasipuram, M. M. (2006). Defining Requirements for Business Process Flexibility. In *Proceedings of CAiSE06 Workshops, Workshop on Business Process Modeling, Design and Support (BP-MDS06)* (pp. 137–148).
- Kumar, R. L., & Stylianou, A. C. (2013). A process model for analyzing and managing flexibility in information systems. *European Journal of Information Systems Advance*.
- Küster, U., Stern, M., & König-Ries, B. (2005). A classification of issues and approaches in automatic service composition. In *Proceedings of the 1st International Workshop on Engineering Service Compositions (WESCO5)*. Amsterdam, Netherlands.

- Lahire, P., Morin, B., Vanwormhoudt, G., Gaignard, A., Barais, O., & Jézequel, J. (2007). Introducing Variability into Aspect-Oriented Modeling Approaches. In *Proceedings of the 10th ACM/IEEE International Conference on Model Driven Engineering Languages and Systems* (pp. 498–513). Nashville USA.
- Laliwala, Z., Sorathia, V., & Chaudhary, S. (2006). Semantics based event-driven publish/subscribe service-oriented architecture. In *Proceedings of the workshop on “Emerging Services delivery platforms and software models for Next Generation Network Services” (SOFTPLATFORM2006), co-located with “The 1st International Conference on Communication Systems Software and Middleware” (COMSWARE)* (pp. 1–5). New Delhi, India.
- Lane, N., Miluzzo, E., Lu, H., Peebles, D., Choudhury, T., & Campbell, A. (2010). A survey of mobile phone sensing. *IEEE Communications Magazine*, 48(9), 140–150. doi: 10.1109/MCOM.2010.5560598
- Lanese, I., Bucchiarone, A., & Montesi, F. (2010). A framework for rule-based dynamic adaptation. In *Proceedings of the 5th International Symposium on Trustworthy Global Computing (TGC 2010)* (pp. 284–300). Munchen, Germany.
- Lee, K, Lee, W., Jeon, J., Lee, S., & Park, J. (2006). Event-driven Coordination Rule of Web Services enabled Devices in Ubiquitous environments. In *Proceedings of the W3C Ubiquitous Web Workshop*. Tokyo, Japan.
- Lee, Karen, Lunney, T., Curran, K., & Santos, J. (2009). Ambient Middleware for Context-Awareness (AMiCA): *International Journal of Ambient Computing and Intelligence*, 1(3), 66–78. doi: 10.4018/jaci.2009070105
- Lee, S., Holme, P., & Wu, Z. X. (2011). Emergent hierarchical structures in multiadaptive games. *Physical Review Letters*, 106(2): 028702).
- Leeming, N. (2004). Why Automate Business Processes? Retrieved from <http://www.ivencia.com/softwarearchitect/articles/bpm/BPM2%20-%20Why%20automate%20business%20processes.pdf>
- Lemahieu, W., Snoeck, M., Michiels, C., Goethals, F., Dedene, G., & Vandenbulcke, J. (2003). Event based web service description and coordination. In *Proceedings of the Web Services, e-Business, and the Semantic Web Workshop of the CaiSE’03 Conference* (Vol. 3095, pp. 120–133). Klagenfurt, Austria.
- Leymann, F. (2005). The (Service) Bus: Services Penetrate Everyday Life. In *Proceedings of the 3rd International Conference in Service-Oriented Computing, (ICSOC’2005)* (pp. 12–20). Amsterdam, Netherlands.
- Li, L., Liu, D., & Bouguettaya, A. (2009). Semantic Weaving for Context-Aware Web Service Composition. In *Proceedings of the 10th International Conference on Web Information Systems Engineering* (pp. 101–114).
- Lim, C. U., Baumgarten, R., & Colton, S. (2010). Evolving Behaviour Trees for the Commercial Game DEFCON. *Applications of Evolutionary Computation*, 100–110.
- Loke, S. W. (2010). Incremental awareness and compositionality: A design philosophy for context-aware pervasive systems. *Pervasive and Mobile Computing*, 6(2), 239–253. doi: 10.1016/j.pmcj.2009.03.004
- Lu, Q., Li, S., & Zhang, W. (2014). A survey on adaptation decision-making of business processes and the affected web service compositions. *Computer Modelling & New Technologies*, 18(12C), 1302–1308.

- Luckham, D. C. (2001). *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
- Maamar, Z., Mostefaoui, S. K., & Mahmoud, Q. H. (2005). Context for personalized web services. In *Proceedings The 38th Annual Hawaii International Conference*.
- Makris, P., Skoutas, D. N., & Skianis, C. (2013). A Survey on Context-Aware Mobile and Wireless Networking: On Networking and Computing Environments' Integration. *IEEE Communications Surveys & Tutorials*, 15(1), 362–386. doi: 10.1109/SURV.2012.040912.00180
- Mani Chandy, K. (2006). *Event-Driven Applications: Costs, Benefits and Design Approaches*. California Institute of Technology. Retrieved from <http://infospheres.caltech.edu/sites/default/files/Event-Driven%20Applications%20-%20Costs,%20Benefits%20and%20Design%20Approaches.pdf>
- marketsandmarkets.com. (2014). *Business Process Management Market by Solutions (Process Modeling, Automation, Integration, Content & Document Management, and Monitoring & Optimization Management), by End User (SMBs, Enterprises & Large Enterprises) - Global Forecast to 2019* (No. TC 2958). Retrieved from <http://www.marketsandmarkets.com/Market-Reports/business-process-management-market-157890056.html?gclid=CLW3hrbRo8UCFWgOwwodFJkARQ>
- Marrella, A., Mecella, M., & Sardina, S. (2014). SmartPM: An Adaptive Process Management System through Situation Calculus, IndiGolog, and Classical Planning. In *Proceedings of Principles of Knowledge Representation and Reasoning* (pp. 518–527). Palo Alto.
- Martin, E. (2010). *The Oxford Dictionary of Science* (6th ed.). Oxford University Press.
- Martin, P. Y., & Turner, B. A. (1986). Grounded Theory and Organizational Research. *Journal of Applied Behavioral Science*, 22(2), 141–157.
- Mateas, M., & Stern, A. (2002). A Behavior Language for Story-Based Believable Agents. *IEEE Intelligent Systems*, 39–47.
- Matheus, C. J., Kokar, M. M., & Baclawski, K. (2003). A core ontology for situation awareness. In *Proceedings of the 6th International Conference of Information Fusion* (Vol. 1, pp. 545–552). doi: 10.1109/ICIF.2003.177494
- Mathie, M. J., Celler, B. G., Lovell, N. H., & Coster, A. C. F. (2004). Classification of basic daily movements using a triaxial accelerometer. *Medical & Biological Engineering & Computing*, 42(5), 679–687. doi: 10.1007/BF02347551
- McCarthy, J., & Hayes, P. (1969). Some philosophical problems from the standpoint of artificial intelligence. In D. Michie (Ed.), *Machine Intelligence 4* (pp. 463–502). Edinburgh University Press.
- McDermott, D. V. (2002). Estimated-Regression Planning for Interactions with Web Services. In M. Ghallab, J. Hertzberg, & P. Traverso (Eds.), *Proceedings of the 6th International Conference on Artificial Intelligence Planning Systems* (pp. 204–211). Toulouse, France: AAAI.
- McKeever, S., Ye, J., Coyle, L., & Dobson, S. (2009). Using Dempster-Shafer Theory of Evidence for Situation Inference. In P. Barnaghi, K. Moessner, M. Presser, & S. Meissner (Eds.), *Smart Sensing and Context* (Vol. 5741, pp. 149–162). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from http://link.springer.com/10.1007/978-3-642-04471-7_12

- Mehta, B., Niederée, C., Stewart, A., Muscogiuri, C., & Neuhold, E. J. (2004). An Architecture for Recommendation Based Service Mediation. In M. Bouzeghoub & et al (Eds.), *Semantics of a Networked World (ICSNW 2004)* (Vol. 3226, pp. 250–262).
- Mehta, M., & Ram, A. (2009). Runtime Behavior Adaptation for Real-Time Interactive Games. *Computational Intelligence and AI in Games, IEEE Transactions*, 1(3), 187–199.
- Mészáros, C., & Rapcsák, T. (1996). On sensitivity analysis for a class of decision systems. *Decision Support Systems*, 16(3), 231–240. doi: 10.1016/0167-9236(95)00012-7
- Miettinen, K. (1999). *Nonlinear Multiobjective Optimization*. Boston: Kluwer Academic Publishers.
- Miettinen, K., Ruiz, F., & Wierzbicki, A. P. (2008). Introduction to Multiobjective Optimization: Interactive Approaches. In J. Branke, K. Deb, K. Miettinen, & R. Słowiński (Eds.), *Multiobjective Optimization* (Vol. 5252, pp. 27–57). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from http://link.springer.com/10.1007/978-3-540-88908-3_2
- Minnen, D., Starner, T., Ward, J. A., Lukowicz, P., & Troster, G. (2005). Recognizing and Discovering Human Actions from On-Body Sensor Data. In *IEEE International Conference on Multimedia and Expo* (pp. 1545–1548). IEEE. doi: 10.1109/ICME.2005.1521728
- Mitchell, S. W. (1997). A hybrid architecture for real-time mixed-initiative planning and control. In *Proceedings of the IAAI-97*. AAAI Press.
- Moltchanov, B., Knappmeyer, M., Fuchs, O., & Paschetta, E. (2009). Context management and reasoning for adaptive service provisioning. In *Proceedings of the International Conference on Ultra Modern Telecommunications & Workshops (ICUMT)*.
- Moran, T., & Dourish, P. (2001). Introduction to This Special Issue on Context-Aware Computing. *Human-Computer Interaction*, 16(2), 87–95. doi: 10.1207/S15327051HCI16234_01
- Morin, B., Barais, O., Nain, G., & Jézéquel, J. (2009). Taming dynamically adaptive systems using models and aspects. In *Proceedings of the 31st International Conference on Software Engineering* (pp. 122–132). Vancouver, Canada.
- Morin, B., Fleurey, F., Bencomo, N., Jézéquel, J. M., Solberg, A., Dehlen, V., & Blair, G. (2008). An Aspect-Oriented and Model-Driven Approach for Managing Dynamic Variability. In *Proceedings of the 11th ACM/IEEE International Conference on Model Driven Engineering Languages and Systems (MODELS'08)* (pp. 782–796). Toulouse, France.
- Moser, O., Rosenberg, F., & Dustdar, S. (2008). Non-Intrusive Monitoring and Service Adaptation for WS-BPEL. In *Proceedings of the International Conference on World Wide Web* (pp. 815–824).
- Muhlenbrock, M., Brdiczka, O., Snowdon, D., & Meunier, J.-L. (2004). Learning to detect user activity and availability from a variety of sensor data. In *2nd IEEE Annual Conference on Pervasive Computing and Communications* (pp. 13–22). IEEE. doi: 10.1109/PERCOM.2004.1276841
- Muller, P., Fleurey, F., & Jézéquel, J. (2005). Weaving Executability into Object-Oriented Meta-languages. In *Proceedings of the 8th ACM/IEEE International Conference on Model Driven Engineering Languages and Systems* (pp. 264–278). Montego Bay, Jamaica.

- Munnelly, J., Fritsch, S., & Clarke, S. (2007). An aspect-oriented approach to the modularisation of context. In *Proceedings of the PerCom* (pp. 114–124).
- Myers, D., Wesley, L., & Center, A. (1994). CYPRESS: Reacting and Planning under Uncertainty. In *DARPA Proceedings: Rome Laboratory Planning Initiative* (p. 111). Morgan Kaufmann.
- Nau, D., Au, T.-C., Ilghami, O., Kuter, U., Muñoz-Avila, H., Murdock, J. W., ... Yaman, F. (2005). Applications of SHOP and SHOP2. *IEEE Intelligent Systems*, 20(2).
- Nguyen, N. T., Phung, D. Q., Venkatesh, S., & Bui, H. (2005). Learning and Detecting Activities from Movement Trajectories Using the Hierarchical Hidden Markov Models. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* (Vol. 2, pp. 955–960). IEEE. doi: 10.1109/CVPR.2005.203
- Nickols, F. (2003). The Goals Grid: A Tool for Clarifying Goals & Objectives. Retrieved from http://www.jeslen.com/goals_grid%202010.pdf
- Norberg, S. (2010). *Event Driven Architecture*. Retrieved from <http://www.slideshare.net/stnor/event-driven-architecture-3395407>
- Oracle. (2011). Using JConsole. Retrieved from <http://docs.oracle.com/javase/6/docs/technotes/guides/management/jconsole.html>
- Osinga, F. P. B. (2007). *Science, strategy and war: the strategic theory of John Boyd*. London ; New York: Routledge.
- Palma Durán, R. J. (2010). Java Behaviour Trees, User Guide. Retrieved from <https://github.com/gaia-ucm/jbt/blob/master/UserGuide/UserGuide.pdf>
- Palma, R., González-Calero, P. A., Gómez-Martin, M. A., & Gómez-Martin, P. P. (2011). Extending Case-Based Planning with Behavior Trees. In *Proceedings of the Twenty-Fourth International Florida Artificial Intelligence Research Society Conference*.
- Panagacos, T. (2012). *The Ultimate Guide to Business Process Management: Everything You Need to Know and How to Apply It to Your Organization*.
- Paolucci, M., Kalp, D., Pannu, A. S., Shehory, O., & Sycara, K. (1999). A planning component for RETSINA agents. *Intelligent Agents VI*.
- Papazoglou, M. P. (2008). *Web services: principles and technology*. Harlow: Pearson/Prentice Hall.
- Park, J., Lee, H., Park, P., & Shin, D. (2008). Development of a WS-ECA Rule Management System for a Healthcare Robot. In *Proceedings of the 8th Asia-Pacific Computer-Human Interaction Conference (APCHI)* (pp. 149–156). Seoul, Korea.
- Patel, S. N., Robertson, T., Kientz, J. A., Reynolds, M. S., & Abowd, G. D. (2007). At the Flick of a Switch: Detecting and Classifying Unique Electrical Events on the Residential Power Line (Nominated for the Best Paper Award). In J. Krumm, G. D. Abowd, A. Seneviratne, & T. Strang (Eds.), *UbiComp 2007: Ubiquitous Computing* (Vol. 4717, pp. 271–288). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from http://link.springer.com/10.1007/978-3-540-74853-3_16
- Patiniotakis, I., Apostolou, D., & Mentzas, G. (2011). Fuzzy UTASTAR: A method for discovering utility functions from fuzzy data. *Expert Systems with Applications*, 38(12), 15463–15474. doi: 10.1016/j.eswa.2011.06.014

- Patterson, D. J., Liao, L., Fox, D., & Kautz, H. (2003). Inferring High-Level Behavior from Low-Level Sensors. In A. K. Dey, A. Schmidt, & J. F. McCarthy (Eds.), *UbiComp 2003: Ubiquitous Computing* (Vol. 2864, pp. 73–89). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from http://link.springer.com/10.1007/978-3-540-39653-6_6
- Peloquin, J. J. (2001). Knowledge As A Corporate Asset: Thought Leadership from The Performance Paradigm. Retrieved from http://cog.kent.edu/lib/Peloquin-Knowledge_as_a_Corporate_Asset.pdf
- Perera, C., Zaslavsky, A., Christen, P., & Georgakopoulos, D. (2014). Context Aware Computing for The Internet of Things: A Survey. *IEEE Communications Surveys & Tutorials*, 16(1), 414–454. doi: 10.1109/SURV.2013.042313.00197
- Perez, D., Nicolau, M., O’Neill, M., & Brabazon, A. (2011). Evolving Behaviour Trees for the Mario AI Competition Using Grammatical Evolution. In C. Di Chio & et al (Eds.), *EvoApplications 2011, Part I* (Vol. 6624, pp. 123–132).
- Pesic, M., Schonenberg, M. H., Sidorova, N., & van der Aalst, W. M. P. (2007). Constraint-based workflow models: Change made easy. In *Proceedings of the OTM Conferences 2007* (Vol. 1, pp. 77–94).
- Pesic, M., & van der Aalst, W. M. P. (2006). A declarative approach for flexible business processes management. In *Proceedings of the 2006 international conference on Business Process Management Workshops (BPM 2006)* (pp. 169–180).
- Pirri, F., & Reiter, R. (1999). Some contributions to the situation calculus. *J. ACM*, 46(3), 325–364.
- Rahman, S., Aoumeur, N., & Saake, G. (2008). An adaptive eca-centric architecture for agile service-based business processes with compliant aspectual .NET environment. In *Proceedings of the 10th International Conference on Information Integration and Web-based Applications & Services* (pp. 240–247). ACM.
- Regev, G., Bider, I., & Wegmann, A. (2007). Defining business process flexibility with the help of invariants. *Software Process: Improvement and Practice*, 12(1), 65–79. doi: 10.1002/spip.301
- Regev, G., Soffer, P., & Schmidt, R. (2006). Taxonomy of Flexibility in Business Processes. In *Proceedings of the 7th Workshop on Business Process Modelling, Development and Support(BPMDS’06)*.
- Regev, G., & Wegmann, A. (2005). A Regulation-Based View on Business Process and Supporting System Flexibility. In *Proceedings of CAiSE05 Workshops, Workshop on Business Process Modeling, Design and Support (BPMDS05)* (pp. 35–42).
- Regev, G., & Wegmann, A. (2006). Business Process Flexibility: Weick’s Organizational Theory to the Rescue. In *CAISE*06 Workshop on Business Process Modelling, Development, and Support BPMDS ’06*. Luxemburg. Retrieved from <http://ceur-ws.org/Vol-236/paper14.pdf>
- Richards, C. (2004). *Certain to win: The strategy of John Boyd, applied to business*. Philadelphia: PA: Xlibris.
- Rodriguez-Dominguez, C., Benghazi, K., & Noguera, M. (2010). Redefinable events for dynamic reconfiguration of communications in ubiquitous computing. In *Proceedings of the 1st International Workshop on Data Dissemination for Large Scale Complex Critical Infrastructures (DD4LCCI’10)* (pp. 17–22). Valencia, Spain.

- Rong, W., Liu, K., & Liang, L. (2009). Personalized Web Service Ranking via User Group combining Association Rule. In *Proceedings of the IEEE International Conference on Web Services* (pp. 445–452).
- Rosemann, M., Recker, J., & Flender, C. (2008). Contextualisation of business processes. *International Journal of Business Process Integration and Management*, 3(1), 47. doi: 10.1504/IJBPIIM.2008.019347
- Roy, B. (1968). Classement et choix en présence de points de vue multiples (la méthode ELECTRE). *Revue Française D'automatique, D'informatique et de Recherche Opérationnelle. Recherche Opérationnelle*, 2(1), 57–75.
- Roy, B. (1978). ELECTRE III: Un algorithme de classements fonde sur une representation floue des preference en presence de criteres multiples. *Cahiers de CERO.*, 20(1), 3–24.
- Roy, B., & Bertier, P. (1971). La methode ELECTRE II: Une methode de classement en presence de critteres multiples (Vol. 142, pp. 25–32). Paris, France: SEMA Metra International, Direction Scientifique.
- Roy, B., & Bertier, P. (1973). La methode ELECTRE II: Une methode au media-planning. In M. Ross (Ed.), *Operational Research* (pp. 291–302). Amsterdam, Netherlands: North-Holland Publishing Company.
- Roy, B., & Bouyssou, D. (1993). *Aide Multicritère à la Décision : méthodes et Cas*. Paris, France: Economica.
- Ruth, M., Oh, S., Loup, A., Horton, B., Gallet, O., Mata, M., & Tu, S. (2007). Towards automatic regression test selection for web services. In *Proceedings of the 31st Annual International Computer Software and Applications Conference (COMPSAC)* (pp. 729–734).
- Saaty, T. L. (1980). *The Analytic Hierarchy Process, Planning, Piority Setting, Resource Allocation*. New York, NY, USA: McGraw-Hill.
- Saaty, T. L. (1996). *Decision making with dependence and feedback: the analytic network process: the organization and prioritization of complexity* (1st ed.). Pittsburgh, PA: RWS Publications.
- Saaty, T. L. (1999). Fundamentals of the Analytic Network Process. In *ISAHP 1999*. Kobe, Japan.
- Saidani, O., & Nurcan, S. (2007). Towards context aware business process modeling. Presented at the Business Process Modeling, Development, and Support (BPMDS), Trondheim, Norway.
- Salo, A., & Hämäläinen, R. P. (2010). Preference Programming – Multicriteria Weighting Models under Incomplete Information. In C. Zopounidis & P. M. Pardalos (Eds.), *Handbook of Multicriteria Analysis* (Vol. 103, pp. 167–187). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from http://link.springer.com/10.1007/978-3-540-92828-7_5
- Sánchez, M., & Villalobos, J. (2008). A flexible architecture to build workflows using aspect-oriented concepts. In *Proceedings of the 2008 AOSD workshop on Aspect-oriented modeling* (pp. 25–30). ACM.

- Santos, L., Wijnen, R., & Vink, P. (2007). A service-oriented middleware for context-aware applications. In *Proceedings of the 5th international workshop on Middleware for pervasive and ad-hoc computing* (pp. 37–42).
- Schiefer, J., Obweiger, H., & Suntinger, M. (2009). Correlating business events for event-triggered rules. In *Proceedings of the 2009 International Symposium on Rule Interchange and Applications, ser. RuleML '09* (pp. 67–81). Springer-Verlag, Berlin, Heidelberg.
- Schiefer, J., Rozsnyai, S., Rauscher, C., & Saurer, G. (2007). Event-driven rules for sensing and responding to business situations. In *Proceedings of the 2007 inaugural international conference on Distributed event-based systems, ser. DEBS '07* (pp. 198–205). ACM, New York, NY, USA.
- Schiefer, J., & Seufert, A. (2005). Management and controlling of time-sensitive business processes with sense & respond. In *Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation (CICMA'05) and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (IAWTIC'06)* (Vol. 1 (CIMCA-IAWTIC'06), pp. 77–82). IEEE Computer Society, Washington, DC, USA.
- Schilit, B. N., & Theimer, M. M. (1994). Disseminating active map information to mobile hosts. *IEEE Network*, 8(5), 22–32. doi: 10.1109/65.313011
- Schober, F., & Gebauer, J. (2011). How much to spend on flexibility? Determining the value of information system flexibility. *Decision Support Systems*, 51(3), 638–647.
- Schonenberg, H., Mans, R., Russell, N., Mulyar, N., & van der Aalst, W. (2008). Process flexibility: A survey of contemporary approaches. In *Advances in Enterprise Engineering I* (Vol. 10, pp. 16–30). Springer.
- Schonenberg, H., Weber, B., Dongen, B., & Aalst, W. (2008). Supporting flexible processes through recommendations based on history. In M. Dumas, M. Reichert, & M.-C. Shan (Eds.), *BPM 2008* (Vol. 5240, pp. 51–66).
- S-Cube. (2008a). Deliverable CD-JRA-1.3.2: Quality Reference Model for SBA. Retrieved from [http://www.s-cube-network.eu/results/deliverables/wp-jra-1.3/Reference Model for SBA.pdf](http://www.s-cube-network.eu/results/deliverables/wp-jra-1.3/Reference%20Model%20for%20SBA.pdf)
- S-Cube. (2008b). Deliverable PO-JRA-1.2.1: State of the Art Report, Gap Analysis of Knowledge on Principles, Techniques and Methodologies for Monitoring and Adaptation of SBAs. Retrieved from <http://www.s-cube-network.eu/results/deliverables/wp-jra-1.2/PO-JRA-1.2.1-State-of-the-Art-report-on-principles-techniques-and-methodologies-for-monitoring-and-adaptation.pdf>
- S-Cube. (2009). Deliverable # CD-JRA-1.2.2. Retrieved from <http://www.s-cube-network.eu/>
- Sentz, K., & Ferson, S. (2002). *Combination of Evidence in Dempster-Shafer Theory* (Technical Report No. SAND2002-0835, 800792). Sandia National Laboratories. Retrieved from <http://www.osti.gov/servlets/purl/800792-s9WKeP/native/>
- Shao, L., Zhang, J., Wei, Y., Zhao, J., Xie, B., & Mei, H. (2007). Personalised QoS prediction for web services via collaborative filtering. In *Proceedings 5th International Conference on Web Services (ICWS 2007)* (pp. 439–446).
- Siljee, J., Bosloper, I., Nijhuis, J., & Hammer, D. (2005). DySOA: Making Service Systems Selfadaptive. In *Proceedings of the 3rd International Conference Service-Oriented Computing (ICSOC)* (pp. 255–268).

- Simpson, C. (2014). Behavior trees for AI: How they work. Retrieved from http://www.gamasutra.com/blogs/ChrisSimpson/20140717/221339/Behavior_trees_for_AI_How_they_work.php
- Simpson, R. C. (2004). An XML Representation for Crew Procedures. Final Report NASA Faculty Fellowship Program (Johnson Space Center). Retrieved from http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20050202022_2005020152.pdf
- Sindhya, K., Deb, K., & Miettinen, K. (2008). A Local Search Based Evolutionary Multi-objective Optimization Approach for Fast and Accurate Convergence. In G. Rudolph, T. Jansen, S. Lucas, C. Poloni, & N. Beume (Eds.), *Parallel Problem Solving from Nature – PPSN X* (Vol. 5199, pp. 815–824). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from http://link.springer.com/10.1007/978-3-540-87700-4_81
- Sindhya, K., Ruiz, A. B., & Miettinen, K. (2011). A Preference Based Interactive Evolutionary Algorithm for Multi-objective Optimization: PIE. In R. H. C. Takahashi, K. Deb, E. F. Wanner, & S. Greco (Eds.), *Evolutionary Multi-Criterion Optimization* (Vol. 6576, pp. 212–225). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from http://link.springer.com/10.1007/978-3-642-19893-9_15
- Siskos, Y., & Grigoroudis, E. (2010). New Trends in Aggregation-Disaggregation Approaches. In C. Zopounidis & P. M. Pardalos (Eds.), *Handbook of Multicriteria Analysis* (1st ed., Vol. 103, pp. 189–214). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from http://link.springer.com/10.1007/978-3-540-92828-7_6
- Siskos, Y., Grigoroudis, E., & Matsatsinis, N. F. (2005). UTA Methods. In J. Figueira, S. Greco, & M. Ehrgott (Eds.), *Multiple Criteria Decision Analysis: State of the Art Surveys* (Vol. 78, pp. 297–334). New York: Springer-Verlag. Retrieved from http://link.springer.com/10.1007/0-387-23081-5_8
- Skorupski, J., & Mateas, M. (2010). Novice-friendly Authoring of Plan-based Interactive Storyboards. *Artificial Intelligence and Interactive Digital Entertainment –AIIDE*.
- Smimov, A., Kashevnik, A., Levashova, T., Pashkin, M., & Shilov, N. (2007). Situation Modeling in Decision Support Systems. In *International Conference on Integration of Knowledge Intensive Multi-Agent Systems* (pp. 34–39). IEEE. doi: 10.1109/KIMAS.2007.369781
- Snowdon, R. A., Warboys, B. C., Greenwood, R. M., Holland, C. P., Kawalek, P. J., & Shaw, D. R. (2007). On the architecture and form of flexible process support. *Software Process: Improvement and Practice*, 12(1), 21–34.
- So, R., & Sonenberg, L. (2004). Situation awareness in intelligent agents: foundations for a theory of proactive agent behavior. In *Proceedings of IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT), 2004* (pp. 86–92). IEEE. doi: 10.1109/IAT.2004.1342928
- Soffer, P. (2005). On the Notion of Flexibility in Business Processes. In *Proceedings of CAiSE05 Workshops, Design and Support. Workshop on Business Process Modeling, Design and Support (BPMD505)* (pp. 35–42).
- Soffer, P., Pnina, G., Ghattas, J., & Peleg, M. (2010). A Goal-Based Approach for Learning in Business Processes. In S. Nurcan, C. Salinesi, C. Souveyet, & J. Ralyté (Eds.), *Intentional Perspectives on Information Systems Engineering* (pp. 239–256). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from http://link.springer.com/10.1007/978-3-642-12544-7_13

- Sommerville, I. (2009). *Βασικές Αρχές Τεχνολογίας Λογισμικού*. (Δ. Τσιλογιάννης, Trans.). Εκδόσεις Κλειδάριθμος.
- Spanoudakis, G., & Zisman, A. (2010). Designing and Adapting Service-based Systems: A Service Discovery Framework. In *Service Engineering: European Research Results* (pp. 261–297). Springer.
- Spanoudakis, G., Zisman, A., & Kozlenkov, A. (2005). A service discovery framework for service centric systems. In *Proceedings of the IEEE International Conference on Service Computing* (Vol. 1, pp. 251–259).
- Sreenath, R. M., & Singh, M. P. (2004). Agent-Based Service Selection. *Web Semantics: Science, Services and Agents on the World Wide Web*, 1(3), 261–279.
- Stoitsev, T., Scheidl, S., & Spahn, M. (2007). A framework for light-weight composition and management of ad-hoc business processes. In M. Winckler, H. Johnson, & P. Palanque (Eds.), *TAMODIA 2007* (Vol. 4849, pp. 213–226).
- Sybase. (2010). Complex Event Processing: Ten Design Patterns. Retrieved from http://m.sybase.com/files/White_Papers/CEP-10-Design-Patterns-WP.pdf
- Tapia, E. M., Intille, S. S., & Larson, K. (2004). Activity Recognition in the Home Using Simple and Ubiquitous Sensors. In A. Ferscha & F. Mattern (Eds.), *Pervasive Computing* (Vol. 3001, pp. 158–175). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from http://link.springer.com/10.1007/978-3-540-24646-6_10
- Ter Hofstede, A. H. M., van der Aalst, W. M. P., Adams, M., & Russell, N. (2010). *Modern Business Process Automation: YAWL and its Support Environment*. Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from <http://link.springer.com/10.1007/978-3-642-03121-2>
- Thangarajah, J., Padgham, L., & Sardina, S. (2006). Modelling situations in intelligent agents. In *AAMAS 2006* (p. 1049). ACM Press. doi: 10.1145/1160633.1160819
- Toma, I., Ding, Y., Chalersmook, K., Simperl, E., & Fensel, D. (2009). Utilizing Web2.0 in Web Service Ranking. In *Proceedings of the Third International Conference on Digital Society* (pp. 174–179).
- Truong, B. A., Lee, Y.-K., & Lee, S.-Y. (2005). Modeling uncertainty in context-aware computing. In *4th Annual ACIS International Conference on Computer and Information Science (ICIS)* (pp. 676–681). IEEE. doi: 10.1109/ICIS.2005.89
- Urlings, P., Tweedale, J., Sioutis, C., & Ichalkaranje, N. (2003). Intelligent Agents and Situation Awareness. In V. Palade, R. J. Howlett, & L. Jain (Eds.), *Knowledge-Based Intelligent Information and Engineering Systems* (Vol. 2774, pp. 723–733). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from http://link.springer.com/10.1007/978-3-540-45226-3_99
- Van der Aalst, W., Mooij, A., Stahl, C., & Wolf, K. (2009). Service interaction: Patterns, formalization, and analysis. *LNCS*, 5569, 42–88.
- Van Kasteren, T., & Krose, B. (2007). Bayesian activity recognition in residence for elders. In *3rd IET International Conference on Intelligent Environments (IE 07)* (Vol. 2007, pp. 209–212). IEE. doi: 10.1049/cp:20070370
- Van Kasteren, Tim, Noulas, A., Englebienne, G., & Kröse, B. (2008). Accurate activity recognition in a home setting. In *10th international conference on Ubiquitous computing (UbiComp '08)* (p. 1). ACM Press. doi: 10.1145/1409635.1409637

- Vanderfeesten, I. T. P., Reijers, H. A., & van der Aalst, W. M. P. (2008). Product based workflow support: Dynamic workflow execution. In Z. Bellahsene & M. Leonard (Eds.), *CAiSE 2008* (Vol. 5074, pp. 571–574).
- Verbert, K., Manouselis, N., Ochoa, X., Wolpers, M., Drachsler, H., Bosnic, I., & Duval, E. (2012). Context-Aware Recommender Systems for Learning: A Survey and Future Challenges. *IEEE Transactions on Learning Technologies*, 5(4), 318–335. doi: 10.1109/TLT.2012.11
- Vieira, V., Tedesco, P., & Salgado, A. C. (2011). Designing context-sensitive systems: An integrated approach. *Expert Systems with Applications*, 38(2), 1119–1138. doi: 10.1016/j.eswa.2010.05.006
- Wang, Q., Quan, L., & Ying, F. (2004). Online testing of Web-based applications. In *Proceedings of the 28th Annual International Computer Software and Applications Conference (COMPSAC)* (pp. 166–169).
- Wang, X. (2007). *Study of Ranking Irregularities When Evaluating Alternatives by Using Some ELECTRE Methods and a Proposed New MCDM Method Based on Regret and Rejoicing* (Master's Thesis). Louisiana State University. Retrieved from URN: etd-07112007-012708
- Warfield, I., Hogg, C., Lee-Urban, S., & Munoz-Avila, H. (2007). Adaptation of hierarchical task network plans. In *Proceedings of the Twentieth International FLAIRS Conference (FLAIRS-07)*.
- Webera, B., Reichert, M., & Rinderle-Mab, S. (2008). Change patterns and change support features – Enhancing flexibility in process-aware information systems. *Data & Knowledge Engineering*, 66(3), 438–466.
- Weißberg, N., Gartmann, R., & Voisard, A. (2006). An Ontology-Based Approach to Personalized Situation-Aware Mobile Service Supply. *Geoinformatica*, 10(1), 55–90. doi: 10.1007/s10707-005-4886-9
- White, S. A. (2004, May 3). Business Process Modeling Notation v1.0. Retrieved from http://www.omg.org/bpmn/Documents/BPMN_V1-0_May_3_2004.pdf
- White, S. A. (2006). Process Modeling Notations and Workflow Patterns. Retrieved from http://www.bpmn.org/Documents/Notations_and_Workflow_Patterns.pdf
- Whitestein Technologies AG. (2010). Goal-Oriented Business Process Management. Retrieved from <http://www.whitestein.com/goal-oriented-bpm-suite/goal-oriented-bpmn>
- Williams, G. (1966). *Adaptation and natural selection: a critique of some current evolutionary thought*. Princeton, NJ: Princeton Univ. Press.
- Witteborg, H., Charfi, A., Colomer Collell, D., & Mezini, M. (2014). Weaving Aspects and Business Processes through Model Transformation. In M. Villari, W. Zimmermann, & K.-K. Lau (Eds.), *Service-Oriented and Cloud Computing* (Vol. 8745, pp. 47–61). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from http://link.springer.com/10.1007/978-3-662-44879-3_4
- Wojek, C., Nickel, K., & Stiefelhagen, R. (2006). Activity Recognition and Room-Level Tracking in an Office Environment. In *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems* (pp. 25–30). IEEE. doi: 10.1109/MFI.2006.265608

- Xiao, Z., Cao, D., You, C., & Mei, H. (2011). Towards a Constraint-Based Framework for Dynamic Business Process Adaptation. In *IEEE International Conference on Services Computing (SCC), 2011* (pp. 685–692). IEEE. doi: 10.1109/SCC.2011.95
- Yager, R. R. (1988). On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE Transactions on Systems, Man, and Cybernetics*, 18(1), 183–190. doi: 10.1109/21.87068
- Yang, J.-Y., Wang, J.-S., & Chen, Y.-P. (2008). Using acceleration measurements for activity recognition: An effective learning algorithm for constructing neural classifiers. *Pattern Recognition Letters*, 29(16), 2213–2220. doi: 10.1016/j.patrec.2008.08.002
- Yau, S. S., Huang, D., Gong, H., & Davulcu, H. (2005). Situation-awareness for adaptive coordination in service-based systems. In *Proceedings of the 29th Annual Inter. Computer Software and Appl. Conf.* (pp. 107–112).
- Yau, S. S., & Liu, J. (2006). Hierarchical Situation Modeling and Reasoning for Pervasive Computing. In *4th IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems, and 2nd International Workshop on Collaborative Computing, Integration, and Assurance (SEUS-WCCIA'06)* (pp. 5–10). IEEE. doi: 10.1109/SEUS-WCCIA.2006.25
- Ye, J., Coyle, L., Dobson, S., & Nixon, P. (2007). Using Situation Lattices to Model and Reason about Context. In *4th International Workshop Modeling and Reasoning in Context*. Roskilde, Denmark.
- Ye, Juan, Dobson, S., & McKeever, S. (2012). Situation identification techniques in pervasive computing: A review. *Pervasive and Mobile Computing*, 8(1), 36–66. doi: 10.1016/j.pmcj.2011.01.004
- Yurur, O., Liu, C., Sheng, Z., Leung, V., Moreno, W., & Leung, K. (2014). Context-Awareness for Mobile Sensing: A Survey and Future Directions. *Communications Surveys Tutorials, IEEE, PP(99)*. doi: 10.1109/COMST.2014.2381246
- Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, 8(3), 338–353. doi: 10.1016/S0019-9958(65)90241-X
- Zadeh, L. A. (1975). The concept of a linguistic variable and its application to approximate reasoning - Parts I, II, III. *Information Sciences*, 8(3) 199–249; 8(4) 301–357; 9(1) 43–58. doi: 10.1016/0020-0255(75)90036-5
- Zainol, Z., & Nakata, K. (2010). Generic context ontology modelling: A review and framework (pp. 126–130). IEEE. doi: 10.1109/ICCTD.2010.5646137
- Zhang, D., Cao, J., Zhou, J., & Guo, M. (2009). Extended Dempster-Shafer Theory in Context Reasoning for Ubiquitous Computing Environments. In *International Conference on Computational Science and Engineering* (pp. 205–212). IEEE. doi: 10.1109/CSE.2009.201
- Zheng, Z., Ma, H., Lyu, M. ., & King, I. (2009). WSRec: a collaborative filtering based web service recommendation system. In *Proceedings 7th International Conference on Web Services (ICWS 2009)* (pp. 437–444).
- Zopounidis, C., & Pardalos, P. M. (2010). *Handbook of multicriteria analysis* (1st ed., Vol. 103). Berlin; Heidelberg: Springer-Verlag. Retrieved from <http://public.eblib.com/EBLPublic/PublicView.do?ptilID=571938>

ΠΑΡΑΡΤΗΜΑΤΑ

ΠΑΡΑΡΤΗΜΑ Ι – Παραδείγματα κόμβων SAN σε RDF/N3

Ακολούθως δίνονται παραδείγματα των σημαντικότερων δομικών στοιχείων των SANs σε συντακτικό RDF. Συγκεκριμένα για τους κόμβους Στόχου/Υποστόχου, Κατάστασης, Contextualizer, Συνθήκης Πλαισίου, Στοιχειώδους Ενέργειας.

Root Goal/Subgoal node:

```
:_A_Root_Goal a san:RootGoal ;  
    san:hasSituation    :_A_Situation ;  
    san:hasContextCondition  :_A_context_condition ;  
    san:hasAction    :_An_action ;  
    san:auto-start    "yes"^^xsd:string ;  
    san:name    "A sample root goal"^^xsd:string .
```

Situation node:

```
:_A_Situation a san:Situation ;  
    san:dialect    "default"^^xsd:string ;  
    san:defined-by    "  
        http://streams.event-processing.org/ids/VesselStream  
    "^^xsd:string ;  
    san:hasContextualizer :_A_contextualizer;  
    san:name    "A sample situation"^^xsd:string .
```

Situation node – Contextualizer:

```
:_A_contextualizer a san:Contextualizer ;  
    san:type    "RDF"^^xsd:string ;  
    san:contextualizer-query :_A_contextualizer_Qry1 .
```

```

: _A_contextualizer_Query a san:ContextualizerQuery ;
  san:language "SPARQL"^^xsd:string ;
  san:context "LOCAL"^^xsd:string ;
  san:dialect "default"^^xsd:string ;
  san:defined-by "
    SELECT ?uri ?mmsi ?tm
    WHERE { ?uri :MMSI ?mmsi . ?uri t:endTime ?tm }
  "^^xsd:string .

```

Context Condition:

```

: _A_context_condition a san:ContextCondition ;
  san:dialect "JS"^^xsd:string ;
  san:defined-by "/* JS code */
    function check() {
      /* some javascript code */
      return true;
    }
    check();
  "^^xsd:string ;
  san:name "A sample context condition"^^xsd:string .

```

Primitive Action:

```

: _An_action a san:PrimitiveAction ;
  san:command "EXPRESSION"^^xsd:string ;
  san:dialect "JS"^^xsd:string ;
  san:defined-by "/* JS code */
    /* Some javascript code here */
    out.println('Your message...');
  "^^xsd:string ;
  san:name "A sample primitive action"^^xsd:string .

```

ΠΑΡΑΡΤΗΜΑ II – Παράδειγμα κόμβου αφηρημένης ενέργειας & δεξαμενής ενεργειών

Ακολούθως δίνεται ένα πλήρες παράδειγμα κόμβου αφηρημένης ενέργειας μαζί με τη δεξαμενή ενεργειών που χρησιμοποιεί. Το συντακτικό που ακολουθείτε είναι κι εδώ RDF/N3. Για να βελτιωθεί η αναγνωσιμότητα του παραδείγματος, συγκεκριμένα σημεία του έχουν χρωματιστεί, μορφοποιηθεί και σχολιασθεί. Οι αρχές των ορισμών εμφανίζονται με μπλε χρώμα και έντονο τύπο (bold), οι αλφαριθμητικές ακολουθίες (strings) είναι με κόκκινο χρώμα ενώ τα σχόλια με πράσινο.

```
#
# ABSTRACT ACTION NODE DEFINITION
# Must be referenced by a SAN Goal or other valid SAN construct
#

# ABSTRACT ACTION NODE
# a: _Recommend_Sensors
: _Recommend_Sensors      a      san:AbstractAction ;
    #
    # Action pool, Search & Selection and
    # Resolution Policy settings
    #
    san:action-pool "<A static action pool URI>"^^xsd:string ;
    # NOTE: Metadata '_Recommend_Sensors_ACTION_POOL_URI'
    # overrides the above setting
    san:search-method
        "org.iccs.san.util.mcdm.LowaMCDMSearchMethod"^^xsd:string ;
    san:execution-policy "PARALLEL_TIMEOUT"^^xsd:string ;
    san:timeout "180000"^^xsd:string ;
```

```
#
# Metadata overriding Action pool, Search & Selection and
# Resolution Policy settings above (OPTINAL USE)
#
san:hasMetadata :_Recommend_Sensors_ACTION_POOL_URI ;
san:hasMetadata :_Recommend_Sensors_SEARCH_METHOD ;
san:hasMetadata :_Recommend_Sensors_EXEC_POLICY ;

#
# Metadata with LOWA search & selection method settings
#
san:hasMetadata :_Recommend_Sensors_LOAD_ACTION_POOL ;
san:hasMetadata :_Recommend_Sensors_CRITERIA ;
san:hasMetadata :_Recommend_Sensors_ALLOWED_VALUES ;
san:hasMetadata :_Recommend_Sensors_MAPPING ;
san:hasMetadata :_Recommend_Sensors_RESULT_FILTER ;
san:hasMetadata :_Recommend_Sensors_RESULT_COUNT ;
san:hasMetadata :_Recommend_Sensors_RESULT_ORDER ;

#
# Metadata with LOWA criteria and weights
#
san:hasMetadata :Recommend_Sensors_MANUFACTURER ;
san:hasMetadata :Recommend_Sensors_OWNER ;
san:hasMetadata :Recommend_Sensors_THROUGHPUT ;
san:hasMetadata :Recommend_Sensors_AGE ;
san:hasMetadata :Recommend_Sensors_ACCURACY ;

#
# Other abstract action node parameters
#
san:name "Recommend Subscription to Radiation
Sensors"^^xsd:string .
```

```

# *****
# ***** ABSTRACT ACTION METADATA *****
# *****

#
# DYNAMIC ACTION POOL RETRIEVAL and REFERENCING
#

# Metadata overriding Action Pool setting
# Specifies expected action pool URI
    :_Recommend_Sensors_ACTION_POOL_URI a san:Metadata ;
    san:name "action-pool"^^xsd:string ;
    san:type "expression"^^xsd:string ;
    san:dialect "default"^^xsd:string ;
    san:defined-by "http://imu.ntua.gr/san/uc/nuclear/keep-
informed/1#_SENSORS_ACTION_POOL_%APID%"^^xsd:string .

# Query an external web service that generates action pool definition
# and load RDF returned into RDF repository.
# Returned content MUST contain a valid Action Pool definition
# identified with a URI identical to that specified at metadata
# "action-pool" (see above) or at 'san:action-pool' setting
#
    :_Recommend_Sensors_LOAD_ACTION_POOL a san:Metadata ;
    san:name "load-resource"^^xsd:string ;
    san:type "expression"^^xsd:string ;
    san:dialect "default"^^xsd:string ;
# san:defined-by "+++ use a local file +++"^^xsd:string .
    san:defined-by "http://10.10.254.111/keep-informed/sensors-
query.php?lat=%LOCAL:lat%&lon=%LOCAL:lon%&range=%RANGE%%LOCAL:range%&
base=_SENSORS_ACTION_POOL_%APID%&format=.n3"^^xsd:string .
    # NOTE: placeholders (%XXX%) are replaced using values from
    # context elements named as 'XXX'

#
# LOWA specific metadata (general settings)
#

# LOWA Criteria (list of criteria identifiers)
    :_Recommend_Sensors_CRITERIA a san:Metadata ;
    san:name "LOWA CRITERIA"^^xsd:string ;
    san:type "string"^^xsd:string ;
    san:value "Manufacturer;Owner;Throughput;Age;Accuracy"^^xsd:string.

```

```

# Allowed linguistic values (list of values)
  :_Recommend_Sensors_ALLOWED_VALUES      a      san:Metadata ;
  san:name    "Allowed Values"^^xsd:string ;
  san:type    "string"^^xsd:string ;
  san:value   "UNACCEPTABLE;POOR;ACCEPTABLE;GOOD"^^xsd:string .

# 'Actual value-to-linguistic term' mappings
# Defined per criterion
  :_Recommend_Sensors_MAPPING      a      san:Metadata ;
  san:name    "Mapping"^^xsd:string ;
  san:type    "string"^^xsd:string ;
  san:value   "
                Manufacturer=      MANUFACTURER #2:ACCEPTABLE,
MANUFACTURER #1:GOOD, MANUFACTURER #3:POOR, *:POOR,
UNKNOWN:UNACCEPTABLE
                Owner              =      University:GOOD, Research
Center:GOOD, Company:ACCEPTABLE, Household:POOR, *:UNACCEPTABLE
                Throughput         =      [0..5]:UNACCEPTABLE,
(5..10]:POOR, (10..20]:ACCEPTABLE, (10..inf):GOOD
                Age                 =      [0..2]:GOOD, (2..5]:ACCEPTABLE,
(5..10]:POOR, (10..inf):UNACCEPTABLE
                Accuracy            =      [0..0.1]:GOOD,
(0.1..0.5]:ACCEPTABLE, (0.5..1.0]:POOR, (1.0..inf):UNACCEPTABLE,
*:UNACCEPTABLE
                "^^xsd:string .

# Results filtering- NO FILTER
  :_Recommend_Sensors_RESULT_FILTER      a      san:Metadata ;
  san:name    "Results Filter"^^xsd:string ;
  san:type    "constraint"^^xsd:string ;
  san:dialect "default"^^xsd:string ;
  san:defined-by ""^^xsd:string .

# Results count - TOP 3
  :_Recommend_Sensors_RESULT_COUNT      a      san:Metadata ;
  san:name    "Results Count"^^xsd:string ;
  san:type    "string"^^xsd:string ;
  san:value   "TOP 3"^^xsd:string .

# Results sorting - DESCENDING order
  :_Recommend_Sensors_RESULT_ORDER      a      san:Metadata ;
  san:name    "Ascending Order"^^xsd:string ;
  san:type    "boolean"^^xsd:string ;
  san:value   "false"^^xsd:string .

```

```
#  
# LOWA specific metadata - Criteria weights  
#  
:Recommend_Sensors_MANUFACTURER a san:Metadata ;  
  san:name "Manufacturer"^^xsd:string ;  
  san:type "numeric"^^xsd:string ;  
  san:value "0.1"^^xsd:string .  
  
:Recommend_Sensors_OWNER a san:Metadata ;  
  san:name "Owner"^^xsd:string ;  
  san:type "numeric"^^xsd:string ;  
  san:value "0.3"^^xsd:string .  
  
:Recommend_Sensors_THROUGHPUT a san:Metadata ;  
  san:name "Throughput"^^xsd:string ;  
  san:type "numeric"^^xsd:string ;  
  san:value "0.2"^^xsd:string .  
  
:Recommend_Sensors_AGE a san:Metadata ;  
  san:name "Age"^^xsd:string ;  
  san:type "numeric"^^xsd:string ;  
  san:value "0.1"^^xsd:string .  
  
:Recommend_Sensors_ACCURACY a san:Metadata ;  
  san:name "Accuracy"^^xsd:string ;  
  san:type "numeric"^^xsd:string ;  
  san:value "0.3"^^xsd:string .
```

Ακολουθως δίνεται ένα παράδειγμα δεξαμενής ενεργειών (επίσης σε συντακτικό RDF/N3). Η δεξαμενή αυτή δημιουργήθηκε δυναμικά από μια κατάλληλη υπηρεσία ιστού που έχουμε αναπτύξει. Ο τρόπος χρήσης της υπηρεσίας φαίνεται ακολούθως :

```
http://a.valid.URL/path/sensors-query.php?lat=<decimal>&lon
=<decimal>&range=<in_meters>&base=<any_valid_rdf_identifier>&
format=.n3
```

Το πραγματικό URL που χρησιμοποιήθηκε για να παραχθεί η δεξαμενή ενεργειών ήταν :

```
http://10.10.254.111/keep-informed/sensors-query.php?lat=
43.6085806&lon=7.0594214&range=100000.0&base=_SENSORS_ACTION_P
OOL_id&format=.n3
```

Generated by : Sensors Database Query WS

Generated on : 2013-02-28T07:41:54+00:00

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix san:
<http://imu.ntua.gr/ontologies/san/2011/06/01/san.rdf#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix : <http://imu.ntua.gr/san/uc/nuclear/keep-informed/1#>
.
```

Dynamic Action Pool

```
:_SENSORS_ACTION_POOL_id          a san:ActionPool ;
```

```
#
```

```
# Action Pool common metadata declarations
```

```
#
```

```
san:hasMetadata :_SENSORS_ACTION_POOL_id__MANUFACTURER ;
```

```
san:hasMetadata :_SENSORS_ACTION_POOL_id__OWNER ;
```

```
san:hasMetadata :_SENSORS_ACTION_POOL_id__THROUGHPUT ;
```

```
san:hasMetadata :_SENSORS_ACTION_POOL_id__AGE ;
```

```
san:hasMetadata :_SENSORS_ACTION_POOL_id__ACCURACY ;
```

```
#
```

```
# Action Pool items declarations
```

```
#
```

```
san:hasAction:_SENSORS_ACTION_POOL_id__SENSOR_1 ;
```

```
san:hasAction:_SENSORS_ACTION_POOL_id__SENSOR_2 ;
```

```
san:hasAction:_SENSORS_ACTION_POOL_id__SENSOR_3 ;
```

```
san:hasAction:_SENSORS_ACTION_POOL_id__SENSOR_5 ;
```

```
san:hasAction:_SENSORS_ACTION_POOL_id__SENSOR_6 ;
```

```
san:hasAction:_SENSORS_ACTION_POOL_id__SENSOR_7 ;
```



```
#
# Action Pool name
#
  san:name "Action Pool generated by EventCloud query WS. Contains
  Sensor recommendations."^^xsd:string .

# Action Pool COMMON METADATA definitions

:_SENSORS_ACTION_POOL_id__MANUFACTURER      a      san:Metadata ;
  san:name "Manufacturer"^^xsd:string ;
  san:type "string"^^xsd:string ;
  san:value ""^^xsd:string .

:_SENSORS_ACTION_POOL_id__OWNERa      san:Metadata ;
  san:name "Owner"^^xsd:string ;
  san:type "string"^^xsd:string ;
  san:value ""^^xsd:string .

:_SENSORS_ACTION_POOL_id__THROUGHPUT a      san:Metadata ;
  san:name "Throughput"^^xsd:string ;
  san:type "numeric"^^xsd:string ;
  san:value ""^^xsd:string .

:_SENSORS_ACTION_POOL_id__AGE a      san:Metadata ;
  san:name "Age"^^xsd:string ;
  san:type "numeric"^^xsd:string ;
  san:value ""^^xsd:string .

:_SENSORS_ACTION_POOL_id__ACCURACY      a      san:Metadata ;
  san:name "Accuracy"^^xsd:string ;
  san:type "numeric"^^xsd:string ;
  san:value ""^^xsd:string .
```

```

# -----
# Action Pool ALTERNATIVE ITEMS definitions
# -----

# SENSOR : 1      (distance: 0)

:_SENSORS_ACTION_POOL_id_SENSOR_1  a      san:PrimitiveAction ;
    san:command  "EXPRESSION"^^xsd:string ;
    san:dialect  "JS"^^xsd:string ;
    san:defined-by      "
        function sendRecom() {
            // Code has been removed
        }
        sendRecom();
    "^^xsd:string ;
#
san:hasMetadata :_SENSORS_ACTION_POOL_id_SENSOR_1__MANUFACTURER
;
san:hasMetadata :_SENSORS_ACTION_POOL_id_SENSOR_1__OWNER ;
san:hasMetadata :_SENSORS_ACTION_POOL_id_SENSOR_1__THROUGHPUT ;
san:hasMetadata :_SENSORS_ACTION_POOL_id_SENSOR_1__AGE ;
san:hasMetadata :_SENSORS_ACTION_POOL_id_SENSOR_1__ACCURACY ;
#
san:name "Recommend subscription to Sensor #1 event
stream"^^xsd:string.

# METADATA of SENSOR : 1
:_SENSORS_ACTION_POOL_id_SENSOR_1__MANUFACTURER a
san:Metadata;
    san:name "Manufacturer"^^xsd:string ;
    san:type "string"^^xsd:string ;
    san:value "MANUFACTURER #1"^^xsd:string .

:_SENSORS_ACTION_POOL_id_SENSOR_1__OWNER  a      san:Metadata ;
    san:name "Owner"^^xsd:string ;
    san:type "string"^^xsd:string ;
    san:value "University"^^xsd:string .

:_SENSORS_ACTION_POOL_id_SENSOR_1__THROUGHPUT  a
san:Metadata ;
    san:name "Throughput"^^xsd:string ;
    san:type "string"^^xsd:string ;
    san:value "15"^^xsd:string .

:_SENSORS_ACTION_POOL_id_SENSOR_1__AGE      a      san:Metadata ;
    san:name "Age"^^xsd:string ;
    san:type "string"^^xsd:string ;
    san:value "8"^^xsd:string .

```

```

: _SENSORS_ACTION_POOL_id__SENSOR_1__ACCURACY a
san:Metadata ;
    san:name "Accuracy"^^xsd:string ;
    san:type "string"^^xsd:string ;
    san:value "0.7"^^xsd:string .

# -----

# SENSOR : 2 (distance: 67842)

: _SENSORS_ACTION_POOL_id__SENSOR_2 a san:PrimitiveAction ;
    san:command "EXPRESSION"^^xsd:string ;
    san:dialect "JS"^^xsd:string ;
    san:defined-by "
        function sendRecom() {
            // Code has been removed
        }
        sendRecom();
    "^^xsd:string ;
#
san:hasMetadata : _SENSORS_ACTION_POOL_id__SENSOR_2__MANUFACTURER
;
san:hasMetadata : _SENSORS_ACTION_POOL_id__SENSOR_2__OWNER ;
san:hasMetadata : _SENSORS_ACTION_POOL_id__SENSOR_2__THROUGHPUT ;
san:hasMetadata : _SENSORS_ACTION_POOL_id__SENSOR_2__AGE ;
san:hasMetadata : _SENSORS_ACTION_POOL_id__SENSOR_2__ACCURACY ;
#
san:name "Recommend subscription to Sensor #2 event
stream"^^xsd:string.

# METADATA of SENSOR : 2
: _SENSORS_ACTION_POOL_id__SENSOR_2__MANUFACTURER a
san:Metadata ;
    san:name "Manufacturer"^^xsd:string ;
    san:type "string"^^xsd:string ;
    san:value "MANUFACTURER #2"^^xsd:string .

: _SENSORS_ACTION_POOL_id__SENSOR_2__OWNER a san:Metadata ;
    san:name "Owner"^^xsd:string ;
    san:type "string"^^xsd:string ;
    san:value "Company"^^xsd:string .

: _SENSORS_ACTION_POOL_id__SENSOR_2__THROUGHPUT a san:Metadata ;
    san:name "Throughput"^^xsd:string ;
    san:type "string"^^xsd:string ;
    san:value "3"^^xsd:string .

```

```
:_SENSORS_ACTION_POOL_id_SENSOR_2_AGE    a    san:Metadata ;
    san:name    "Age"^^xsd:string ;
    san:type    "string"^^xsd:string ;
    san:value    "4"^^xsd:string .

:_SENSORS_ACTION_POOL_id_SENSOR_2_ACCURACY    a    san:Metadata ;
    san:name    "Accuracy"^^xsd:string ;
    san:type    "string"^^xsd:string ;
    san:value    "0.8"^^xsd:string .

# -----
# ....
# more action pool items for sensors 3, 5, 6, 7 are defined next
# ....
# -----

#EOF
```

ΠΑΡΑΡΤΗΜΑ ΙΙΙ – Προγραμματιστική διεπαφή (API) του SAN Engine

Η προγραμματιστική διεπαφή (API) του SAN Engine δίνεται με τη μορφή Java interfaces. Η τρέχουσα έκδοσή του είναι η 1.1.

```
/*
 * SAN Engine Application Programming Interface, version 1.1
 */
package org.iccs.san.api;

import org.iccs.san.cep.Event;
import org.iccs.san.context.Context;
import org.iccs.san.context.contextualizer.Contextualizer;
import org.iccs.san.util.Configurator;
import org.iccs.san.util.Metadata;

import java.util.Comparator;
import java.util.Enumeration;
import java.util.Iterator;
import java.util.Properties;

public interface AbstractAction extends Action {
    public abstract ActionPool getActionPool();
    public abstract ActionPool getActionPool(String uri);
    public abstract String getSearchMethod();
    public abstract String getExecutionPolicy();

    public abstract void setActionPool(ActionPool pool);
    public abstract void setSearchMethod(String searchMethod);
    public abstract void setExecutionPolicy(String execPolicy);
}
```

```
public interface Action extends SANNode {
    public abstract Expression getExpression();
    public abstract void setExpression(Expression expr);
}

public interface ActionPool extends SANObject {
    public abstract SANNode[] getPoolJobs();
    public abstract void setPoolJobs(SANNode[] jobs);
}

public interface ActionPoolSearchMethod {
    public abstract Iterator<Result> searchAndOrderJobs(ActionPool
actionPool, Context localContext, SANObject caller);

    public static interface Result {
        public abstract SANNode getJob();
        public abstract int getRank();
        public abstract double getRelevance();
        public abstract String getDescription();
    }
}

public interface BreakDecorator extends Decorator {
    public abstract String getMessage();
    public abstract void setMessage(String message);
}

public interface CalculationAction extends Action {
    public abstract Expression getExpression();
    public abstract void setExpression(Expression expr);
}

public interface CEPAT extends SANObject {
    public abstract Expression getDefinition();
    public abstract String[] getInputEventTypes();
    public abstract String[] getOutputEventTypes();
    public abstract void setDefinition(Expression definition);
    public abstract void setInputEventTypes(String[] types);
    public abstract void setOutputEventTypes(String[] types);
}

public interface CompositeAction extends Action {
    public abstract SANNode[] getJobs();
    public abstract void setJobs(SANNode[] jobs);
    public abstract void addJob(SANNode job);
    public abstract SANNode getFirstJob();
    public abstract void setFirstJob(SANNode firstJob);
}
```

```
public interface ConditionDecorator extends Decorator {
    public abstract Expression getCondition();
    public abstract void setCondition(Expression expr);
}

public interface ContextCondition extends SANNode {
    public abstract Expression getExpression();
    public abstract void setExpression(Expression e);
}

public interface CounterDecorator extends Decorator {
    public abstract String getVariable();
    public abstract int getStartCount();
    public abstract int getStep();
    public abstract void setVariable(String varName);
    public abstract void setStartCount(int startValue);
    public abstract void setStep(int step);
}

public interface Decorator extends SANNode {
    public abstract SANNode getJob();
    public abstract String getDecoratorType();
    public abstract void setDecoratorType(String t);
    public abstract void setJob(SANNode job);
}

public interface ExceptionHandlerDecorator extends Decorator {
    public abstract String getExceptionType();
    public abstract SANNode getFailoverJob();
    public abstract void setExceptionType(String exceptionType);
    public abstract void setFailoverJob(SANNode failoverJob);
}

public interface Expression extends SANObject {
    public abstract String getDialect();
    public abstract String getDefinition();
    public abstract Iterator<Parameter> getParameters();
    public abstract Object getValue();
    public abstract void setDialect(String dialect);
    public abstract void setDefinition(String expr);
    public abstract void addParameter(Parameter param);
    public abstract void deleteParameter(Parameter param);
    public abstract void setValue(Object value);
}

public interface FailureDecorator extends Decorator { }
```

```
public interface Goal extends SANNode {
    public abstract Situation getSituation();
    public abstract ContextCondition getContextCondition();
    public abstract SANNode getJob();
    public abstract void setSituation(Situation s);
    public abstract void setContextCondition(ContextCondition cc);
    public abstract void setJob(SANNode job);

    public abstract void stopGoal();
}

public interface LoopDecorator extends Decorator {
    public abstract Expression getLoopExpression();
    public abstract void setLoopExpression(Expression expr);
}

public interface Manager {
    public abstract void notify(String src, String dest, String type,
String msg);
    public abstract void notifyNodeStart(SANNode node, String msg);
    public abstract void notifyNodeEnd(SANNode node, String msg);
    public abstract void notifyEventSend(Event event, String msg);
    public abstract void notifyEventReceive(Event event, String
msg);
}

public interface MountAction extends Action {
    public abstract String getReferencedSAN();
    public abstract void setReferencedSAN(String rootGoal);
}

public interface NotDecorator extends Decorator {
}

public interface ParallelAllAction extends CompositeAction {
}

public interface ParallelAnyAction extends CompositeAction {
}

public interface ParallelTimeoutAction extends CompositeAction {
    public abstract long getTimeout();
    public abstract void setTimeout(long timeout);
}

public interface Parameter extends SANObject {
    public abstract String getSource();
    public abstract void setSource(String source);
}
```



```
public interface PrimitiveAction extends Action {
    public abstract String getCommand();
    public abstract String[] getBoundVarNames();
    public abstract String getBoundVar(String key);
    public abstract void setCommand(String cmd);
    public abstract void setBoundVars(java.util.Hashtable h);
    public abstract void setBoundVar(String key, String value);
}

public interface PrintDecorator extends Decorator {
    public abstract String getMessage();
    public abstract void setMessage(String message);
}

public interface RootGoal extends Goal {
    public static final int EXEC_DEFAULT = 0;
    public static final int EXEC_PERSISTENT = 0;
    public static final int EXEC_ONCE = 1;

    public abstract boolean isAutoStart();
    public abstract void setAutoStart(boolean autoStart);
    public abstract boolean getMultipleInstancesAllowed();
    public abstract void setMultipleInstancesAllowed(boolean allow);
    public abstract int getExecMode();
    public abstract void setExecMode(int execMode);
}

/**
 *   The SAN Execution Engine Factory
 */
public interface SANEngine {
    public abstract void startExecution();
    public abstract void stopExecution();
    public abstract void suspend();
    public abstract void resume();

    public abstract Configurator getConfigurator();
    public abstract void setConfigurator(Configurator configurator);

    // Entity methods
    public abstract SANEntity createEntity(String entityURI);
    public abstract SANEntity createEntity(String entityURI, boolean
autoStartRootGoals);
    public abstract SANEntity destroyEntity(String entityURI);
    public abstract Enumeration<SANEntity> getEntities();

    // Root Goal methods
```

```

    public abstract void startRootGoal(String rootURI, String
entityURI);
    public abstract void stopRootGoal(String rootURI);

    // Helper methods
    public abstract Context getGlobalContext();
    public abstract Object evaluateExpression(Expression expr);
    public abstract Object evaluateExpression(Expression expr,
Context localContext);
}

public interface SANEntity extends SANObject {
    public abstract RootGoal[] getRootGoals();
    public abstract boolean isAutoStart();
}

public interface SANNode extends SANObject {
    public static final int UNSPECIFIED = 0;
    public static final int SUCCESS = 1;
    public static final int FAILURE = 2;
    public static final int ERROR = 3;
    public static final int EXCEPTION = 4;

    public abstract int getOrder();
    public abstract void setOrder(int ord);

    public abstract SANNode getNextJob();
    public abstract void setNextJob(SANNode nextJob);
}

public interface SANObject {
    public abstract String getName();
    public abstract String getObjectURI();
    public abstract String getType();
    public abstract void setName(String name);
    public abstract void setObjectURI(String uri);
    public abstract void setType(String type);

    public abstract String getProperty(String key);
    public abstract void setProperty(String key, String value);

    public abstract Metadata<?> addMetadata(Metadata<?> meta);
    public abstract Metadata<?> replaceMetadata(Metadata<?> metaOld,
Metadata<?> metaNew);
    public abstract Metadata<?> replaceMetadata(String metaOldName,
Metadata<?> metaNew);
    public abstract Metadata<?> removeMetadata(Metadata<?> meta);
    public abstract Metadata<?> removeMetadata(String metaName);
}

```

```

public abstract Metadata<?> getMetadata (String metaName);
public abstract Iterator<Metadata<?>> getAllMetadata ();

public abstract void attachObject (String name, Object object);
public abstract Object detachObject (String name);
public abstract Object retrieveObject (String name);
public abstract Iterator<String> getObjectNames ();
}

public interface SANRepository {
    public abstract SANEntity[] getAutoStartEntities ();
    public abstract SANEntity getEntity (String entityURI);
    public abstract SANEntity[] getEntityByName (String name);
    public abstract RootGoal[] getEntityRootGoals (String entityURI);

    public abstract SANObject[] getAllObjects ();

    public abstract Goal getSAN (String objectURI);
    public abstract SANObject getSANObject (String objectURI);
    public abstract Goal getSANByName (String name);
    public abstract SANObject[] getSANObjectByName (String name);

    public abstract String getSource ();
    public abstract void setSource (String source);
    public abstract void addSource (String source);

    public abstract String getRepositoryContents ();
    public abstract String getRepositoryContents (Object format);
    public abstract Object[] getRepositoryFormats ();

    public abstract Configurator getConfigurator ();
    public abstract void setConfigurator (Configurator configurator);
}

public interface SelectorAction extends CompositeAction {
    public static final int ORDER_UNSPECIFIED = 0;
    public static final int ORDER_SEQUENTIAL = 1;
    public static final int ORDER_RANDOM = 2;
    public static final int ORDER_PROBABLISTIC = 3;
    public abstract int getSelectionMethod ();
    public abstract void setSelectionMethod (int method);
}

public interface SequenceAction extends CompositeAction { }

public interface Situation extends SANNode {
    public static enum Policy {
        UNDEFINED, CONTINUE_ON_EVENT, CONTINUE_ON_CONTEXT_CHANGE
    }
}

```

```
};
public abstract CEPAT getCEPAT();
public abstract void setCEPAT(CEPAT c);
public abstract Contextualizer.Descriptor
getContextualizerDescriptor(Contextualizer ctxlizr);
public abstract Policy getPolicy();
}

public interface SuccessDecorator extends Decorator { }

public interface TimerDecorator extends Decorator {
    public abstract long getTimeout();
    public abstract int getReturnCode();
    public abstract void setTimeout(long msec);
    public abstract void setReturnCode(int returnCode);
}
// EOF
```

