



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ & ΣΥΣΤΗΜΑΤΩΝ
ΠΛΗΡΟΦΟΡΙΚΗΣ

**Σημασιολογικό Πλαίσιο &
Βελτιστοποίηση Ποιότητας
Διαδικτυακών Υπηρεσιών**

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

Δημήτριος Θ. Τσεσμετζής

Αθήνα, Ιούνιος 2008



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ & ΣΥΣΤΗΜΑΤΩΝ
ΠΛΗΡΟΦΟΡΙΚΗΣ

Σημασιολογικό Πλαίσιο & Βελτιστοποίηση Ποιότητας Διαδικτυακών Υπηρεσιών

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

Δημήτριος Θ. Τσεσμετζής

Συμβουλευτική Επιτροπή : Ευστάθιος Δ. Συκάς

Μιλτιάδης Ε. Αναγνώστου

Ελευθέριος Α. Καγιάφας

Εγκρίθηκε από την επταμελή εξεταστική επιτροπή την 31^η Ιουνίου 2008.

.....
Ε. Συκάς
Καθηγητής Ε.Μ.Π.

.....
Μ. Αναγνώστου
Καθηγητής Ε.Μ.Π.

.....
Ε. Καγιάφας
Καθηγητής Ε.Μ.Π.

.....
Ν Μήτρου
Καθηγητής Ε.Μ.Π.

.....
Π. Τσανάκας
Καθηγητής Ε.Μ.Π.

.....
Μ. Θεολόγου
Καθηγητής Ε.Μ.Π.

.....
Π. Δεμέστιχας
Αναπληρωτής Καθηγητής
Πανεπιστημίου Πειραιώς

Αθήνα, Ιούνιος 2008

.....
Δημήτριος Θ. Τσεσμετζής

Διδάκτωρ Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Δημήτριος Θ. Τσεσμετζής, 2008.

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.

Περίληψη

Η εκτενής μελέτη των διαδικτυακών υπηρεσιών και του σημασιολογικού δικτύου (semantic web) οδήγησε στον εντοπισμό αδυναμιών / ελλείψεων των υπάρχουσών σχεδιασμών για οντολογίες παραμέτρων ποιότητας διαδικτυακών υπηρεσιών. Επιπρόσθετα, τα κριτήρια στα οποία στηρίζεται ένας πάροχος διαδικτυακών υπηρεσιών κατά την επιλογή πελατών, αποτελούν ένα ερευνητικό πεδίο μείζονος σημασίας.

Στόχος της παρούσας διατριβής αποτελεί η συμβολή στις ερευνητικές περιοχές των πεδίων αυτών, παρέχοντας πρωτότυπους σχεδιασμούς που αφορούν τις παραμέτρους ποιότητας διαδικτυακών υπηρεσιών. Παράλληλα προτείνονται κατάλληλοι αλγόριθμοι βελτιστοποίησης του συνολικού κέρδους ενός παρόχου διαδικτυακών υπηρεσιών.

Το πρόβλημα που αντιμετωπίζει η παρούσα διατριβή είναι το ακόλουθο: Έστω ένας πάροχος που προσφέρει διαδικτυακές υπηρεσίες, οι οποίες διαθέτουν παραμέτρους ποιότητας στην περιγραφή τους. Οι υπηρεσίες αυτές στηρίζουν την περιγραφή τους αποκλειστικά σε οντολογίες. Λαμβάνοντας ακόμη υπόψη το γεγονός ότι ο πάροχος μπορεί να εξυπηρετήσει συγκεκριμένο αριθμό πελατών που είναι απόρροια του γεγονότος ότι έχει περιορισμένους πόρους, ζητείται να βρεθούν μέθοδοι επιλογής πελατών και υπηρεσιών που θα προσφέρει στους πελάτες του, οι οποίες θα του αποφέρουν το μεγαλύτερο δυνατό κέρδος. Τα κριτήρια επιλογής πελατών / υπηρεσιών που θα εφαρμοστούν, θα στηρίζονται στις παραμέτρους ποιότητας των υπηρεσιών. Οι μέθοδοι βέβαια αυτές είναι εύλογο ότι θα είναι αξιοποιήσιμες και εφαρμόσιμες μόνο σε περιπτώσεις εξάντλησης των διαθέσιμων πόρων του παρόχου.

Συνοπτικά, στην παρούσα διατριβή προτείνονται δύο οντολογίες που αφορούν τις παραμέτρους ποιότητας διαδικτυακών υπηρεσιών: (i) η οντολογία Γλώσσα, η οποία αποτελεί ένα γενικό μοντέλο αναπαράστασης των παραμέτρων αυτών καθώς και των σχέσεων που μπορεί να υπάρχουν μεταξύ τους και (ii) η οντολογία Λεξιλόγιο, η οποία περιλαμβάνει τις παραμέτρους ποιότητας που συναντώνται πιο συχνά σε διαδικτυακές υπηρεσίες. Εξάλλου, η μελέτη της οικογένειας του σακιδίου (knapsack family), η οποία περιλαμβάνει αλγοριθμικά προβλήματα που ανήκουν στην κατηγορία NP – Δύσκολα (NP – Hard), αποτέλεσε τους ακρογωνιαίους λίθους στους οποίους στηρίχθηκε ο σχεδιασμός ευρυστικών αλγορίθμων. Στόχος των προτεινόμενων αλγορίθμων *SMCK* είναι η μεγιστοποίηση του συνολικού κέρδους των προσφερόμενων υπηρεσιών από τη μεριά ενός παρόχου. Επισημαίνεται ότι σε πρώτο στάδιο γίνεται μια αξιολόγηση των παραμέτρων ποιότητας – που περιλαμβάνονται στην οντολογία Λεξιλόγιο – που μπορούν να χρησιμοποιηθούν στους προτεινόμενους αλγόριθμους. Αυτό οφείλεται στο γεγονός ότι το πρόβλημα προσεγγίζεται από τη μεριά του παρόχου και τη διαθεσιμότητα των πόρων του, με αποτέλεσμα το ενδιαφέρον να εστιάζεται

στις παραμέτρους εκείνες που η τιμή τους επηρεάζεται με τη μεταβολή του πλήθους των χρηστών / αιτήσεων.

Λέξεις κλειδιά: Διαδικτυακές Υπηρεσίες, Σημασιολογικό Δίκτυο, Ποιότητα Υπηρεσίας, Οντολογίες, Αλγόριθμοι, Ακέραιος Προγραμματισμός, Οικογένεια Σακιδίου, Βελτιστοποίηση Κέρδους, SMCK

Abstract

The extensive study of semantic web identified the lack of QoS (Quality of Service) Ontologies for web services. Furthermore, the criteria a provider is based on, for the selection of its customers is another critical concept. This thesis attempts to resolve both these issues with the provision of innovative design for QoS Ontologies for web services and selection algorithms for the maximization of the provider's profit.

The case study of the general problem that this thesis deals with, is: Consider a web services provider, where the web services' characteristics (including QoS parameters) are described via Ontologies. The provider is able to serve only a limited amount of customers due to limited resources. Thus, a methodology is required for the selection of customers and the provided services in order to maximize its profit, where the selection criteria will be based on QoS parameters. Obviously, this procedure is valid only in cases where the provider falls short of resources.

Specifically, two QoS ontologies for web services are designed: (i) A Language Ontology, which is a general model for a QoS parameter and the characteristics that may have and (ii) a Vocabulary Ontology, which consists of the QoS parameters that are most frequently used for a web service. The design of the - heuristic - selection algorithms is based on solutions for the knapsack family, a family which has well-known NP-Hard problems. The proposed *SMCK* algorithms attempt to maximize provider's profit. It should be mentioned, that a filtering of the QoS parameters that are applicable for these algorithms is taking place initially, as not all these parameters are correlated with the provider's resources.

Keywords: Web Services, Semantic Web, Quality of Service, Ontologies, Algorithms, Integer Programming, Knapsack Family, Profit Maximization, SMCK

Ευχαριστίες

Η ολοκλήρωση της διατριβής μου πραγματώνεται με τη συγγραφή αυτής της παραγράφου. Κάνοντας μια σύντομη αναδρομή σε όλα τα στάδια από τα οποία πέρασα προκειμένου να φτάσω σε αυτό το σημείο, κρίνω απαραίτητο να ευχαριστήσω αρκετούς ανθρώπους που με τον έναν ή τον άλλο τρόπο συντέλεσαν στην πορεία που ακολούθησα όλα αυτά τα χρόνια.

Ευχαριστώ θερμά τον επιβλέποντα καθηγητή μου κ. Ευστάθιο Συκά για δύο λόγους: αφενός για την εμπιστοσύνη που μου έδειξε, επιλέγοντάς με ως υποψήφιο διδάκτορα και αφετέρου για όλη τη συμπαράσταση και καθοδήγηση που μου πρόσφερε απλόχερα σε όλα τα στάδια της διατριβής μου. Ιδιαίτερες ευχαριστίες στον καθηγητή μου κ. Μιλτιάδη Αναγνώστου, τόσο για τις πολύτιμες υποδείξεις του όσον αφορά τη διατριβή μου όσο και για την ευκαιρία που μου έδωσε να εργαστώ στο ευρωπαϊκό ερευνητικό πρόγραμμα AMIGO. Ευχαριστώ επίσης τον καθηγητή μου κ. Νικόλαο Μήτρου που μου έδωσε τη δυνατότητα να ασχοληθώ με τα peer to peer (P2P) συστήματα, στα πλαίσια του ερευνητικού προγράμματος διαχείρισης και δυναμικής διανομής περιεχομένου. Ευχαριστώ ακόμη τους καθηγητές κ. Ελευθέριο Καγιάφα, κ. Παναγιώτη Τσανάκα, κ. Μιχάλη Θεολόγου και κ. Παναγιώτη Δεμέστιχα που μου έκαναν την τιμή να συμμετέχουν στην επταμελή επιτροπή εξέτασης της διατριβής μου. Τις πολύτιμες συμβουλές και συζητήσεις που είχα με τον καθηγητή μου κ. Μιχάλη Καζαντζάκη θα τις θυμάμαι πάντα με νοσταλγία.

Ιδιαίτερες ευχαριστίες στην Ιωάννα Ρουσάκη και στο Γιάννη Παπαϊωάννου για την αμέριστη συμπαράσταση και βοήθειά τους σε όλα τα στάδια της διατριβής μου. Είναι βέβαιο ότι χωρίς τη συμβολή τους, η διατριβή αυτή θα είχε σίγουρα διαφορετική μορφή.

Θα θυμάμαι πάντα την παρέα από το εργαστήριο ISO και τις εποικοδομητικές / χαλαρωτικές συζητήσεις που γινόντουσαν επί παντός θέματος, διευρύνοντας τους ορίζοντές μου προς κάθε κατεύθυνση. Την παρέα αυτή απαρτίζουν ο Άγγελος Τράκος, ο Δημήτρης Νικητόπουλος, ο Μιχάλης Φουκόπουλος, ο Νίκος Παπαουλάκης, ο Φώτης Ανδριτσόπουλος, ο Γιώργος Καρβουνάς, ο Μάριος Δράκος, ο Δημήτρης Καγκλής, ο Χρήστος Χαρόπουλος, ο Αποστόλης Νικολαΐδης και ο Νίκος Λιαμπότης.

Κλείνοντας θα ήθελα να ευχαριστήσω την οικογένειά μου και τους ανθρώπους που με αγαπούν και βρίσκονται δίπλα μου, για όλη την κατανόηση και συμπαράσταση που μου έδειξαν όλα αυτά τα χρόνια. Επιτέλους μπορώ να τους πω ότι πάει και αυτό ☺

Δημήτρης Τσεσμετζής

Ιούνιος 2008

Περιεχόμενα

1. Εισαγωγή	21
2. Διαδικτυακές υπηρεσίες, σημασιολογικό δίκτυο και οντολογίες	27
2.1. Διαδικτυακές υπηρεσίες.....	27
2.1.1. SOAP (Simple Object Access Protocol).....	31
2.1.2. WSDL (Web Service Description Language).....	33
2.1.3. UDDI (Universal Description, Discovery and Integration).....	35
2.2. Σημασιολογικό δίκτυο	36
2.2.1. Στρώμα URI - Unicode.....	39
2.2.2. Στρώμα XML - XSD	40
2.2.3. Στρώμα RDF - RDFS.....	41
2.2.4. Στρώμα Οντολογιών	44
2.2.5. Στρώματα Λογικής (Logic), Απόδειξης (Proof), Εμπιστοσύνης (Trust)	48
2.3. Οντολογίες και Διαδικτυακές Υπηρεσίες	48
2.3.1. Σημασιολογικές Διαδικτυακές Υπηρεσίες.....	49
2.3.2. Χαρακτηριστικά OWL-S	51
2.3.3. Εύρεση Διαδικτυακών Υπηρεσιών	53
3. Οντολογίες ποιότητας υπηρεσίας (QoS)	55
3.1. Ποιότητα υπηρεσίας (QoS) σε διαδικτυακές υπηρεσίες.....	55
3.2. Σχετική βιβλιογραφία	58
3.3. Σχεδιασμός οντολογιών παραμέτρων ποιότητας	64
3.3.1. Βήματα σχεδιασμού οντολογιών	64
3.3.2. Οντολογία Γλώσσα	66
3.3.3. Οντολογία Λεξιλόγιο	72
3.3.4. Αποτίμηση οντολογιών	80
3.3.5. Παράδειγμα εφαρμογής οντολογιών	84
4. Αλγόριθμοι και τεχνικές επίλυσης προβλημάτων που ανήκουν στο knapsack family	92
4.1. Πολυπλοκότητα Αλγορίθμων	93
4.2. Οικογένεια προβλημάτων σακιδίου	95
4.3. 0-1 Knapsack Problem (KP)	100

4.4.	Multiple Choice Knapsack Problem (MCKP)	104
5.	Σχεδιασμός αλγορίθμων επιλογής υπηρεσιών με στόχο τη βελτιστοποίηση του κέρδους.....	109
5.1.	Μοντελοποίηση προβλήματος	110
5.2.	Εφαρμογή προτεινόμενων τεχνικών	113
5.3.	Σχετική βιβλιογραφία	115
5.4.	Γραμμικές και μη γραμμικές παράμετροι.....	124
5.5.	Αλγόριθμος επιλογής υπηρεσιών – Selective Multiple Choice Knapsack (SMCK).....	129
5.6.	Πολλαπλοί περιορισμοί	136
5.7.	Αποτίμηση επίδοσης προτεινόμενου αλγόριθμου	140
6.	Επίλογος.....	155
	Παράρτημα Α	159
	Παράρτημα Β	166
	Βιβλιογραφία	174

Ευρετήριο Σχημάτων

Σχήμα 1-1 Στρώματα σημασιολογικού δικτύου	23
Σχήμα 2-1 Δημοσίευση, αναζήτηση και εκτέλεση διαδικτυακής υπηρεσίας.....	31
Σχήμα 2-2 Δομή SOAP μηνύματος	32
Σχήμα 2-3 Δομή WSDL κειμένου	34
Σχήμα 2-4 Δομή UDDI σελίδας.....	36
Σχήμα 2-5 Στρώματα σημασιολογικού δικτύου	39
Σχήμα 2-6 Ιεραρχία και σχέσεις κλάσεων της γλώσσας RDFS	43
Σχήμα 2-7 Μοντέλο διαδικτυακής υπηρεσίας με OWL - S.....	51
Σχήμα 3-1 Βήματα σχεδιασμού οντολογιών ποιότητας υπηρεσίας.....	66
Σχήμα 3-2 Οντολογία Γλώσσα	71
Σχήμα 3-3 Οντολογία Λεξιλόγιο	80
Σχήμα 3-4 Παράδειγμα εφαρμογής της οντολογίας Γλώσσας σε μια υπηρεσία «Τηλεϊατρικής με πολυμέσα»	88
Σχήμα 3-5 Παράδειγμα εφαρμογής της οντολογίας Λεξιλόγιο σε μια υπηρεσία «Τηλεϊατρικής με πολυμέσα»	91
Σχήμα 4-1 Όροι που απομακρύνονται στο <i>LMCKP</i> σύμφωνα με το 2 ^ο κριτήριο επικράτησης	105
Σχήμα 5-1 Μοντέλο παρόχου σημείο-προς-σημείο για τον υπολογισμό της Καθυστερήσης	126
Σχήμα 5-2 Ποσοστό απόκλισης τελικού κέρδους σε σχέση με το αρχικό όταν η προσφερόμενη Καθυστερήση και το κέρδος είναι ασυσχέτιστα μεταξύ τους και πλήθος πελατών=[35-45]%.....	143
Σχήμα 5-3 Ποσοστό απόκλισης τελικού κέρδους σε σχέση με το αρχικό όταν η προσφερόμενη Καθυστερήση και το κέρδος είναι ασυσχέτιστα μεταξύ τους και πλήθος πελατών=[45-55]%.....	144
Σχήμα 5-4 Ποσοστό απόκλισης τελικού κέρδους σε σχέση με το αρχικό όταν η προσφερόμενη Καθυστερήση και το κέρδος είναι ασυσχέτιστα μεταξύ τους και πλήθος πελατών=[55-65]%.....	144
Σχήμα 5-5 Ποσοστό απόκλισης τελικού κέρδους σε σχέση με το αρχικό όταν η προσφερόμενη Καθυστερήση και το κέρδος έχουν ασθενή σχέση συσχέτισης και πλήθος πελατών=[35-45]%.....	145

Σχήμα 5-6 Ποσοστό απόκλισης τελικού κέρδους σε σχέση με το αρχικό όταν η προσφερόμενη Καθυστερήση και το κέρδος έχουν ασθενή σχέση συσχέτισης και πλήθος πελατών=[45-55]%.....	145
Σχήμα 5-7 Ποσοστό απόκλισης τελικού κέρδους σε σχέση με το αρχικό όταν η προσφερόμενη Καθυστερήση και το κέρδος έχουν ασθενή σχέση συσχέτισης και πλήθος πελατών=[55-65]%.....	146
Σχήμα 5-8 Ποσοστό απόκλισης τελικού κέρδους σε σχέση με το αρχικό όταν η προσφερόμενη Καθυστερήση και το κέρδος έχουν ισχυρή σχέση συσχέτισης και πλήθος πελατών=[35-45]%.....	146
Σχήμα 5-9 Ποσοστό απόκλισης τελικού κέρδους σε σχέση με το αρχικό όταν η προσφερόμενη Καθυστερήση και το κέρδος έχουν ισχυρή σχέση συσχέτισης και πλήθος πελατών=[45-55]%	147
Σχήμα 5-10 Ποσοστό απόκλισης τελικού κέρδους σε σχέση με το αρχικό όταν η προσφερόμενη Καθυστερήση και το κέρδος έχουν ισχυρή σχέση συσχέτισης και πλήθος πελατών=[55-65]%.....	147
Σχήμα 5-11 Κέρδος παρόχου όταν η προσφερόμενη Καθυστερήση και το κέρδος είναι ασυσχέτιστα μεταξύ τους.....	148
Σχήμα 5-12 Κέρδος παρόχου όταν η προσφερόμενη Καθυστερήση και το κέρδος έχουν ασθενή σχέση συσχέτισης	148
Σχήμα 5-13 Κέρδος παρόχου όταν η προσφερόμενη Καθυστερήση και το κέρδος έχουν ισχυρή σχέση συσχέτισης.....	149
Σχήμα 5-14 Κέρδος παρόχου για διάφορες σχέσεις συσχέτισης και πλήθος πελατών=[45-55]%	149
Σχήμα 5-15 Ποσοστό απόκλισης ελάχιστης και μέγιστης Καθυστερήσης για διάφορες σχέσεις συσχέτισης και πλήθους πελατών.....	152
Σχήμα 5-16 Χρονική απόκριση προτεινόμενου αλγορίθμου με επιλεγμένη σταθερή Καθυστερήση για τον πάροχο ίση με D_p και πλήθος υπηρεσιών / πελάτη = 50 για διάφορες σχέσεις συσχέτισης.....	154

Ευρετήριο Πινάκων

Πίνακας 3-1 Μονάδες μέτρησης παραμέτρων ποιότητας	79
Πίνακας 4-1 Σύγκριση χρόνου εκτέλεσης πολυωνυμικής (n^3) και εκθετικής (3^n) συνάρτησης.....	94
Πίνακας 4-2 Αριθμητικό παράδειγμα υπολογισμού βέλτιστης συνολικής αξίας για το 0-1 KP	103
Πίνακας 4-3 Αριθμητικό παράδειγμα υπολογισμού βέλτιστης συνολικής αξίας για το MCKP	107
Πίνακας 4-4 Αριθμητικό παράδειγμα υπολογισμού βέλτιστης συνολικής αξίας για το MCKP με χωρητικότητα $c = 6$	108
Πίνακας 4-5 Αριθμητικό παράδειγμα υπολογισμού βέλτιστης συνολικής αξίας για το MCKP με χωρητικότητα $c = 5$	108
Πίνακας 5-1 Σύγκριση αλγορίθμων επιλογής διαδικτυακών υπηρεσιών παγκόσμιας βιβλιογραφίας	124
Πίνακας 5-2 Παράμετροι ποιότητας υπηρεσίας που συμμετέχουν στη συνθήκη περιορισμού	128
Πίνακας 5-3 Κέρδος παρόχου όταν η προσφερόμενη Καθυστέρηση και το κέρδος είναι ασυσχέτιστα μεταξύ τους.....	142
Πίνακας 5-4 Κέρδος παρόχου όταν η προσφερόμενη Καθυστέρηση και το κέρδος έχουν ασθενή σχέση συσχέτισης	143
Πίνακας 5-5 Κέρδος παρόχου όταν η προσφερόμενη Καθυστέρηση και το κέρδος έχουν ισχυρή σχέση συσχέτισης.....	143
Πίνακας 5-6 Ελάχιστη Καθυστέρηση και πλήθος πελατών που εξυπηρετούνται όταν η προσφερόμενη Καθυστέρηση και το κέρδος είναι ασυσχέτιστα μεταξύ τους	150
Πίνακας 5-7 Ελάχιστη Καθυστέρηση και πλήθος πελατών που εξυπηρετούνται όταν η προσφερόμενη Καθυστέρηση και το κέρδος έχουν ασθενή σχέση συσχέτισης.....	151
Πίνακας 5-8 Ελάχιστη Καθυστέρηση και πλήθος πελατών που εξυπηρετούνται όταν η προσφερόμενη Καθυστέρηση και το κέρδος έχουν ισχυρή σχέση συσχέτισης.....	151
Πίνακας 5-9 Εύρος Ελάχιστης Καθυστέρησης για κάθε σχέση συσχέτισης.....	151
Πίνακας 5-10 Χρονική απόκριση SMCK όταν η προσφερόμενη Καθυστέρηση και το κέρδος είναι ασυσχέτιστα μεταξύ τους.....	153
Πίνακας 5-11 Χρονική απόκριση SMCK όταν η προσφερόμενη Καθυστέρηση και το κέρδος έχουν ασθενή σχέση συσχέτισης	153

Πίνακας 5-12 Χρονική απόκριση <i>SMCK</i> όταν η προσφερόμενη Καθυστέρηση και το κέρδος έχουν ισχυρή σχέση συσχέτισης.....	153
--	-----

Αρκτικόλεξα

[B2B]	Business To Business
[B2C]	Business To Customers
[KP]	Knapsack Problem
[MCKP]	Multiple Choice Knapsack Problem
[MMKP]	Multiple-Choice Multi-Dimension Knapsack Problem
[OWL]	Ontology Web Language
[OWL-S]	Ontology Web Language for Web Services
[QoS]	Quality of Service
[RDF]	Resource Description Framework
[RDFS]	Resource Description Framework Schema
[SOA]	Service Oriented Architecture
[SOAP]	Simple Object Access Protocol
[SMCK]	Selective Multiple Choice Knapsack
[UDDI]	Universal Description, Discovery and Integration
[URI]	Uniform Resource Identifier
[URL]	Uniform Resource Locator
[WSDL]	Web Service Description Language
[WWW]	World Wide Web
[XML]	eXtensible Markup Language

Κεφάλαιο 1ο

1. Εισαγωγή

Η εξάπλωση του παγκόσμιου ιστού (WWW) έχει ανοίξει νέους διαύλους επικοινωνίας μεταξύ των ανθρώπων, έχοντας δημιουργήσει παράλληλα νέους τρόπους ανάπτυξης των επιχειρήσεων. Καθημερινά, άνθρωποι από όλον τον κόσμο αναζητούν πληροφορίες για θέματα που τους ενδιαφέρουν, χρησιμοποιούν τις πληροφορίες αυτές σε προσωπικό ή επαγγελματικό επίπεδο, επικοινωνούν και ανταλλάζουν απόψεις με ανθρώπους που μπορεί να βρίσκονται στο άλλο άκρο της γης, ή ακόμα πραγματοποιούν αγορές σε αντικείμενα που τους ενδιαφέρουν.

Ωστόσο, παρά την εξάπλωση και την εκτεταμένη χρήση του παγκόσμιου ιστού από ανθρώπους όλων των ηλικιών, το μεγαλύτερο μέρος της πληροφορίας που υπάρχει σε αυτόν, δε μπορεί να επεξεργαστεί από τις υπολογιστικές μηχανές, αλλά μόνο από τους ανθρώπους. Τις περισσότερες φορές, ο ρόλος των υπολογιστών περιορίζεται στον εντοπισμό της πληροφορίας, στον τρόπο που θα αναπαρασταθεί η πληροφορία, ή στο να συνδέσουν την πληροφορία αυτή με μια άλλη μέσω υπερσυνδέσεων (hyperlinks). Αυτό οφείλεται κατά κύριο λόγο στο γεγονός ότι η πληροφορία που υπάρχει στον παγκόσμιο ιστό είναι σχεδόν σε απόλυτο βαθμό άναρχη. Πράγματι, παρά το γεγονός ότι η πληροφορία βρίσκεται αποθηκευμένη σε βάσεις δεδομένων, με συγκεκριμένη δομή και ιεράρχηση, ωστόσο η πληροφορία αυτή δεν είναι εφικτό να επεξεργαστεί από υπολογιστές, εξάγοντας εκμεταλλεύσιμα στοιχεία, τουλάχιστον στην υπάρχουσα μορφή της [1]. Αυτό γίνεται περισσότερο αντιληπτό, όταν χρησιμοποιήσει κανείς δημοφιλείς μηχανές αναζήτησης όπως είναι το Google, το Yahoo ή το AltaVista. Όσο και αν οι μηχανές αναζήτησης έχουν παίξει καθοριστικό ρόλο στην εξάπλωση και εδραίωση του παγκόσμιου ιστού, παρουσιάζουν ένα σημαντικό μειονέκτημα: Το μεγαλύτερο μέρος των αποτελεσμάτων που επιστρέφουν σε κάθε αναζήτηση είναι λίγο ή σχεδόν καθόλου σχετικό με αυτό που αιτήθηκε ο χρήστης.

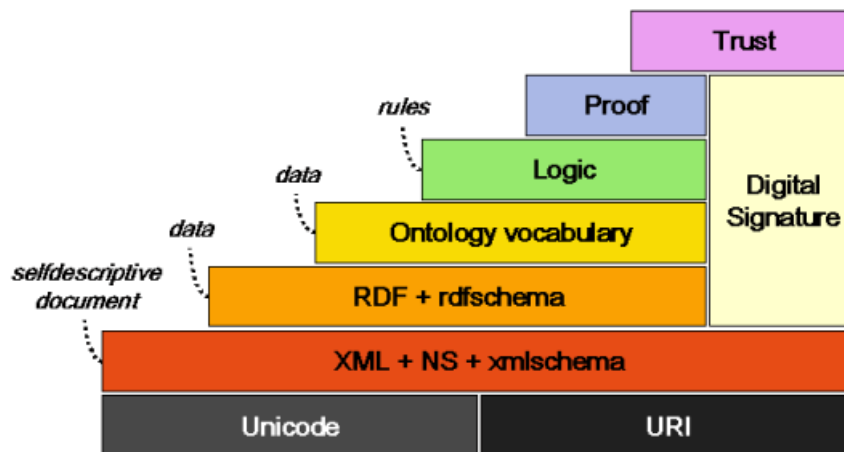
Προκειμένου να γίνει αυτό περισσότερο σαφές, ας υποθεθεί η περίπτωση που κάποιος, χρησιμοποιώντας τις μηχανές αυτές, εισάγει στο πεδίο αναζήτησής τους τις λέξεις συγγραφέας, Παπαδόπουλος και 1965. Τότε τα αποτελέσματα που θα επιστραφούν θα είναι όσα κείμενα περιέχουν τις λέξεις Παπαδόπουλος, συγγραφέας και 1965, χωρίς να προσφέρεται η δυνατότητα φιλτραρίσματος αυτών, ώστε να παραμένουν τελικά αυτά που αφορούν τον συγγραφέα Παπαδόπουλο που γεννήθηκε το 1965, που αποτελεί και το πραγματικό αντικείμενο αναζήτησης του χρήστη. Ακόμη, πολλά αποτελέσματα αναφέρονται σε υπερσυνδέσεις (hyperlinks) που δεν υπάρχουν πια, ενώ πολύ συχνά τα αποτελέσματα αυτά είναι τελείως διαφορετικά, αν χρησιμοποιηθούν στο πεδίο αναζήτησης πολύ συγγενικές (ή συνώνυμες) αλλά ωστόσο διαφορετικές λέξεις. Αυτό οφείλεται στο γεγονός ότι οι μηχανές αυτές μπορούν να εντοπίσουν την πληροφορία στο διαδίκτυο αλλά δε μπορούν να τη διαχειριστούν. Επομένως, θα μπορούσε να ειπωθεί ότι οι μηχανές αυτές είναι περισσότερο μηχανές εντοπισμού της πληροφορίας παρά μηχανές αναζήτησης / διαχείρισης της πληροφορίας. Λέγοντας διαχείριση, εννοείται το αποτέλεσμα της επεξεργασίας των προτάσεων που βρίσκονται σε ένα κείμενο και της εξαγωγής πολύτιμων πληροφοριών, σχετιζόμενων με την αίτηση του εκάστοτε χρήστη [1]. Αιτία του γεγονότος αυτού αποτελεί ότι η πληροφορία που υπάρχει στον παγκόσμιο ιστό δεν παρουσιάζει καμία σχεδόν δομή.

Λύση στο πρόβλημα αυτό προτείνει το σημασιολογικό δίκτυο (semantic web), το οποίο δεν αποτελεί ένα άλλο νέο δίκτυο, το οποίο απορρίπτει τις τεχνικές και μεθοδολογίες που έχουν χρησιμοποιηθεί έως τώρα στον παγκόσμιο ιστό. Αντίθετα, στηρίζει και επεκτείνει τις ήδη υπάρχουσες τεχνολογίες και μεθοδολογίες, ανοίγοντας νέους ορίζοντες και προσφέροντας νέους τρόπους δόμησης και αναζήτησης της πληροφορίας, που θα παρουσιάζουν απολύτως έγκυρα και αξιόπιστα αποτελέσματα.

Προκειμένου να επιτευχθεί αυτός ο πολύ δύσκολος στόχος, είναι απαραίτητο να ακολουθηθούν συγκεκριμένα βήματα. Νέοι τρόποι επικοινωνίας πρέπει να υιοθετηθούν από όλες τις οντότητες που συμμετέχουν στο διαδίκτυο και νέες γλώσσες επικοινωνίας να αναπτυχθούν. Οι γλώσσες αυτές θα πρέπει να δομούν την πληροφορία με τέτοιο τρόπο, ώστε να μπορεί να είναι διαχειρίσιμη από τις υπολογιστικές μηχανές. Απώτερος στόχος και σκοπός είναι η εκμετάλλευση των υπολογιστικών πόρων και δυνατοτήτων της κάθε υπολογιστικής μηχανής του διαδικτύου, που με την πάροδο του χρόνου γίνονται όλο και μεγαλύτερες. Πράγματι, είναι λογικό επακόλουθο ότι δομώντας την πληροφορία στηριζόμενοι σε συγκεκριμένα πρότυπα και χρησιμοποιώντας κοινούς τρόπους επικοινωνίας, να δίνεται η δυνατότητα στις υπολογιστικές μηχανές να εκμεταλλευτούν τις δυνατότητές τους.

Η ανάγκη υιοθέτησης και χρήσης νέων προτύπων δόμησης πληροφορίας οδήγησε στη δημιουργία των οντολογιών. Ο όρος οντολογία [2], αν και χρησιμοποιείται ήδη από το παρελθόν έχοντας φιλοσοφική έννοια με ρίζες στην αρχαία ελληνική γλώσσα (οντολογία: ον και λέγω – κλάδος της μεταφυσικής που μελετά τη φύση, την ουσία των ανθρώπων), ωστόσο

όσον αφορά το σημασιολογικό δίκτυο, αποτελεί τη λύση στην πλήρη δόμηση της πληροφορίας. Νέες γλώσσες αναπτύσσονται και εξελίσσονται, οι οποίες προσφέρουν τη δυνατότητα πλήρους περιγραφής και δόμησης της πληροφορίας που υπάρχει στο διαδίκτυο. Οι γλώσσες αυτές στηρίζονται σε πρότυπα και ακολουθούν συντακτικούς - και όχι μόνο - κανόνες παρόμοιους με αυτούς της γλώσσας XML [3]. Οι σημαντικότερες από αυτές είναι η RDFS (Resource Description Framework Schema) [4] και η OWL (Ontology Web Language) [5]. Ωστόσο, προκειμένου να είναι εφικτή η εκμετάλλευση της δόμησης της πληροφορίας η οποία στηρίζεται στις γλώσσες αυτές, νέες μέθοδοι και τεχνικές πρέπει να υλοποιηθούν, οι οποίες θα δώσουν τη δυνατότητα στις υπολογιστικές μηχανές να διαχειριστούν την πληροφορία που διαθέτουν, χωρίς την παρεμβολή της ανθρώπινης παρουσίας. Οι μέθοδοι αυτές και οι τεχνικές εκμετάλλευσής τους από τις υπολογιστικές μηχανές θα πρέπει να υπακούουν σε συγκεκριμένους κανόνες, όπως θα αναλυθεί εκτενώς στο επόμενο κεφάλαιο. Εν συντομία, η όλη διαδικασία επιτυγχάνεται κάνοντας χρήση τεχνικών διαστρωμάτωσης, με το κάθε στρώμα να υλοποιεί και κάτι διαφορετικό, το οποίο όμως οφείλει να βρίσκεται σε άρρηκτη συνεργασία και αρμονία τόσο με το επόμενο στρώμα όσο και με το προηγούμενό του. Η διαγραμματική αναπαράσταση της διαστρωμάτωσης αυτής σκιαγραφείται στο σχήμα που ακολουθεί [6] (Σχήμα 1-1), ενώ η ανάλυσή της είναι ένα από τα θέματα που θα εξεταστούν εκτενώς στο επόμενο κεφάλαιο.



Σχήμα 1-1 Στρώματα σημασιολογικού δικτύου

Η χρήση των οντολογιών δεν περιορίζεται ωστόσο στη δόμηση και αναζήτηση πληροφορίας. Αντίθετα, έχει επεκταθεί και σε άλλους τομείς, όπως για παράδειγμα στις διαδικτυακές υπηρεσίες (web services) ([7], [8]). Με τον όρο διαδικτυακές υπηρεσίες ορίζονται όλες οι «εφαρμογές οι οποίες είναι ανεξάρτητες, διαθέτουν μεθόδους περιγραφής της λειτουργίας τους, περιλαμβάνουν ένα ή περισσότερα στάδια εκτέλεσης και μπορούν να δημοσιευθούν, εντοπιστούν και εκτελεστούν στο διαδίκτυο. Μια διαδικτυακή υπηρεσία μπορεί να υλοποιεί από την πιο απλή εφαρμογή μέχρι την πιο πολύπλοκη και σύνθετη διεργασία» [9]. Πράγματι, με την εξάπλωση του διαδικτύου, νέες υπηρεσίες έχουν δημιουργηθεί, που

ικανοποιούν ακόμα και τις πιο ιδιαίτερες προτιμήσεις από τη μεριά του πελάτη / χρήστη. Ωστόσο, στηριζόμενοι στα υπάρχοντα πρότυπα και τεχνολογίες, ο τελικός χρήστης δε μπορεί να εκμεταλλευτεί σε ικανοποιητικό βαθμό τις υπηρεσίες που υπάρχουν στον παγκόσμιο ιστό, λόγω των αστοχιών και αδυναμιών που αναφέρθηκαν προηγουμένως. Αυτό έχει σαν αποτέλεσμα τη δημιουργία νέων αρχιτεκτονικών, πρωτοκόλλων και γλωσσών για διαδικτυακές υπηρεσίες, όπως για παράδειγμα SOA [10], SOAP [11], WSDL [12], DAML-S [13] και OWL-S [14].

Βέβαια, η αναζήτηση με ελλιπή αποτελέσματα δεν είναι η μόνη αδυναμία που σχετίζεται με τις διαδικτυακές υπηρεσίες. Μεγαλύτερο μειονέκτημα που παρουσιάζουν και χρήζει άμεσης αντιμετώπισης αποτελεί το γεγονός ότι οι πάροχοι των υπηρεσιών αδυνατούν να προσφέρουν αυτό που τελικά υπόσχονται. Αυτό μπορεί να οφείλεται σε ποικίλες αιτίες μεταξύ των οποίων περιλαμβάνεται ότι τα συστήματά τους δεν είναι κλιμακωτά, με αποτέλεσμα σε χρονικές περιόδους μεγάλης ζήτησης να μην είναι εφικτή η εξυπηρέτηση όλων των πελατών τους. Επιπρόσθετα, πλήθος παρόχων, οι οποίοι προσφέρουν ίδιες ή παραπλήσιες υπηρεσίες έχουν κάνει την εμφάνισή τους, με αποτέλεσμα να μην υπάρχουν σαφή κριτήρια επιλογής για ένα συγκεκριμένο πάροχο από τη μεριά του χρήστη [15]. Οι λόγοι αυτοί οδήγησαν στην εισαγωγή της έννοιας ποιότητας υπηρεσίας QoS (Quality of Service). Παράμετροι που χαρακτηρίζουν την ποιότητα μιας υπηρεσίας όπως διαθεσιμότητα, χωρητικότητα ή ασφάλεια αναλύονται εκτενώς σε επόμενο κεφάλαιο. Αξίζει να επισημανθεί ωστόσο σε αυτό το σημείο, ότι με την προσθήκη των παραμέτρων αυτών στην περιγραφή των υπηρεσιών, δίνεται η δυνατότητα στον πάροχο να διαφημίζει τις υπηρεσίες του με τρόπο που ανταποκρίνεται περισσότερο στην πραγματικότητα. Επιπρόσθετα, προκειμένου να ενισχύσει την ανταγωνιστικότητα και κατά συνέπεια και τα κέρδη του, οδηγείται στην παροχή πιο ανταγωνιστικών και ελκυστικών υπηρεσιών. Από τη μεριά του ο χρήστης / πελάτης της υπηρεσίας στηρίζεται πλέον σε περισσότερα κριτήρια επιλογής μιας συγκεκριμένης υπηρεσίας από έναν συγκεκριμένο πάροχο, η οποία ικανοποιεί σε μεγαλύτερο βαθμό τις ιδιαίτερες προτιμήσεις / απαιτήσεις του. Επομένως, καθίσταται σαφές ότι είναι θέμα μείζονος σημασίας να υποστηρίξει και να περιλαμβάνει η περιγραφή των διαδικτυακών υπηρεσιών παραμέτρους ποιότητας. Οι παράμετροι αυτές οφείλουν να στηρίζονται σε πρότυπα που ανταποκρίνονται στις υπάρχουσες και μελλοντικές ανάγκες των διαδικτυακών υπηρεσιών. Αυτό οδηγεί ανεπιφύλακτα στο συμπέρασμα ότι η περιγραφή των παραμέτρων αυτών πρέπει να στηρίζεται σε γλώσσες που είναι συμβατές με τις υπάρχουσες εξελίξεις του σημασιολογικού δικτύου, δηλαδή να στηρίζουν την περιγραφή τους σε οντολογίες.

Εξάλλου, η περιγραφή των υπηρεσιών που προσφέρει ένας πάροχος με οντολογίες καθώς και των παραμέτρων που χαρακτηρίζουν την ποιότητά τους δεν είναι η μόνη αδυναμία που εντοπίζεται στα υπάρχοντα συστήματα. Ένα άλλο πολύ σημαντικό πρόβλημα που έχει προβάλλει σαν αποτέλεσμα της εξάπλωσης και χρήσης των διαδικτυακών υπηρεσιών και

απορρέει από το γεγονός ότι οι υπολογιστικοί πόροι κάθε παρόχου, όσο μεγάλοι και να είναι, ωστόσο δεν είναι ανεξάντλητοι, είναι το ακόλουθο: Η ανάγκη επιλογής συγκεκριμένων πελατών ή υπηρεσιών που θα προσφέρει ένας πάροχος στους πελάτες του, προκειμένου να μεγιστοποιήσει το κέρδος του, όταν οι πόροι που διαθέτει δεν επαρκούν για να εξυπηρετήσουν το σύνολο των πελατών που έχει τη δεδομένη χρονική στιγμή.

Λύση και στα δύο αυτά κρίσιμα ζητήματα θα προσφέρει η παρούσα διατριβή. Τα προβλήματα που θα επιλυθούν, όπως θα αποδειχθεί και παρακάτω είναι συμπληρωματικά, ενώ η υιοθέτηση των προτεινόμενων λύσεων θα αντιμετωπίσει αποτελεσματικά και διεξοδικά το ακόλουθο γενικό πρόβλημα της επιλογής διαδικτυακών υπηρεσιών στο σημασιολογικό δίκτυο: Έστω ένας πάροχος που προσφέρει διαδικτυακές υπηρεσίες, οι οποίες διαθέτουν παραμέτρους ποιότητας στην περιγραφή τους. Οι υπηρεσίες αυτές στηρίζουν την περιγραφή τους αποκλειστικά σε οντολογίες. Λαμβάνοντας ακόμη υπόψη το γεγονός ότι ο πάροχος μπορεί να εξυπηρετήσει συγκεκριμένο αριθμό πελατών που είναι απόρροια του γεγονότος ότι έχει περιορισμένους πόρους, ζητείται να βρεθούν μέθοδοι επιλογής πελατών και υπηρεσιών που θα προσφέρει στους πελάτες του που θα του αποφέρουν το μεγαλύτερο δυνατό κέρδος. Τα κριτήρια επιλογής πελατών / υπηρεσιών που θα εφαρμοστούν, θα στηρίζονται στις παραμέτρους ποιότητας των υπηρεσιών. Οι μέθοδοι βέβαια αυτές είναι εύλογο ότι θα είναι αξιοποιήσιμες και εφαρμόσιμες μόνο σε περιπτώσεις εξάντλησης των διαθέσιμων πόρων του παρόχου.

Το πρώτο πρόβλημα που πρέπει να επιλυθεί αφορά το σχεδιασμό οντολογιών παραμέτρων ποιότητας, οι οποίες θα είναι έγκυρες και πλήρεις και θα μπορούν να χρησιμοποιηθούν από κάθε διαδικτυακή υπηρεσία, η οποία θα στηρίζει την περιγραφή της σε οντολογίες. Αυτό αποτελεί το αντικείμενο του κεφαλαίου 3, στο οποίο αφού γίνει μια λεπτομερής και εκτενής αναφορά και ανάλυση σχετικών οντολογιών της παγκόσμιας βιβλιογραφίας, προτείνονται στη συνέχεια 2 οντολογίες παραμέτρων ποιότητας υπηρεσίας, οι οποίες θα δώσουν ουσιαστική λύση στο πρόβλημα αυτό. Αξίζει να αναφερθεί ότι ο σχεδιασμός και η ανάλυση των οντολογιών αυτών έχει δημοσιευτεί στις εργασίες [16], [17], [18] και [19], ενώ έχουν γίνει αναφορά και από πλήθος άλλων εργασιών ([20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31]).

Το επόμενο λογικό βήμα που θα συμβάλει στην επίλυση του γενικού προβλήματος επιλογής υπηρεσιών του σημασιολογικού δικτύου είναι η εύρεση των υπηρεσιών που θα ικανοποιούν την κάθε αίτηση / πελάτη. Δηλαδή θα πρέπει να εφαρμοστεί μια διαδικασία κατά την οποία θα εντοπιστούν οι υπηρεσίες από το σύνολο των υπηρεσιών που διαθέτει ένας πάροχος, που εξυπηρετούν τον κάθε πελάτη ξεχωριστά. Αυτή είναι μια διαδικασία που αποτελείται από 2 βήματα: Στο πρώτο βήμα εφαρμόζεται μια διαδικασία φιλτραρίσματος του είδους (π.χ. Βίντεο κατ' απαίτηση – VoD (Video on Demand)) της υπηρεσίας που προσφέρει ένας πάροχος σε σχέση με το είδος της υπηρεσίας που επιθυμεί ο κάθε πελάτης, δεδομένου

ότι κάθε πάροχος προσφέρει διάφορα είδη υπηρεσιών με διαφορετικές παραμέτρους ποιότητας για κάθε ένα από αυτά. Το επόμενο βήμα περιλαμβάνει μια άλλη διαδικασία φιλτραρίσματος κατά την οποία από το σύνολο ενός συγκεκριμένου είδους υπηρεσιών που προσφέρει ένας πάροχος, επιλέγονται οι υπηρεσίες εκείνες οι οποίες ικανοποιούν το επίπεδο ποιότητας που επιθυμεί ο χρήστης. Οι τεχνικές επίλυσης / φιλτραρίσματος που εφαρμόζονται στα βήματα αυτά στηρίζονται στην εφαρμογή συγκεκριμένων μεθόδων ταιριάσματος (matchmaking) [32], που ουσιαστικά αυτό που υλοποιείται τελικά είναι η παράθεση και αντιστοίχιση όλων των ειδών και όλων των παραμέτρων ποιότητας που διαθέτει ο πάροχος με το είδος και τις παραμέτρους ποιότητας της υπηρεσίας που περιλαμβάνονται στην κάθε αίτηση ξεχωριστά. Ωστόσο, επειδή πρόκειται για μια τετριμμένη και απλή διαδικασία, δεν κρίνεται σκόπιμο στην παρούσα διατριβή να προταθούν νέες μέθοδοι που να την υλοποιούν, χωρίς αυτό να σημαίνει ότι είναι μια διαδικασία που δε συμβάλει σημαντικά στην επίλυση του γενικότερου προβλήματος. Γι' αυτό το λόγο άλλωστε αντικείμενο του επόμενου κεφαλαίου αποτελεί μεταξύ άλλων η παρουσίαση και ανάλυση των χαρακτηριστικών στα οποία στηρίζονται οι μέθοδοι αυτές (§2.3.3).

Το επόμενο και τελευταίο βήμα που πρέπει να ακολουθηθεί και θα οδηγήσει τελικά στην επίλυση του γενικού προβλήματος που προαναφέρθηκε, είναι η επιλογή των πελατών που πρέπει να εξυπηρετηθούν αλλά και των υπηρεσιών / πελάτη που πρέπει να προσφερθούν που θα αποφέρει το μέγιστο δυνατό κέρδος στον πάροχο. Η επιλογή αυτή, όπως έχει ήδη αναφερθεί θα στηρίζεται σε κριτήρια που αφορούν τις παραμέτρους ποιότητας των υπηρεσιών. Οι αλγόριθμοι επιλογής διαδικτυακών υπηρεσιών, που παρουσιάζονται και αναλύονται στο κεφάλαιο 5, θα δώσουν τη λύση ακριβώς στο ζήτημα αυτό. Υπενθυμίζεται, ότι οι προτεινόμενοι αλγόριθμοι εφαρμόζονται μόνο σε περιπτώσεις εξάντλησης των διαθέσιμων πόρων του παρόχου. Τέλος, ο σχεδιασμός των αλγορίθμων καθώς και πειράματα που έγιναν προκειμένου να γίνει αποτίμηση της αποτελεσματικότητάς τους, έχουν δημοσιευτεί στις εργασίες [33] και [34].

=====

Κεφάλαιο 2ο

2. Διαδικτυακές υπηρεσίες, σημασιολογικό δίκτυο και οντολογίες

=====

Το κεφάλαιο αυτό αποτελεί μια θεωρητική εισαγωγή στις έννοιες των διαδικτυακών υπηρεσιών, του σημασιολογικού δικτύου και των οντολογιών. Απώτερος στόχος είναι η κατανόηση της σπουδαιότητας που έχει η εφαρμογή των τεχνολογιών αυτών στο διαδίκτυο, οι νέες δυνατότητες που προσφέρονται καθώς και οι μηχανισμοί εκμετάλλευσης αυτών, ώστε να επωφεληθούν όλες οι συμμετέχοντες οντότητες. Επιπρόσθετα, αποτελεί το απαραίτητο θεωρητικό υπόβαθρο για τα κεφάλαια που ακολουθούν, και ιδιαίτερα για το κεφάλαιο 3 στο οποίο σχεδιάζονται 2 οντολογίες ποιότητας (Quality of Service) διαδικτυακών υπηρεσιών.

Πιο αναλυτικά, στην παράγραφο 2.1 παρουσιάζονται οι διαδικτυακές υπηρεσίες, η αρχιτεκτονική σχεδιασμού τους και οι τεχνολογίες στις οποίες έχει στηριχθεί ο σχεδιασμός αυτός. Στην παράγραφο 2.2 παρουσιάζεται το σημασιολογικό δίκτυο, κάνοντας μια σύντομη αναδρομή στο διαδίκτυο, από τα πρώτα στάδια εμφάνισής του μέχρι σήμερα. Αναλύονται τα στρώματα του σημασιολογικού δικτύου, δίνοντας έμφαση στις οντολογίες, που αποτελούν ένα από τα σημαντικότερα θέματα της διατριβής αυτής. Το κεφάλαιο ολοκληρώνεται στην παράγραφο 2.3 όπου παρουσιάζονται οι δυνατότητες συνδυασμού των αντικειμένων των δύο προηγούμενων παραγράφων: των διαδικτυακών υπηρεσιών και των οντολογιών, με απώτερο στόχο την αυτοματοποίηση όσον αφορά την εύρεση, σύνθεση και εκτέλεση υπηρεσιών που προσφέρονται στο διαδίκτυο.

2.1. Διαδικτυακές υπηρεσίες

Με την πάροδο του χρόνου οι υπηρεσίες που προσφέρονται από επιχειρήσεις που δραστηριοποιούνται στον τομέα της πληροφορικής και πιο συγκεκριμένα στο διαδίκτυο

γίνονται όλο και πιο σύνθετες, όλο και πιο πολύπλοκες. Αυτό είναι απόρροια πολλαπλών αιτιών. Τα πληροφοριακά συστήματα των επιχειρήσεων, στην προσπάθειά τους να ακολουθήσουν τους ρυθμούς εξέλιξης της πληροφορικής και να προσαρμοστούν / εναρμονιστούν με τις νέες καταστάσεις, εμπλουτίζουν και ανανεώνουν διαρκώς τις υπηρεσίες που προσφέρουν, λαμβάνοντας υπόψη τις τρέχουσες και μελλοντικές ανάγκες από τη μεριά των πελατών τους. Η προσαρμογή αυτή προϋποθέτει όπωσ είναι εύλογο την υιοθέτηση των γλωσσών και πρωτοκόλλων επικοινωνίας που αναπτύσσονται και χρησιμοποιούνται στη σύγχρονη εποχή. Ωστόσο, επειδή η κάθε επιχείρηση έχει στηριχθεί σε πληροφοριακά συστήματα που εξυπηρετούν τις δικές της ανάγκες ξεχωριστά, είναι αναμενόμενο ότι η πλειονότητα των επιχειρήσεων επιθυμεί την υιοθέτηση προτύπων, η οποία δε θα απορρίπτει τις ήδη υπάρχουσες τεχνικές / μεθοδολογίες που χρησιμοποιεί αυτή. Κάτι τέτοιο βέβαια είναι αναμενόμενο, αφού καμία επιχείρηση δε θα θυσιάζε την τεχνογνωσία και την εμπειρία που έχει αποκτήσει κατά την δραστηριοποίησή της στο χώρο της, και μάλιστα για κάτι εντελώς καινοτόμο και καινούριο, το οποίο στις περισσότερες περιπτώσεις είναι και αμφισβητήσιμης αξιοπιστίας και εγκυρότητας. Η επιθυμία της προσαρμογής της εκάστοτε επιχείρησης δηλαδή, θα έχει πάντα ως γνώμονα την εύρεση λύσης με το μικρότερο δυνατό κόστος και ρίσκο για αυτήν.

Επιπρόσθετα, η ανάπτυξη και εδραίωση του διαδικτύου στις μέρες μας είναι πλέον γεγονός. Η εδραίωση αυτή συνοδεύτηκε και από τη δημιουργία πλήθους παρόχων υπηρεσιών, οι οποίοι προκειμένου να αυξήσουν την ανταγωνιστικότητά τους και κατά συνέπεια τα κέρδη τους προσφέρουν υπηρεσίες που ικανοποιούν κάθε επιθυμία και απαίτηση. Προκειμένου να είναι εκμεταλλεύσιμες οι υπηρεσίες αυτές βέβαια, θα πρέπει να υιοθετηθούν νέοι, ενιαίοι τρόποι επικοινωνίας. Οι τρόποι αυτοί θα δίνουν τη δυνατότητα με έναν καθολικό και απολύτως διαυγή ως προς τους πελάτες της υπηρεσίας τρόπο, να χρησιμοποιούν την υπηρεσία που επιθυμούν χωρίς να τους ενδιαφέρουν οι τεχνικές λεπτομέρειες στις οποίες στηρίχθηκε κατά την ανάπτυξή της η υπηρεσία, δηλαδή να θεωρούν την υπηρεσία ως ένα «μαύρο» κουτί γνωρίζοντας απλά ότι υλοποιεί αυτό που επιθυμούν. Για να χρησιμοποιήσουν βέβαια την εκάστοτε υπηρεσία, θα πρέπει να είναι σε θέση να εντοπίσουν την υπηρεσία που τους ενδιαφέρει και να είναι σε θέση να επικοινωνήσουν με αυτήν.

Από τα παραπάνω είναι έκδηλη η ανάγκη για υιοθέτηση νέων αρχιτεκτονικών, πρωτοκόλλων και γλωσσών επικοινωνίας μεταξύ παρόχων υπηρεσιών και χρηστών – πελατών των υπηρεσιών αυτών, που όμως οφείλει να λάβει υπόψη το γεγονός ότι οι προτεινόμενες λύσεις δε θα πρέπει να προϋποθέτουν την απόρριψη από τη μεριά των επιχειρήσεων των τεχνικών που έχουν αναπτύξει και στηριχθεί, για λόγους που έγιναν σαφείς σε προηγούμενη παράγραφο. Η υιοθέτηση ενός κοινού «κώδικα» επικοινωνίας είναι προς όφελος όλων των συμμετεχουσών οντοτήτων. Από τη μεριά τους οι πάροχοι των υπηρεσιών

θα προσφέρουν τις υπηρεσίες που πρόσφεραν και πριν, αλλά πλέον απευθυνόμενοι σε ένα πολύ μεγαλύτερο μέρος πελατών ενισχύοντας κατ' αυτόν τον τρόπο τα κέρδη τους. Αυτό οφείλεται στο γεγονός ότι δε θα είναι υποχρεωμένοι οι πελάτες πλέον να γνωρίζουν την αρχιτεκτονική των υπηρεσιών του εκάστοτε παρόχου και να προσαρμόζουν το σύστημά τους ώστε να είναι εφικτή η επικοινωνία, διότι κάτι τέτοιο αποθαρρύνει και απομακρύνει υποψήφιους πελάτες. Από τη μεριά τους οι πελάτες μπορούν να επιλέξουν μεταξύ ενός πολύ μεγαλύτερου πλήθους παρόχων υπηρεσιών, οι οποίοι προκειμένου να ενισχύσουν την ανταγωνιστικότητά τους και να προσελκύσουν το ενδιαφέρον μεγαλύτερου πλήθους πελατών, προσφέρουν υπηρεσίες που ικανοποιούν ακόμα και τις πιο ιδιαίτερες προτιμήσεις και απαιτήσεις.

Προκειμένου να δοθεί λύση εφαρμόσιμη και αξιόπιστη, η οποία θα χρήζει καθολικής αποδοχής από όλους τους συμμετέχοντες φορείς, το πρώτο βήμα που πρέπει να γίνει, είναι η ανάπτυξη κατάλληλης αρχιτεκτονικής, η οποία θα αποτελέσει τον ακρογωνιαίο λίθο της προσφερόμενης λύσης. Η αρχιτεκτονική αυτή, η οποία ικανοποιεί όλες τις απαιτήσεις που προαναφέρθηκαν, είναι η SOA (Service Oriented Architecture) [10].

Δύο είναι τα κύρια χαρακτηριστικά της αρχιτεκτονικής αυτής: (i) ότι η κάθε διεργασία που υλοποιείται από έναν πάροχο θεωρείται ως μια υπηρεσία και (ii) ότι η στοιχειώδης μονάδα που λαμβάνεται υπόψη είναι η υπηρεσία, απομονώνοντας και αποσυνδένοντας κατ' αυτόν τον τρόπο τον τελικό χρήστη από τις τεχνικές λεπτομέρειες όσον αφορά την υλοποίηση της κάθε υπηρεσίας. Αυτό βέβαια είναι πολύ σημαντικό και αποτελεί το μεγαλύτερο πλεονέκτημα της αρχιτεκτονικής SOA, αφού προσφέρει τη δυνατότητα μέσω της διαδικασίας αυτής να επικοινωνούν μεταξύ τους διαφορετικών ειδών λειτουργικά συστήματα, γλώσσες προγραμματισμού κ.α.. Επισημαίνεται, ότι η υπηρεσία δύναται να προσφέρεται είτε απευθείας στον πελάτη / χρήστη είτε σε άλλες υπηρεσίες, κάτι που υποδηλώνει ότι μια υπηρεσία μπορεί να είναι το αποτέλεσμα της άθροισης / σύνθεσης άλλων απλούστερων υπηρεσιών.

Εφαρμογή της αρχιτεκτονικής αυτής συναντάται στις διαδικτυακές υπηρεσίες (web services) ([7], [8]). Ως διαδικτυακές υπηρεσίες ορίζονται όπως αναφέρθηκε και στο προηγούμενο κεφάλαιο οι *«εφαρμογές οι οποίες είναι ανεξάρτητες, διαθέτουν μεθόδους περιγραφής της λειτουργίας τους, περιλαμβάνουν ένα ή περισσότερα στάδια εκτέλεσης και μπορούν να δημοσιευθούν, εντοπιστούν και εκτελεστούν στο διαδίκτυο. Μια διαδικτυακή υπηρεσία μπορεί να υλοποιεί από την πιο απλή εφαρμογή μέχρι την πιο πολύπλοκη και σύνθετη διεργασία»* [9].

Όσον αφορά την υλοποίηση και χρήση / εκτέλεση των διαδικτυακών υπηρεσιών, πρέπει να επισημανθεί ότι οι υπηρεσίες αυτές υποστηρίζουν μια στοίβα πρωτοκόλλων [35]:

- Πρωτόκολλα τα οποία είναι υπεύθυνα για τη μεταφορά μηνυμάτων μεταξύ των υπηρεσιών. Κυριότεροι εκπρόσωποι της κατηγορίας αυτής είναι το HTTP,

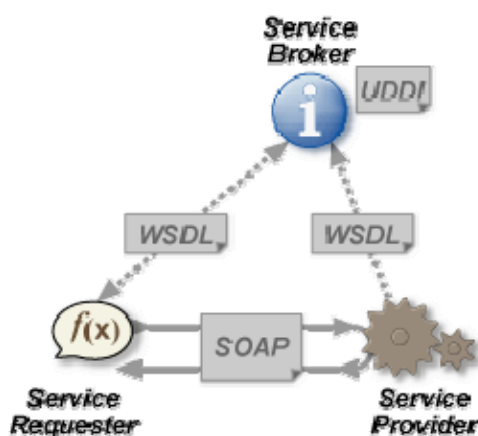
το SMTP, το FTP, ενώ πιο πρόσφατη υλοποίηση είναι το BEEP (Blocks Exchange Extensible Protocol) .

- Πρωτόκολλα τα οποία είναι υπεύθυνα για την κωδικοποίηση και μορφοποίηση των μηνυμάτων επικοινωνίας μεταξύ πελάτη και παρόχου σε XML. Κυριότεροι εκπρόσωποι της κατηγορίας αυτής είναι το XML-RPC, το SOAP και το WS-Addressing.
- Πρωτόκολλα τα οποία είναι υπεύθυνα για την περιγραφή της λειτουργικότητας της υπηρεσίας. Κυριότερος εκπρόσωπος της κατηγορίας αυτής είναι το WSDL.
- Πρωτόκολλα τα οποία είναι υπεύθυνα για τη δημοσίευση και τον εντοπισμό των διαδικτυακών υπηρεσιών, μέσω καταχωρήσεων σε κεντρικές μονάδες εγγραφών (registries). Σε αυτή την κατηγορία χρησιμοποιούνται συνήθως οι διεπαφές του UDDI.

Προτού αναλυθούν οι σημαντικότεροι εκπρόσωποι των πρωτοκόλλων που αναφέρθηκαν (SOAP, WSDL και UDDI), παρατίθεται ένα τυπικό παράδειγμα εφαρμογής αυτών κατά τη δημοσίευση, αναζήτηση και εκτέλεση μιας διαδικτυακής υπηρεσίας.

Προκειμένου να εκτελεστεί μια διαδικτυακή υπηρεσία είναι απαραίτητη η συμμετοχή δύο οντοτήτων: του παρόχου της υπηρεσίας και του πελάτη της υπηρεσίας (που όπως αναφέρθηκε μπορεί να είναι επίσης υπηρεσία κάποιου άλλου παρόχου). Στο σχήμα που ακολουθεί (Σχήμα 2-1) [36] σκιαγραφείται μια τυπική περίπτωση των βημάτων που περιλαμβάνονται προκειμένου να χρησιμοποιηθεί μια διαδικτυακή υπηρεσία. Αρχικά ο πάροχος της υπηρεσίας (Service Provider), αφού υλοποιήσει την υπηρεσία που επιθυμεί να προσφέρει, θα πρέπει να τη δημοσιεύσει σε μια μονάδα εγγραφών UDDI (Universal Description, Discovery and Integration) [37]. Με την έννοια δημοσίευση εννοείται η δημιουργία ενός κειμένου WSDL (Web Service Description Language) [12] που περιλαμβάνει πληροφορίες για τη λειτουργικότητα της υπηρεσίας καθώς και πως μπορεί κάποιος να τη χρησιμοποιήσει / εκτελέσει. Αξίζει να επισημανθεί ότι τα σημαντικότερα και επικρατέστερα περιβάλλοντα υλοποίησης τέτοιων υπηρεσιών (.NET, Java), δίνουν τη δυνατότητα για αυτόματη δημιουργία πηγαίου κώδικα στηριζόμενα στις περιγραφές της κάθε υπηρεσίας. Αυτό το βήμα είναι απαραίτητο για να είναι εφικτή η αναζήτηση και ο εντοπισμός της υπηρεσίας από πιθανούς πελάτες. Προκειμένου να αναζητήσει κάποιος πελάτης (Service Requester) μια υπηρεσία, θα πρέπει η εφαρμογή που χρησιμοποιεί να δημιουργήσει ένα αντίστοιχο κείμενο WSDL που θα περιέχονται οι πληροφορίες για το είδος της υπηρεσίας που αναζητά. Συνήθως, ο πελάτης απευθύνεται σε κάποιον μεσίτη (Service Broker), ο οποίος αναλαμβάνει την αναζήτηση στις μονάδες εγγραφών UDDI εκ μέρους του πελάτη αυτού. Ο μεσίτης αυτός δεν είναι κάτι άλλο από μια απλή εφαρμογή που συνήθως περιλαμβάνεται σαν

πρόσθετη λειτουργικότητα της εφαρμογής που χρησιμοποιεί ο πελάτης για τη δημιουργία των κειμένων WSDL καθώς και για τη διαχείριση της επικοινωνίας. Εφόσον λοιπόν κατά τη διαδικασία της αναζήτησης, ο μεσίτης εντοπίσει κάποια υπηρεσία που ικανοποιεί τις απαιτήσεις του πελάτη (Service Requester), τότε επιστρέφει σε αυτόν τις πληροφορίες που είναι απαραίτητες προκειμένου ο πελάτης να εντοπίσει τον πάροχο της υπηρεσίας και να επικοινωνήσει μαζί του. Ακολούθως, η εφαρμογή του πελάτη εγκαθιστά σύνδεση με τον πάροχο της υπηρεσίας και ξεκινά η ανταλλαγή μηνυμάτων SOAP (Simple Object Access Protocol) [11], μέσω των οποίων θα επιτευχθεί η εκτέλεση και χρήση της υπηρεσίας. Όπως είναι φανερό, σε κανένα από τα παραπάνω στάδια δεν περιλαμβάνονται συνθήκες περιορισμού όσον αφορά τις τεχνικές λεπτομέρειες που σχετίζονται με το περιβάλλον που αναπτύχθηκε η υπηρεσία. Ο πελάτης της υπηρεσίας δηλαδή τη θεωρεί ως «μαύρο» κουτί και απλά κάνει χρήση της λειτουργικότητάς της, αδιαφορώντας για κάθε άλλη πληροφορία. Αυτό είναι απόρροια του γεγονότος ότι η αρχιτεκτονική που έχει στηριχθεί η υλοποίηση αυτή είναι η SOA, όπου στοιχειώδεις μονάδες της αρχιτεκτονικής όπως ήδη αναφέρθηκε είναι οι υπηρεσίες.



Σχήμα 2-1 Δημοσίευση, αναζήτηση και εκτέλεση διαδικτυακής υπηρεσίας

2.1.1. SOAP (Simple Object Access Protocol)

Σε αυτή την παράγραφο θα παρουσιαστούν τα βασικά χαρακτηριστικά του πρωτοκόλλου SOAP [11]. Ωστόσο, προτού γίνει αναφορά των χαρακτηριστικών αυτών, κρίνεται σκόπιμο να παρουσιαστούν οι λόγοι που χρησιμοποιείται το πρωτόκολλο αυτό σαν το κατεξοχήν πρωτόκολλο επικοινωνίας στις διαδικτυακές υπηρεσίες.

Η αποδοχή του πρωτοκόλλου SOAP ήταν άμεση συνέπεια της εξάπλωσης και εδραίωσης του πρωτοκόλλου HTTP [38] σαν πρωτόκολλο μεταφοράς στο διαδίκτυο. Πράγματι, τα υπάρχοντα πρωτόκολλα επικοινωνίας, όπως DCOM [39], CORBA [40] κ.α. είχαν ένα σημαντικό μειονέκτημα: Εξαιτίας της αρχιτεκτονικής τους, τα πρωτόκολλα μεταφοράς που χρησιμοποιούσαν δεν είχαν τη δυνατότητα να παρακάμψουν τα τείχη

ασφαλείας (firewalls), με αποτέλεσμα η επικοινωνία μεταξύ των διαδικτυακών υπηρεσιών ή μεταξύ πελάτη και παρόχου να μην είναι πάντα εφικτή [35]. Το πρωτόκολλο HTTP ωστόσο αντιμετωπίζει το πρόβλημα αυτό με επιτυχία. Επιπρόσθετα, λόγοι συμβατότητας εμποδίζουν τη χρήση του πρωτοκόλλου HTTP σαν πρωτόκολλο μεταφοράς για τα DCOM και CORBA. Επομένως, ένα άλλο πρωτόκολλο επικοινωνίας έπρεπε να χρησιμοποιηθεί, το οποίο οφείλει να είναι σύμφωνο με τις αρχές της αρχιτεκτονικής SOA που αναφέρθηκαν στην προηγούμενη παράγραφο. Δηλαδή, το πρωτόκολλο αυτό θα πρέπει να μην εμποδίζει την επικοινωνία μεταξύ υπηρεσιών που είναι υλοποιημένες για διαφορετικά λειτουργικά συστήματα, ή αναπτυγμένες σε διαφορετικές γλώσσες προγραμματισμού, ενώ θα πρέπει να στηριχθεί σε κάποια γλώσσα που είναι αποδεκτή και εκμεταλλεύσιμη από όλες τις οντότητες του διαδικτύου. Η γλώσσα που ικανοποιεί τις προδιαγραφές αυτές είναι η XML [3] και το πρωτόκολλο επικοινωνίας το SOAP. Άλλα πρωτόκολλα μεταφοράς που χρησιμοποιεί το SOAP είναι το SMTP [41] και το MQSeries [42]. Ωστόσο, η χρησιμοποίηση του HTTP ως πρωτόκολλο μεταφοράς είναι ο πιο συνήθης τρόπος που συναντάται.

Το πρωτόκολλο αυτό όπως φαίνεται και στο Σχήμα 2-1 χρησιμοποιείται για την επικοινωνία μεταξύ παρόχου και πελάτη υπηρεσίας ή μεταξύ διαδικτυακών υπηρεσιών. Ο τρόπος αυτός - Απομακρυσμένες κλήσεις διεργασιών (Remote Procedure Calls (RPCs)) - κατά τον οποίο ο πελάτης στέλνει μια αίτηση σε έναν εξυπηρετητή και ο εξυπηρετητής του απαντά άμεσα είναι και ο πιο συνήθης τρόπος χρήσης του πρωτοκόλλου αυτού. Αξίζει να επισημανθεί, ότι το SOAP θεωρείται ο διάδοχος του XML – RPC [43], όσον αφορά αυτόν τον τρόπο επικοινωνίας.

Στο Σχήμα 2-2 που ακολουθεί [44], παρατίθεται η γενική μορφή ενός SOAP μηνύματος:

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
  <soap:Header>
    <!-- content of header goes here -->
  </soap:Header>
  <soap:Body>
    <!-- content of body goes here -->
    <soap:Fault>
      <!-- content of fault goes here -->
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

Σχήμα 2-2 Δομή SOAP μηνύματος

Όπως φαίνεται και από το παραπάνω σχήμα, σε ένα τυπικό SOAP μήνυμα δύναται να περιλαμβάνονται τα ακόλουθα στοιχεία (elements):

- Το υποχρεωτικό στοιχείο *Envelope*, το οποίο χαρακτηρίζει αυτό το XML κείμενο ως μήνυμα SOAP, ώστε να μπορεί να επεξεργαστεί ανάλογα από τη μεριά που λαμβάνεται.
- Το μη υποχρεωτικό στοιχείο *Header*, το οποίο χρησιμοποιείται για να μεταφέρει πληροφορίες που σχετίζονται με θέματα πιστοποίησης και διαχείρισης της επικοινωνίας μεταξύ πελάτη και εξυπηρετητή.
- Το υποχρεωτικό στοιχείο *Body*, το οποίο περιλαμβάνει τις πληροφορίες που αφορούν την αίτηση και την απόκριση της υπηρεσίας.
- Το μη υποχρεωτικό στοιχείο *Fault*, το οποίο περιλαμβάνει πληροφορίες για τα λάθη που μπορεί να προκύψουν κατά την επεξεργασία του μηνύματος.

2.1.2. WSDL (Web Service Description Language)

Σε αυτή την παράγραφο θα παρουσιαστούν οι βασικές παράμετροι που μπορούν να απαντηθούν σε ένα κείμενο WSDL [12]. Ωστόσο, προτού γίνει αναφορά των παραμέτρων αυτών, κρίνεται σκόπιμο να γίνει μια σύντομη αναφορά της λειτουργικότητας της γλώσσας αυτής.

Όπως έχει γίνει φανερό από τα παραπάνω, το πρωτόκολλο SOAP ενώ δίνει τη δυνατότητα για επικοινωνία μεταξύ των διαφορετικών υπηρεσιών ή μεταξύ παρόχου και πελάτη σε ένα πολύ υψηλό επίπεδο, επιτρέποντας κατ' αυτόν τον τρόπο σε κάθε εμπλεκόμενη οντότητα να χρησιμοποιεί την τεχνολογία (λειτουργικό, γλώσσα προγραμματισμού κ.α..) που επιθυμεί, ωστόσο δεν περιλαμβάνει κανένα στοιχείο που να σχετίζεται με την περιγραφή της λειτουργικότητας της υπηρεσίας. Η περιγραφή της λειτουργικότητας της υπηρεσίας όμως είναι απαραίτητο να παρέχεται με κάποιο τρόπο, αφού σε διαφορετική περίπτωση δε θα υπάρχουν κριτήρια επιλογής στα οποία θα μπορεί να στηριχθεί ο υποψήφιος πελάτης, ούτε προφανώς θα μπορεί να εντοπίσει την υπηρεσία που επιθυμεί. Αυτός είναι ο ρόλος της γλώσσας WSDL. Αρχικά, πρέπει να επισημανθεί ότι όπως και στην περίπτωση του πρωτοκόλλου SOAP, η γλώσσα WSDL στηρίζει την περιγραφή της υπακούοντας κανόνες μορφοποίησης XML για λόγους συμβατότητας, ταυτόσημους με αυτούς που αναλύθηκαν στην προηγούμενη παράγραφο. Κύριο χαρακτηριστικό της γλώσσας είναι ότι θεωρεί τις διαδικτυακές υπηρεσίες ως μια συλλογή από δικτυακά σημεία τα οποία μπορούν να ανταλλάξουν μεταξύ τους μηνύματα [45]. Επομένως, σε ένα κείμενο WSDL περιγράφονται οι διεπαφές της υπηρεσίας καθώς και ο τρόπος με τον οποίο μπορεί ο υποψήφιος πελάτης να εκτελέσει την υπηρεσία.

Στο Σχήμα 2-3 που ακολουθεί [12], παρατίθεται η γενική μορφή ενός WSDL κειμένου:

```
<?xml version="1.0"?>
<definitions>
  <types>
    definition of types
  </types>
  <message>
    definition of a message
  </message>
  <portType>
    definition of a port
  </portType>
  <binding>
    definition of a binding
  </binding>
  <service>
    definition of a service
  </service>
</definitions>
```

Σχήμα 2-3 Δομή WSDL κειμένου

Όπως φαίνεται και από το παραπάνω σχήμα, σε ένα τυπικό κείμενο WSDL δύναται να περιλαμβάνονται τα ακόλουθα στοιχεία (elements):

- Το υποχρεωτικό στοιχείο *definitions*, το οποίο περιλαμβάνει όλα τα υπόλοιπα στοιχεία του κειμένου, περιλαμβάνει δηλαδή όλες τις πληροφορίες που περιγράφουν τα χαρακτηριστικά των υπηρεσιών.
- Το υποχρεωτικό στοιχείο *types*, το οποίο περιγράφει τον τύπο των δεδομένων που χρησιμοποιούνται από τη διαδικτυακή υπηρεσία. Προκειμένου να εξασφαλιστεί σε μεγαλύτερο βαθμό η «ανεξαρτησία» από λειτουργικά συστήματα, γλώσσες προγραμματισμού κ.α., συνήθως χρησιμοποιούνται οι τύποι δεδομένων που ορίζονται στα σχήματα (schemas) XML.
- Το υποχρεωτικό στοιχείο *message*, το οποίο καθορίζει τα δεδομένα των λειτουργιών (operations). Κάθε στοιχείο *message*, μπορεί να περιλαμβάνει περισσότερα από ένα πεδία, όπου τα πεδία αυτά μπορούν να αντιστοιχούν στις παραμέτρους μιας συνάρτησης ή στα πεδία μιας κλάσης μιας αντικειμενοστραφούς γλώσσας.
- Το υποχρεωτικό στοιχείο *portType*, το οποίο είναι το σημαντικότερο στοιχείο του κειμένου. Περιγράφει τη διαδικτυακή υπηρεσία, τις λειτουργίες που μπορεί να εκτελέσει και τα μηνύματα που ανταλλάσσονται. Επομένως, το στοιχείο αυτό μπορεί να θεωρηθεί ως μια κλάση μιας αντικειμενοστραφούς γλώσσας.

- Το υποχρεωτικό στοιχείο *binding*, το οποίο περιγράφει το πρωτόκολλο (π.χ. SOAP) που θα χρησιμοποιηθεί καθώς και πληροφορίες που σχετίζονται με τη μορφή του κειμένου για κάθε στοιχείο *portType*.
- Το μη υποχρεωτικό στοιχείο *service*, το οποίο δίνει τη δυνατότητα για ομαδοποίηση πολλαπλών διαδικτυακών υπηρεσιών σε ένα κείμενο WSDL.

Αξίζει να επισημανθεί ότι το παραπάνω κείμενο μπορεί να εμπλουτιστεί με νέα στοιχεία (elements), αν αυτό είναι απαραίτητο.

2.1.3. UDDI (Universal Description, Discovery and Integration)

Σε αυτή την παράγραφο θα παρουσιαστούν οι βασικές παράμετροι που μπορούν να απαντηθούν σε μια σελίδα UDDI [37]. Ωστόσο, όπως και στις προηγούμενες παραγράφους, προτού γίνει αναφορά των παραμέτρων αυτών, θα γίνει μια σύντομη περιγραφή της λειτουργικότητας που προσφέρει το UDDI.

Όπως έχει γίνει φανερό από τα παραπάνω, το πρωτόκολλο SOAP δίνει τη δυνατότητα για επικοινωνία μεταξύ διαφορετικών διαδικτυακών υπηρεσιών ή μεταξύ παρόχου και πελάτη, ενώ η γλώσσα WSDL στοχεύει στη δημιουργία κειμένων που περιγράφουν τη λειτουργικότητα της κάθε υπηρεσίας. Ωστόσο, όλα αυτά θα ήταν πρακτικά άχρηστα, αν δεν υπήρχε τρόπος να εντοπιστούν οι υπηρεσίες αυτές. Ένας τρόπος για να επιτευχθεί αυτό είναι η ύπαρξη κεντρικών μονάδων εγγραφών (registries) που θα μπορούν οι πάροχοι των υπηρεσιών να δημοσιεύουν τις υπηρεσίες που θέλουν να προσφέρουν και οι πελάτες των υπηρεσιών να μπορούν να αναζητήσουν τις υπηρεσίες που επιθυμούν. Αυτός είναι και ο ρόλος του UDDI: *Το πρωτόκολλο UDDI ορίζει μεθόδους για δημοσίευση και αναζήτηση / εύρεση διαδικτυακών υπηρεσιών που στηρίζουν την υλοποίηση τους στην αρχιτεκτονική SOA* [37]. Σύμφωνα με τις προδιαγραφές του UDDI, σε κάθε καταχώρηση στις μονάδες εγγραφών που διαθέτει θα πρέπει να περιλαμβάνονται τα ακόλουθα [45]:

- Η πληροφορία που θα παρέχεται για κάθε υπηρεσία καθώς και πως μπορεί να κωδικοποιηθεί αυτή.
- Η διεπαφή ερώτησης και ενημέρωσης της πληροφορίας αυτής στη μονάδα εγγραφών, ώστε να είναι εφικτή η τροποποίησή της αν καταστεί ανάγκη.

Όσον αφορά το είδος της πληροφορίας που καταχωρείται στο UDDI, αυτή είναι ταξινομημένη σε 3 κατηγορίες:

- Στις λευκές σελίδες (white pages), που καταχωρούνται προσωπικά στοιχεία επικοινωνίας του παρόχου.
- Στις κίτρινες σελίδες (yellow pages), που καταχωρούνται πληροφορίες που αφορούν τον πάροχο και το είδος των προσφερόμενων υπηρεσιών.

- Στις πράσινες σελίδες (green pages), που καταχωρούνται τεχνικές λεπτομέρειες που αφορούν τις προσφερόμενες υπηρεσίες.

Στο Σχήμα 2-4 που ακολουθεί [37], παρατίθεται η γενική μορφή μιας σελίδας UDDI:

```
<businessEntity>
  <businessServices>
    <!-- may contain multiple businessService elements -->
    <businessService>
      <bindingTemplates>
        <!-- may contain multiple bindingTemplate elements -->
        <bindingTemplate>
          Contain references to tModel
        </bindingTemplate>
      </bindingTemplates>
    </businessService>
  </businessServices>
</businessEntity>
```

Σχήμα 2-4 Δομή UDDI σελίδας

Όπως φαίνεται και από το παραπάνω σχήμα, σε μια τυπική σελίδα UDDI δύναται να περιλαμβάνονται τα ακόλουθα στοιχεία (elements):

- Το στοιχείο *businessEntity*, το οποίο περιλαμβάνει πληροφορίες για τον πάροχο που προσφέρει την υπηρεσία.
- Το στοιχείο *businessServices*, το οποίο περιλαμβάνει ένα ή περισσότερα στοιχεία *businessService*. Τα στοιχεία αυτά περιγράφουν τη λειτουργικότητα της προσφερόμενης υπηρεσίας.
- Το στοιχείο *bindingTemplates*, το οποίο περιλαμβάνει ένα ή περισσότερα στοιχεία *bindingTemplate*. Τα στοιχεία αυτά περιγράφουν τις τεχνικές λεπτομέρειες που αφορούν την προσφερόμενη υπηρεσία καθώς και τη διεπαφή της. Ακόμη, τα στοιχεία αυτά περιλαμβάνουν πρόσθετα πεδία που αφορούν τις προδιαγραφές με τις οποίες είναι συμβατή η υπηρεσία (tModels).

2.2. Σημασιολογικό δίκτυο

Η εξάπλωση του διαδικτύου και των διαδικτυακών υπηρεσιών εμφάνισε τις αδυναμίες των υπάρχοντων συστημάτων για υιοθέτηση νέων τεχνολογιών και επιχειρηματικών μοντέλων. Πράγματι, όπως έγινε φανερό και στην προηγούμενη παράγραφο, νέα επιχειρηματικά μοντέλα μεταξύ επιχειρήσεων (Business to Business (B2B)) και μεταξύ επιχειρήσεων και πελατών (Business to Customer (B2C)) [46] αναπτύσσονται ή έχουν ήδη αναπτυχθεί. Τα σημαντικότερα προβλήματα που καλούνται να αντιμετωπίσουν τα μοντέλα αυτά είναι η ενοποίηση των συστημάτων τους για επιτυχή και ολοκληρωμένη επικοινωνία μεταξύ τους καθώς και η παροχή μεθόδων «κατανόησης» των δεδομένων του καθενός

συστήματος ξεχωριστά, ώστε να είναι εφικτή η επικοινωνία τους χωρίς να είναι απαραίτητη η παρεμβολή ανθρώπινης παρουσίας. Επιλύοντας τα δύο αυτά πολύ σημαντικά προβλήματα, είναι βέβαιο ότι η αυτοματοποιημένη επικοινωνία μεταξύ των προαναφερθέντων μοντέλων θα είναι πλέον εφικτή. Προτού όμως αναλυθούν οι τεχνολογίες και μέθοδοι που θα δώσουν τη λύση στα προβλήματα αυτά, κρίνεται σκόπιμη η παρουσίαση του διαδικτύου στην παρούσα μορφή του καθώς και η εξέλιξή του από τα πρώτα στάδια εμφάνισής του. Στόχος της παρουσίασης αυτής είναι να καταδειχθούν οι αιτίες αδυναμίας αντιμετώπισης των προβλημάτων που μόλις αναφέρθηκαν.

Όπως είναι γνωστό, προκειμένου να επισκεφτεί κάποιος ένα δικτυακό τόπο είναι απαραίτητο να διαθέτει έναν πλοηγό ιστού (Firefox, Internet Explorer κτλ.), να γνωρίζει τη διεύθυνση URL (Uniform Resource Locator) του τόπου αυτού, ενώ τα δεδομένα που θα ανταλλαχθούν, μεταφέρονται κάνοντας χρήση του πρωτοκόλλου HTTP. Η αναπαράσταση των δεδομένων στον πλοηγό ιστού του χρήστη επιτυγχάνεται χρησιμοποιώντας τη γλώσσα HTML (Hypertext Mark-up Language) [47], μια γλώσσα υπερκειμένου που μπορεί να περιλαμβάνει εκτός από χαρακτήρες ASCII και άλλους τύπους δεδομένων, όπως εικόνες ή άλλα δυαδικά αρχεία. Η γλώσσα HTML ωστόσο είναι μια γλώσσα αναπαράστασης δεδομένων. Η χρησιμότητά της εστιάζεται στη μορφοποίηση του κειμένου, στην τοποθέτηση των εικόνων σε κατάλληλες θέσεις, στην παροχή συνδέσεων με άλλους δικτυακούς τόπους μέσω υπερσυνδέσμων (hyperlinks) και τελικά στην αναπαράσταση του κειμένου με τρόπο αντιληπτό από τους πλοηγούς ιστού. Ο ρόλος της δηλαδή περιορίζεται αυστηρά στον τρόπο που θα εμφανιστούν τα δεδομένα στον πλοηγό του χρήστη, χωρίς να προσφέρει καμία απολύτως δυνατότητα επεξεργασίας των δεδομένων αυτών υπό το πρίσμα της κατανόησης [48].

Καθώς όμως η χρήση του διαδικτύου ολοένα και αυξανόταν, πολύ σύντομα έκαναν την εμφάνισή τους και τα πρώτα προβλήματα σχετιζόμενων με τη στατική αναπαράσταση των δεδομένων από τη γλώσσα HTML. Πράγματι, η στατική αναπαράσταση των δεδομένων εμπόδιζε μεταξύ άλλων την παροχή οποιασδήποτε προσωποποιημένης υπηρεσίας στο χρήστη, αφού το μόνο που μπορούσε να προσφερθεί στα πρωταρχικά στάδια του διαδικτύου ήταν κοινές και καθολικές σε όλους τους χρήστες του υπηρεσίες. Λύση στο πρόβλημα αυτό έφεραν οι σχεσιακές βάσεις δεδομένων (Relational Database Management Systems (RDBMS)) [49] και η εισαγωγή αυτών στις επιχειρήσεις, αποτελώντας πλέον αναπόσπαστο κομμάτι της λογικής (Logic) οποιασδήποτε επιχείρησης. Πρέπει να επισημανθεί βέβαια, ότι η εισαγωγή των βάσεων δεδομένων στις επιχειρήσεις το μόνο πρόβλημα που μπόρεσε να αντιμετωπίσει σε σχέση με αυτά που αναφέρθηκαν είναι η παροχή προσωποποιημένων υπηρεσιών στον κάθε χρήστη, προσφέροντας ένα «δυναμικό» χαρακτήρα στις υπηρεσίες αυτές. Το πρόβλημα δηλαδή της αλλαγής του τρόπου επικοινωνίας μεταξύ των επιχειρήσεων

ή της «κατανόησης» των δεδομένων όπως αυτά αναπαρίστανται από τους πλοηγούς ιστού, παρέμεινε άλυτο.

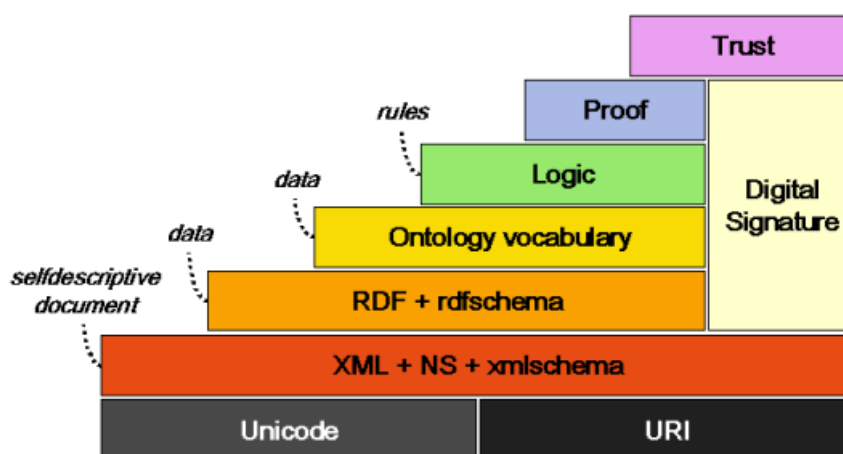
Σημαντικό βήμα για την αλλαγή του τρόπου επικοινωνίας μεταξύ των επιχειρήσεων (B2B) και μεταξύ επιχειρήσεων και πελατών (B2C) έφερε η εμφάνιση και εδραίωση της γλώσσας XML [3]. Πράγματι, με την καθολική αποδοχή της γλώσσας αυτής από όλες τις οντότητες του διαδικτύου, επιτεύχθηκε κάτι πάρα πολύ σημαντικό: η απρόσκοπτη επικοινωνία μεταξύ των οντοτήτων αυτών με έναν ενιαίο και ολοκληρωμένο τρόπο, υπερπηδώντας τα εμπόδια που αδυνατούσε να αντιμετωπίσει η HTML, ακόμα και με τη χρήση σχεσιακών βάσεων δεδομένων. Οι επιχειρήσεις μπορούσαν πλέον να ανταλλάζουν δεδομένα είτε μεταξύ τους, είτε με τους πελάτες τους, χωρίς να είναι απαραίτητη η προσαρμογή των συστημάτων τους. Επιπρόσθετα η αποδοχή και χρήση τεχνολογιών που στηρίζονται στη γλώσσα αυτή επιτάχυνε τη διαδικασία σχεδιασμού και υλοποίησης αυτών των συστημάτων. Αξίζει να επισημανθεί, ότι η επικοινωνία αυτή ενισχύθηκε και με την ανάπτυξη πρόσθετων προτύπων (Document Type Definitions (DTDs), XML Schema Definitions (XSDs)), με κύριο ρόλο τη μορφοποίηση και τον έλεγχο της δομής των δεδομένων XML που ανταλλάσσονται [48].

Ωστόσο, όπως είναι φανερό από τα παραπάνω, η εισαγωγή και υιοθέτηση XML τεχνολογιών μπόρεσε να αντιμετωπίσει μόνο το πρόβλημα της επικοινωνίας των επιχειρηματικών μοντέλων B2B και B2C. Αυτό οφείλεται στο γεγονός ότι η γλώσσα XML καθώς και τα πρότυπά της στοχεύουν στην ορθή - δομημένη σύνταξη των δεδομένων που ανταλλάσσονται και στον έλεγχο αυτής και όχι στη σημασιολογία των δεδομένων. Δηλαδή ακόμα και με την υιοθέτηση των προτύπων αυτών δεν είναι εφικτή η «κατανόηση» των δεδομένων που ανταλλάσσονται.

Λύση στο πρόβλημα αυτό έφερε το σημασιολογικό δίκτυο (Semantic Web): *Το σημασιολογικό δίκτυο δεν είναι ένα ξεχωριστό δίκτυο αλλά μια επέκταση του υπάρχοντος, στο οποίο έχει αποδοθεί σε κάθε πληροφορία σαφή έννοια, διευκολύνοντας με αυτόν τον τρόπο τη συνεργασία μεταξύ ανθρώπων και υπολογιστών* [6]. Όπως φανερώνει και το όνομά του, το δίκτυο αυτό στηρίζεται στη σημασιολογία, δηλαδή στη μελέτη της σημασίας των λέξεων. Η κύρια συμβολή του σημασιολογικού δικτύου είναι ότι στηριζόμενο σε τεχνολογίες που θα αναλυθούν στη συνέχεια, επιτρέπει το διαχωρισμό μεταξύ των δεδομένων και των εννοιών των δεδομένων αυτών [50]. Αυτό βέβαια είναι πάρα πολύ σημαντικό, αφού δίνοντας τη δυνατότητα στους υπολογιστές να διαθέτουν την έννοια των δεδομένων που κατέχουν, μπορούν πλέον να κατανοήσουν τη σημασία ολόκληρων κειμένων. Υπό το πρίσμα αυτό είναι εύλογο ότι θα είναι πλέον εφικτή η αυτοματοποιημένη επεξεργασία των κειμένων αυτών από τους υπολογιστές, καθιστώντας τη συμμετοχή του ανθρώπινου παράγοντα στην επικοινωνία μεταξύ των υπολογιστικών συστημάτων περιττή. Όπως θα φανεί και στη συνέχεια, στο σημασιολογικό δίκτυο η αναπαράσταση της έννοιας των δεδομένων επιτυγχάνεται με χρήση

οντολογιών, ενώ οι σχέσεις - κανόνες που περιέχονται στις οντολογίες αυτές, παίζουν καθοριστικό ρόλο στην κατανόηση της έννοιας αυτής.

Ο σχεδιασμός του σημασιολογικού δικτύου στηρίζεται σε στρώματα, επιτρέποντας την ανεξαρτητοποίηση και το σαφή διαχωρισμό των επιμέρους ενεργειών που πρέπει να ακολουθηθούν, προκειμένου να είναι εφικτή η «κατανόηση» των δεδομένων από τους υπολογιστές. Επιπρόσθετα, «τεμαχίζοντας» το συνολικό πρόβλημα σε μικρότερα, είναι εύλογο ότι είναι πιο εύκολο να βρεθούν λύσεις που το αντιμετωπίζουν. Ανεξάρτητες ομάδες μπορούν να επικεντρωθούν στην επίλυση των επιμέρους προβλημάτων που καλείται να αντιμετωπίσει το κάθε στρώμα ξεχωριστά, ενώ ο συνδυασμός των προσφερόμενων λύσεων από την κάθε ομάδα να οδηγήσει τελικά στην επίλυση του συνολικού προβλήματος που καλείται να αντιμετωπίσει το σημασιολογικό δίκτυο. Ιδιαίτερη προσοχή πρέπει να δοθεί βέβαια όσον αφορά την επικοινωνία μεταξύ των διαφορετικών στρωμάτων, αφού πρέπει να οριστούν σαφείς κανόνες, οι οποίοι θα επιτρέπουν την απρόσκοπτη επικοινωνία των στρωμάτων αυτών είτε προς τα πάνω είτε προς τα κάτω, εμποδίζοντας την εμφάνιση προβλημάτων ασυμβατότητας. Επομένως, η αυτοματοποιημένη επικοινωνία που υπόσχεται το σημασιολογικό δίκτυο είναι αποτέλεσμα συνδυασμού προτύπων (standards) και τεχνολογιών, σαφούς διαστρωμάτωσης των προτύπων και τεχνολογιών αυτών καθώς και σαφών κανόνων επικοινωνίας μεταξύ τους. Στο σχήμα [6] που ακολουθεί σκιαγραφείται η διαγραμματική αναπαράσταση της διαστρωμάτωσης αυτής:



Σχήμα 2-5 Στρώματα σημασιολογικού δικτύου

2.2.1. Στρώμα URI - Unicode

Όπως φαίνεται και από το παραπάνω σχήμα, στη βάση της αρχιτεκτονικής του σημασιολογικού δικτύου βρίσκεται το παγκόσμιο αναγνωριστικό πόρου URI (Universal Resource Identifier) [51]. Το αναγνωριστικό αυτό είναι μια συμβολοακολουθία και δύναται να αναφέρεται σε τοποθεσία, σε όνομα, ή και στα δύο. Στην πρώτη περίπτωση που είναι και η πιο συνηθισμένη, ονομάζεται URL (Uniform Resource Locator) και είναι το αναγνωριστικό

URI το οποίο προσφέρει τρόπους εντοπισμού του πόρου, όπως για παράδειγμα περιγράφοντας τη δικτυακή θέση του (<http://www.ntua.gr>). Στη δεύτερη περίπτωση ονομάζεται URN (Uniform Resource Name) και είναι το αναγνωριστικό URI το οποίο υποχρεούται να μείνει μοναδικό, ακόμα και αν δεν είναι πλέον διαθέσιμο. Χαρακτηριστικό παράδειγμα τέτοιας περίπτωσης είναι το URN κάποιου βιβλίου (isbn:2-343-434343-4). Το πρώτο στρώμα προσφέρει επίσης έναν τρόπο κωδικοποίησης του κάθε χαρακτήρα των δεδομένων που ανταλλάσσονται με την αντιστοίχσή του σε ένα μοναδικό αριθμό (Unicode), ο οποίος είναι ανεξάρτητος από το λειτουργικό σύστημα, το πρόγραμμα ή τη γλώσσα προγραμματισμού που θα χρησιμοποιηθεί. Η συμβολή της κωδικοποίησης αυτής είναι πολύ σημαντική στο συνολικό σχεδιασμό της αρχιτεκτονικής του σημασιολογικού δικτύου. Για να καταστεί αυτό περισσότερο σαφές, αρκεί να αναλογιστεί κανείς ότι σε διαφορετική περίπτωση, θα υπήρχαν πολλοί τρόποι κωδικοποίησης χαρακτήρων (όπως για παράδειγμα ASCII, EBCDIC) και θεωρείται βέβαιο το ενδεχόμενο ασυμβατότητας μεταξύ διαφορετικών λειτουργικών συστημάτων ή γλωσσών προγραμματισμού που μπορεί να οφείλεται είτε στο ότι χρησιμοποιείται ο ίδιος αριθμός για άλλο χαρακτήρα ή και το αντίστροφο, δηλαδή ίδιος χαρακτήρας που κωδικοποιείται από διαφορετικό αριθμό [48].

2.2.2. Στρώμα XML - XSD

Στο επόμενο στρώμα του σημασιολογικού δικτύου, βρίσκεται η γλώσσα XML [3] καθώς και πρότυπα (XSD, DTD) που υιοθετούνται και χρησιμοποιούνται από αυτήν. Όπως έχει ήδη αναφερθεί, η γλώσσα αυτή συνέβαλε σημαντικά στην ανάπτυξη των επιχειρηματικών μοντέλων B2B και B2C [46], εξαιτίας του γεγονότος ότι έδωσε τη δυνατότητα επικοινωνίας μεταξύ των μοντέλων αυτών χωρίς να είναι απαραίτητη η προσαρμογή των συστημάτων τους. Κύριο πλεονέκτημα της γλώσσας αυτής καθώς και των προτύπων της είναι ότι επιτρέπει την αναπαράσταση και τη δόμηση της πληροφορίας με έναν πολύ ευέλικτο και αποτελεσματικό τρόπο. Αυτό επιτυγχάνεται με τη χρήση ετικετών (tags), οι οποίες δεν περιορίζονται σε κάποιες προκαθορισμένες όπως στην περίπτωση της γλώσσας HTML [47], αλλά αντίθετα προσφέρουν τη δυνατότητα δημιουργίας νέων, προκειμένου να αναπαρασταθεί ορθότερα η επιθυμητή πληροφορία. Ο δυναμικός αυτός χαρακτήρας της XML σε συνδυασμό με το γεγονός ότι η γλώσσα αυτή είναι ανεξάρτητη από το λειτουργικό σύστημα ή τη γλώσσα προγραμματισμού που θα χρησιμοποιηθεί για να δημιουργήσει το κείμενο XML που θα περιέχει τα δεδομένα που πρέπει να ανταλλαχθούν, την έκανε ως το κατεξοχήν πρότυπο για τη σύνταξη και ανταλλαγή κειμένων στο διαδίκτυο. Ακολουθεί ένα παράδειγμα δόμησης της πληροφορίας με XML που αναφέρεται στα στοιχεία ενός φοιτητή ενός πανεπιστημιακού ιδρύματος:


```

< Student >
  <id>1438</id>
  <first_name>Dimitrios</first_name>
  <last_name>Tsesmetzis</last_name>
  <organization>National Technical University of Athens</organization>
  <email>jimmis@telecom.ece.ntua.gr</email>
  <phone>00302107721498</phone>
</Student>

```

Επισημαίνεται ότι η χρησιμότητα του στρώματος αυτού περιορίζεται στην ορθή δόμηση και σύνταξη των κειμένων, χωρίς να προσφέρει κανέναν τρόπο κατανόησης της πληροφορίας που περιέχουν. Για παράδειγμα δεν υπάρχει τρόπος όσον αφορά το προηγούμενο παράδειγμα να αντιλαμβάνεται το σύστημα ότι η πληροφορία που περιλαμβάνεται στην ετικέτα του οργανισμού (<organization>...</organization>), θα περιέχει το όνομα κάποιου ιδρύματος και όχι κάτι τελείως διαφορετικό, όπως για παράδειγμα το όνομα του φοιτητή. Αυτός είναι άλλωστε ο ρόλος των ανώτερων στρωμάτων, όπως θα φανεί και στη συνέχεια.

2.2.3. Στρώμα RDF - RDFS

Στους συντακτικούς - και όχι μόνο - κανόνες που προσφέρει η XML και τα πρότυπά της θα στηριχθεί η ανάπτυξη των γλωσσών RDF (Resource Description Framework) και RDFS (RDF Schema), που αποτελούν το επόμενο στρώμα του σημασιολογικού δικτύου. Το στρώμα αυτό αποτελεί τη βάση της κύριας συνεισφοράς του δικτύου αυτού, κάτι που γίνεται άμεσα κατανοητό, αν αναλογιστεί κανείς ότι όλες οι γλώσσες που αφορούν οντολογίες στηρίζονται στο στρώμα αυτό. Αν και καταχρηστικά θεωρείται γλώσσα, η RDF [52] είναι περισσότερο ένα μοντέλο δεδομένων, το οποίο χρησιμοποιήθηκε κυρίως για να παρέχει μεθόδους προτυποποίησης της πληροφορίας που αφορά τα μεταδεδομένα (metadata) των πόρων (resources) του διαδικτύου καθώς και τις σχέσεις μεταξύ των πόρων αυτών. Απώτερος στόχος του μοντέλου είναι η δυνατότητα διαχείρισης της πληροφορίας αυτής από τους υπολογιστές αυτόματα χωρίς την παρεμβολή εξωτερικών παραγόντων. Η RDF ορίζει ως πόρο στο διαδίκτυο κάθε αντικείμενο (object) το οποίο διαθέτει ένα μοναδικό αναγνωριστικό URI. Ο κάθε πόρος διαθέτει κάποιες ιδιότητες (properties), ενώ οι ιδιότητες αυτές έχουν ένα τύπο δεδομένων (π.χ. ακέραιος, δεκαδικός, λογικός, συμβολοακολουθία) και εκχωρούνται σε αυτές τιμές ανάλογα με τον τύπο των δεδομένων που ανήκουν, συσχετίζοντας με αυτόν τον τρόπο τον κάθε πόρο με τις τιμές που μπορεί να λάβει. Το μοντέλο RDF δηλαδή είναι ένα απλό μοντέλο δεδομένων που στηρίζεται στην τριάδα: υποκείμενο (subject), κατηγορούμενο (predicate), αντικείμενο (object) που καλείται αναφορά (statement). Επιπρόσθετα, η RDF όντας μοντέλο δεδομένων δε διαθέτει δικούς της συντακτικούς κανόνες. Επομένως, πρέπει να

στηριχθεί σε κάποιο άλλο μοντέλο, που αυτό δεν είναι άλλο βέβαια από αυτό της XML. Αξίζει να επισημανθεί ωστόσο, ότι η χρήση της XML δεν είναι προαπαιτούμενο για την RDF αλλά για το σημασιολογικό δίκτυο, προκειμένου να εκμεταλλευθεί και επωφεληθεί των πολλών πλεονεκτημάτων που αυτή διαθέτει. Ακολουθεί ένα παράδειγμα προκειμένου να καταστεί σαφής η μορφή μιας τέτοιας αναφοράς (statement) του μοντέλου δεδομένων RDF:

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/">
  <rdf:Description rdf:about="http://www.mypage.com/Life_is_Wonderful">
    <dc:title>Life is Wonderful</dc:title>
    <dc:creator>Dimitrios Tsesmetzis</dc:creator>
    <dc:date>2008-01-28</dc:date>
  </rdf:Description>
</rdf:RDF>
```

Επικεντρώνοντας στα σημεία που αφορούν την τριάδα υποκείμενο – κατηγορούμενο – αντικείμενο, έχουμε:

Υποκείμενο:

- http://www.mypage.com/Life_is_Wonderful, το οποίο αναγνωρίζεται μοναδικά από τη URL του

Κατηγορούμενο (3) - ιδιότητες:

- <http://purl.org/dc/elements/1.1/title>
- <http://purl.org/dc/elements/1.1/creator>
- <http://purl.org/dc/elements/1.1/date>

Αντικείμενο (3) - τιμές:

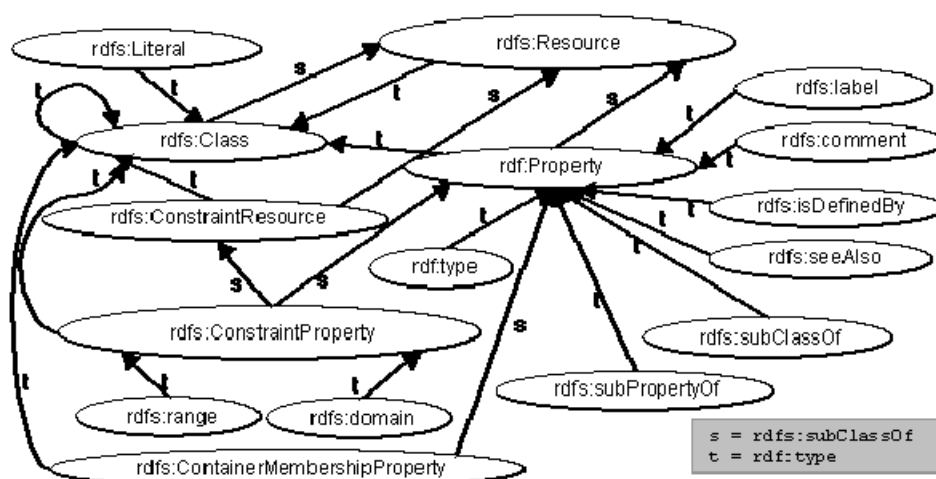
- *Life is Wonderful*
- *Dimitrios Tsesmetzis*
- *2008-01-28*

Όσον αφορά τη γλώσσα RDFS [52], αυτή μπορεί να θεωρηθεί ως μια στοιχειώδης γλώσσα για οντολογίες [1]. Στηρίζεται στη γλώσσα RDF, δίνοντας τη δυνατότητα στο χρήστη να δημιουργήσει για την περιγραφή του κάθε πόρου (resource) που επιθυμεί το δικό του λεξιλόγιο RDF [53]. Αποτέλεσμα αυτού είναι η δημιουργία αντικειμένων τα οποία επιτρέπουν τη δυνατότητα εξαγωγής πολύτιμων συμπερασμάτων υπό το πρίσμα της σημασιολογίας των δεδομένων που περιέχουν. Η «κατανόηση» αυτών των δεδομένων είναι και ο απώτερος στόχος του σημασιολογικού δικτύου όπως έχει ήδη αναφερθεί, καθιστώντας εφικτή την επεξεργασία τους αυτόματα από τα υπολογιστικά συστήματα. Αξίζει να επισημανθεί, ότι τα αντικείμενα που δημιουργούνται από την RDFS διαθέτουν πολλές ομοιότητες με τα αντικείμενα που δημιουργούνται από μια αντικειμενοστραφής γλώσσα όπως είναι η Java ή η C++. Επομένως, πολλές από τις αρχές του αντικειμενοστραφούς

προγραμματισμού (Object-Oriented programming) [54] συναντώνται και σε αυτά τα αντικείμενα.

Πιο αναλυτικά, η γλώσσα RDFS στηρίζεται σε κλάσεις (classes) που ουσιαστικά είναι δομές παρόμοιων χαρακτηριστικών. Όπως και στον αντικειμενοστραφή προγραμματισμό, η κάθε κλάση στην RDFS έχει χαρακτηριστικά που λέγονται ιδιότητες (properties), μπορεί να κληρονομήσει τα χαρακτηριστικά της δημιουργώντας υποκλάσεις (subclasses) και κατά συνέπεια μια δομή ιεραρχίας μεταξύ των κλάσεων. Επιπρόσθετα, υπάρχουν σχέσεις (relationships) μεταξύ των οντοτήτων αυτών, δημιουργούνται αντικείμενα (objects - instances) των κλάσεων αυτών, ενώ υπάρχουν και περιορισμοί που σχετίζονται με το πεδίο ορισμού ή το πεδίο τιμών των ιδιοτήτων αυτών, που ουσιαστικά δηλώνουν ποιες κλάσεις μπορεί να δεχτεί σαν όρισμα ή σαν πεδίο τιμών η κάθε μια από αυτές τις ιδιότητες.

Στο σχήμα [4] που ακολουθεί, σκιαγραφείται η δομή ιεράρχησης των κλάσεων της γλώσσας RDFS καθώς και οι μεταξύ τους σχέσεις:



Σχήμα 2-6 Ιεραρχία και σχέσεις κλάσεων της γλώσσας RDFS

Η ανάλυση όλων των στοιχείων και σχέσεων που περιλαμβάνονται στο παραπάνω σχήμα ξεφεύγει από τα πλαίσια της παρούσας διατριβής, ωστόσο ο ενδιαφερόμενος αναγνώστης μπορεί να ανατρέξει στο [4] για περισσότερες πληροφορίες. Αξίζει να επισημανθεί βέβαια ότι όπως έχει ήδη αναφερθεί, κυρίαρχο ρόλο παίζουν οι κλάσεις *rdfs:Resource*, *rdfs:Class* και *rdf:Property*, ενώ οι άλλες κλάσεις της γλώσσας είναι οι *rdfs:ConstraintProperty* και *rdfs:ContainerMembershipProperty*. Όσον αφορά τις ιδιότητες, οι σημαντικότερες από αυτές είναι:

- *rdf:type*, η οποία δηλώνει ότι ένας πόρος (resource) είναι μέλος μιας κλάσης, κληρονομώντας όπως είναι εύλογο και όλα τα χαρακτηριστικά της κλάσης αυτής.

- *rdfs:subClassOf*, η οποία χρησιμοποιείται για να δηλώσει ότι μια κλάση είναι υποκλάση κάποιας άλλης, με τη δυνατότητα πολλαπλής κληρονομικότητας, υπό την έννοια ότι αν μια κλάση είναι υποκλάση μιας άλλης η οποία είναι υποκλάση μιας τρίτης κλάσης, τότε και η πρώτη είναι υποκλάση της τρίτης κλάσης.
- *rdfs:subPropertyOf*, η οποία χρησιμοποιείται για να δηλώσει ότι μια ιδιότητα είναι υποιδιότητα μιας άλλης ιδιότητας κατά τον ίδιο τρόπο και ακολουθώντας τους ίδιους μηχανισμούς κληρονομικότητας με την περίπτωση που μια κλάση είναι υποκλάση μιας άλλης κλάσης. Αξίζει να αναφερθεί ότι η ιδιότητα αυτή δε συναντάται στον αντικειμενοστραφή προγραμματισμό που περιορίζεται σε κληρονομικότητες μόνο μεταξύ κλάσεων και όχι μεταξύ ιδιοτήτων.
- *rdfs:seeAlso*, η οποία χρησιμοποιείται για να δηλώσει ότι ένας πόρος δύναται να περιέχει συμπληρωματικές πληροφορίες για τον τρέχον πόρο.
- *rdfs:isDefinedBy*, η οποία είναι υποιδιότητα της ιδιότητας *rdfs:seeAlso* και δηλώνει τον πόρο που περιέχει τον ορισμό του τρέχοντος πόρου.

2.2.4. Στρώμα Οντολογιών

Αν και η συνεισφορά των RDF/RDFS στην αυτοματοποιημένη επεξεργασία των δεδομένων από υπολογιστές ήταν καθοριστική, αποτελώντας τις πρώτες στοιχειώδεις γλώσσες που έδωσαν τη δυνατότητα «κατανόησης» των δεδομένων που ανταλλάσσουν και διαχειρίζονται οι υπολογιστές, ωστόσο είχαν αρκετούς περιορισμούς. Οι σημαντικότεροι από αυτούς είναι [48]:

- Δεν υποστηρίζονται σχέσεις ισοδυναμίας μεταξύ των ιδιοτήτων (properties), κάτι που θα επέτρεπε σε ομάδες ερευνητών που δουλεύουν ανεξάρτητα και στο ίδιο αντικείμενο να συνδυάσουν και να ενοποιήσουν τα αποτελέσματα των προσπαθειών τους.
- Δε δίνεται η δυνατότητα να προσδιοριστεί ότι μια ιδιότητα είναι μοναδική (unique) ή να καθοριστεί το πλήθος των ιδιοτήτων (cardinality) που μπορούν να συσχετιστούν με ένα συγκεκριμένο αντικείμενο μιας κλάσης.
- Δε δίνεται η δυνατότητα να προκαθοριστούν οι επιτρεπόμενες τιμές που μπορεί να πάρει μια ιδιότητα υπό μορφή λίστας, χωρίς αυτό βέβαια να σημαίνει ότι δε δίνεται η δυνατότητα να εκχωρηθεί σε μια συγκεκριμένη ιδιότητα μια συγκεκριμένη τιμή.
- Δεν υποστηρίζεται η δυνατότητα απόρριψης μιας τιμής για μια ιδιότητα (disjointness). Δηλαδή ενώ υποστηρίζεται η δήλωση ότι ο Αντρέας *είναι*

παντρεμένος, δεν προσφέρεται η δυνατότητα της δήλωσης ότι ο Αντρέας δεν είναι παντρεμένος.

- Δεν υποστηρίζονται ενώσεις (unions) και διασταυρώσεις (intersections) κλάσεων με αποτέλεσμα να περιορίζεται σημαντικά η δυνατότητα σύνθεσης πολύπλοκων κλάσεων.

Επομένως ήταν απαραίτητο να δημιουργηθούν γλώσσες που θα μπορούσαν να εκφράσουν πιο πολύπλοκες σχέσεις μεταξύ των δικτυακών πόρων (resources) και οι οποίες δε θα δεσμεύονταν από περιορισμούς όπως αυτοί που μόλις αναφέρθηκαν. Οι γλώσσες αυτές όπως είναι λογικό και αναμενόμενο θα στηριχθούν και θα εμπλουτίσουν τις ήδη υπάρχουσες γλώσσες RDF / RDFS. Απώτερος στόχος και σκοπός τους βέβαια δεν είναι άλλος από το να επιτρέψουν την αυτοματοποιημένη επεξεργασία των δεδομένων κάθε ερευνητικού πεδίου από υπολογιστές. Προκειμένου να επιτευχθεί αυτό, θα πρέπει οι γλώσσες αυτές να δημιουργήσουν οντολογίες. Οι οντολογίες, αν και φιλοσοφική έννοια με ρίζες στην αρχαία Ελλάδα όπως αναφέρθηκε και στο προηγούμενο κεφάλαιο, αποτελούν τη λύση στην πλήρη δόμηση της πληροφορίας, επιτρέποντας τη δημιουργία κάθε μορφής σχεδόν πολύπλοκων σχέσεων μεταξύ των διαφόρων πόρων. Σύμφωνα με έναν από τους πιο σαφείς ορισμούς τους [5], «η οντολογία ορίζει τους όρους που χρησιμοποιούνται για να περιγράψουν και να αναπαραστήσουν μια γνωστική περιοχή».

Από θεωρητική σκοπιά, τα είδη των οντολογιών χωρίζονται στις ακόλουθες κατηγορίες [55]:

- Οντολογία Ανώτερου Επιπέδου, η οποία ορίζει τις έννοιες στις οποίες θα στηριχθούν για τη δημιουργία τους οι άλλες οντολογίες.
- Οντολογία Περιοχής, η οποία είναι η πιο συχνή μορφή οντολογιών και η οποία περιέχει την ορολογία και τις έννοιες που αφορούν μια συγκεκριμένη περιοχή ενδιαφέροντος.
- Οντολογία Διεργασίας, η οποία περιέχει τις εισόδους, εξόδους, περιορισμούς, σχέσεις, όρους και πληροφορίες που σχετίζονται με μια συγκεκριμένη ή μια ομάδα από επιχειρησιακές διεργασίες.
- Οντολογία Διεπαφής, η οποία ορίζει τους περιορισμούς στη δομή και στο περιεχόμενο, όσον αφορά μια συγκεκριμένη διεπαφή όπως είναι για παράδειγμα η διεπαφή κάποιου προγράμματος (Application Programming Interface).
- Οντολογία βασισμένη σε ρόλους, η οποία περιέχει την ορολογία και τις έννοιες σχετικές με ένα συγκεκριμένο τελικό χρήστη, ο οποίος μπορεί να είναι είτε φυσική οντότητα είτε κάποια εφαρμογή.

Αναφερόμενοι στα δομικά χαρακτηριστικά των οντολογιών, μια οντολογία αποτελείται από [56]:

- Κλάσεις (classes) που λέγονται και έννοιες (concepts) στην ορολογία των οντολογιών.
- Ιδιότητες (properties) της κάθε έννοιας (concept) που περιγράφουν τα χαρακτηριστικά της έννοιας, που στην ορολογία των οντολογιών λέγονται και slots ή roles.
- Περιορισμούς στα slots (role restrictions).

Όταν η οντολογία συνοδεύεται από ομάδα αντικειμένων της (instances), τότε θεωρείται ότι σχηματίζεται μια βάση γνώσης (knowledge base).

Παρατηρώντας και συγκρίνοντας τα χαρακτηριστικά των οντολογιών με τα αντίστοιχα χαρακτηριστικά της γλώσσας RDFS που αναλύθηκαν στην προηγούμενη παράγραφο, διαπιστώνονται εύκολα οι ομοιότητες που υπάρχουν. Επομένως, η ενσωμάτωσή αυτών των οντολογικών χαρακτηριστικών σε μια νέα γλώσσα που θα στηρίζεται στα χαρακτηριστικά της RDFS είναι εύκολο να επιτευχθεί. Ωστόσο, πρέπει να δοθεί προσοχή στον εμπλουτισμό των δυνατοτήτων της νέας γλώσσας, ώστε να υποστηρίζονται πιο πολύπλοκες σχέσεις μεταξύ των κλάσεων και ιδιοτήτων, αποφεύγοντας τους περιορισμούς που συναντήθηκαν στην RDFS. Η πιο δημοφιλής γλώσσα για οντολογίες, που προσφέρει όλες αυτές τις δυνατότητες είναι η γλώσσα OWL (Ontology Web Language) [5].

Η γλώσσα OWL περιλαμβάνει 3 υπογλώσσες:

- Την OWL Lite, η οποία, αν και προσφέρει αρκετές δυνατότητες, επιτρέποντας τη δημιουργία ταξινομήσεων και περιορισμών, ωστόσο δεν υποστηρίζει το σύνολο των δυνατοτήτων μιας γλώσσας οντολογίας. Για παράδειγμα, ενώ προσφέρει τη δυνατότητα καθορισμού του πλήθους των επιτρεπόμενων τιμών μιας ιδιότητας (cardinality), το πεδίο τιμών του πλήθους αυτού είναι μόνο 0 και 1.
- Την OWL DL (Description Logic) η οποία προσφέρει τη δυνατότητα δημιουργίας πολύπλοκων σχέσεων και εκφράσεων. Επιπρόσθετα, τα προγράμματα (Reasoners) που θα χρησιμοποιηθούν για να εφαρμόσουν τους κανόνες λογικής (περισσότερες πληροφορίες για αυτά τα προγράμματα στο επόμενο στρώμα), ώστε οι οντολογίες που έχουν στηριχθεί σε αυτήν να επεξεργάζονται αυτόματα από τους υπολογιστές, βεβαιώνεται ότι θα τερματίσουν σε πεπερασμένο χρόνο. Αυτοί είναι και οι κύριοι λόγοι που την κάνουν την πιο δημοφιλή από τις 3 γλώσσες.

- Την OWL Full, η οποία, αν και προσφέρει τις πιο πολύπλοκες σχέσεις και συνθέσεις μεταξύ κλάσεων και ιδιοτήτων, δίνοντας τη δυνατότητα να δημιουργηθούν περιορισμοί κάθε είδους, ωστόσο θεωρείται αδύνατο να μπορέσει να δημιουργηθεί ποτέ κάποιο πρόγραμμα (Reasoner) που θα μπορέσει να εφαρμόσει κανόνες λογικής σε οντολογίες που έχουν στηριχθεί σε αυτή τη γλώσσα. Κάτι που έρχεται σε αντίθεση με το βασικό πρόβλημα που έρχεται να αντιμετωπίσει το σημασιολογικό δίκτυο, που είναι η πλήρης αυτοματοποίηση της επεξεργασίας της πληροφορίας από τα υπολογιστικά συστήματα. Αυτή η αδυναμία την καθιστά αυτόματα γλώσσα περιορισμένου ενδιαφέροντος.

Όπως και στην περίπτωση της RDFS, η ανάλυση όλων των δυνατοτήτων και των επιμέρους χαρακτηριστικών των γλωσσών αυτών ξεφεύγει από τα πλαίσια της παρούσας διατριβής, ωστόσο ο ενδιαφερόμενος αναγνώστης μπορεί να ανατρέξει στην εργασία [5] για περισσότερες πληροφορίες. Ωστόσο, ορισμένα από τα σημαντικότερα χαρακτηριστικά όσον αφορά τις ιδιότητές τους (properties) είναι :

Καταρχήν, υπάρχουν δύο ειδών ιδιότητες [50]:

- Οι ιδιότητες αντικειμένων (object properties) που συσχετίζουν οντότητα με οντότητα. Η οντότητα είναι ένα αντικείμενο (instance) μιας κλάσης.
- Οι ιδιότητες τύπου δεδομένων (datatype properties) που συσχετίζουν οντότητα με ένα τύπο δεδομένων, όπως για παράδειγμα ακέραιο, δεκαδικό, λογικό, συμβολοακολουθία κ.α.. Επισημαίνεται, ότι η OWL χρησιμοποιεί για τις ιδιότητες αυτού του τύπου τους τύπους δεδομένων που υποστηρίζονται από την XSD.

Οι ιδιότητες αυτές χαρακτηρίζονται ως:

- Λειτουργικές (functional), η οποία δηλώνει ότι ένα αντικείμενο μπορεί να έχει μία μόνο τιμή για αυτή την ιδιότητα. Ενδεικτικά παραδείγματα είναι η ηλικία, το ύψος ή το βάρος κάποιου ανθρώπου.
- Αντίστροφα λειτουργικές (Inverse functional), η οποία δηλώνει ότι δύο διαφορετικές οντότητες δε μπορούν να έχουν ίδια τιμή. Ενδεικτικό παράδειγμα είναι ο αριθμός τραπεζικού λογαριασμού κάποιου φυσικού ή νομικού προσώπου.
- Συμμετρικές (Symmetric), όταν συνδέουν μια οντότητα (A) με μια άλλη (B) και συμπεραίνεται ότι η ίδια σχέση ισχύει και μεταξύ B και A. Ενδεικτικό παράδειγμα η περίπτωση που κάποιος είναι αδελφός με κάποιον άλλον.
- Μεταβατικές (Transitive), όταν μια οντότητα (A) συνδέεται με την ιδιότητα αυτή με μια οντότητα (B) και η οντότητα (B) με την ίδια ιδιότητα με μια άλλη

οντότητα (Γ), τότε και η οντότητα (Α) συμπεραίνεται ότι συνδέεται με την ίδια ιδιότητα με την οντότητα (Γ). Ενδεικτικό παράδειγμα είναι η περίπτωση που κάποιος (Α) είναι πιο ψηλός από κάποιον άλλον (Β). Αν ο (Β) είναι ψηλότερος από κάποιον τρίτο (Γ), τότε μπορεί να υποτεθεί με ασφάλεια ότι και ο (Α) είναι ψηλότερος από τον (Γ).

2.2.5. Στρώματα Λογικής (Logic), Απόδειξης (Proof), Εμπιστοσύνης (Trust)

Η δημιουργία των πιο ολοκληρωμένων οντολογιών που θα περιγράφουν ακόμα και την πιο εξεζητημένη γνωστική περιοχή θα ήταν ωστόσο πρακτικά άχρηστη αν δε συνοδευόταν από τη δημιουργία αντίστοιχων κανόνων, που θα έδιναν τη δυνατότητα στους υπολογιστές να «σκεφτούν», να εξάγουν συμπεράσματα και να λάβουν τις κατάλληλες αποφάσεις για τα δεδομένα που επεξεργάζονται χωρίς την παρεμβολή του ανθρώπινου παράγοντα [1]. Αυτός είναι ο ρόλος των στρωμάτων Λογικής (Trust) και Απόδειξης (Proof), που στοχεύουν στη δημιουργία κατάλληλων κανόνων, εμπλουτίζοντας κατ' αυτόν τον τρόπο τη λειτουργικότητα των οντολογιών. Ορισμένα από τα πιο δημοφιλή προγράμματα (Reasoners) που δίνουν αυτή τη δυνατότητα στους υπολογιστές είναι τα: Pellet [57], KAON2 [58], RacerPro [59], και Fact++ [60].

Ο ρόλος του ανώτερου στρώματος του σημασιολογικού δικτύου (Εμπιστοσύνης (Trust)) έγκειται στην παροχή μηχανισμών πιστοποίησης και εγκυρότητας των δεδομένων που ανταλλάσσονται. Λόγω της σπουδαιότητάς του είναι εύλογο ότι θα παίζει καθοριστικό ρόλο στην αποδοχή και εξάπλωση του σημασιολογικού δικτύου. Επιπρόσθετα η παροχή υπηρεσιών με ασφάλεια είναι ένα από τα πιο πρωταρχικά ζητήματα που πρέπει να αντιμετωπιστούν συνολικά και ολοκληρωτικά, αφού σε διαφορετική περίπτωση θεωρείται βέβαιο το ενδεχόμενο οι υποψήφιοι πελάτες να μην επιθυμούν να χρησιμοποιήσουν τις προσφερόμενες υπηρεσίες.

Επισημαίνεται ότι τα τρία ανώτερα στρώματα του σημασιολογικού δικτύου βρίσκονται ακόμα σε πρωτογενή στάδια σχεδιασμού και απαιτείται επίπονη και συλλογική προσπάθεια προτού αναπτυχθούν και μορφοποιηθούν κατάλληλα, ώστε να επιτρέψουν το όραμα του σημασιολογικού δικτύου να γίνει πραγματικότητα.

2.3. Οντολογίες και Διαδικτυακές Υπηρεσίες

Η παρουσίαση των χαρακτηριστικών των οντολογιών και της γλώσσας OWL που έγινε σε προηγούμενο παράγραφο, έκανε σαφές ότι η εκμετάλλευση των δυνατοτήτων τους θα συνέβαλε σημαντικά στην αυτοματοποίηση των διαδικασιών που σχετίζονται με το διαδίκτυο. Αντικείμενο της παραγράφου αυτής αποτελεί η εφαρμογή τους στις διαδικτυακές

υπηρεσίες, καταδεικνύοντας τα οφέλη που θα έχει η ενέργεια αυτή. Επιπρόσθετα, θα παρουσιαστεί η πιο δημοφιλής γλώσσα οντολογιών για διαδικτυακές υπηρεσίες (OWL – S) (Ontology Web Language for Web Services) και τα σημαντικότερα χαρακτηριστικά αυτής.

2.3.1. Σημασιολογικές Διαδικτυακές Υπηρεσίες

Η περιγραφή των διαδικτυακών υπηρεσιών με τις υπάρχουσες τεχνολογίες, όπως αυτές παρουσιάστηκαν στην παράγραφο 2.1, έχει ως αποτέλεσμα την περιγραφή των υπηρεσιών αυτών μόνο σε συντακτικό επίπεδο και όχι σε σημασιολογικό. Κάτι τέτοιο ωστόσο περιορίζει σημαντικά τις παροχές που μπορούν να προσφερθούν από τις υπηρεσίες αυτές, αφού δε δίνεται η δυνατότητα να αυτοματοποιηθούν οι διαδικασίες που σχετίζονται με την εύρεση, τη σύνθεση και την εκτέλεση των υπηρεσιών αυτών. Επιπρόσθετα, όπως έγινε σαφές και στην προηγούμενη παράγραφο, ο σχεδιασμός κατάλληλων οντολογιών για διαδικτυακούς πόρους προσφέρει τη δυνατότητα στα υπολογιστικά συστήματα να «κατανοήσουν» την πληροφορία που επεξεργάζονται και να λαμβάνουν κατάλληλες αποφάσεις χωρίς τη συμμετοχή του ανθρώπινου παράγοντα. Επομένως, λύση στο πρόβλημα της αυτοματοποίησης των διαδικασιών που αναφέρθηκαν παραπάνω είναι η σημασιολογική περιγραφή των υπηρεσιών αυτών, η οποία οφείλει να στηρίζεται σε κατάλληλες οντολογίες. Για να είναι εφικτή η περιγραφή αυτή, θα πρέπει να σχεδιαστεί κατάλληλη γλώσσα, που να υπακούει τους κανόνες και τις αρχές των οντολογιών, όπως αυτές αναλύθηκαν στην παράγραφο 2.2.4. Η γλώσσα αυτή, η οποία αποτελεί επέκταση της OWL και προσφέρει τη δυνατότητα περιγραφής σημασιολογικών χαρακτηριστικών διαδικτυακών υπηρεσιών είναι η OWL-S [14]. Ωστόσο, προτού αναλυθούν τα τεχνικά χαρακτηριστικά της γλώσσας OWL-S, κρίνεται σκόπιμο να παρουσιαστούν τα οφέλη που θα μπορούν να αποκομίσουν οι πάροχοι και οι πελάτες των διαδικτυακών υπηρεσιών από την υιοθέτηση της γλώσσας αυτής ως γλώσσας περιγραφής διαδικτυακών υπηρεσιών.

Αυτόματη εύρεση διαδικτυακών υπηρεσιών

Η διαδικασία αυτή στοχεύει στον εντοπισμό της κατάλληλης υπηρεσίας, η οποία ικανοποιεί όλες τις απαιτήσεις του χρήστη. Οι απαιτήσεις αυτές δύναται να διαθέτουν μεγάλο βαθμό πολυπλοκότητας καθώς και σύνθετους περιορισμούς, αφού στηρίζουν την περιγραφή τους σε οντολογίες. Ενδεικτικό παράδειγμα είναι η υπηρεσία αγοράς αεροπορικών εισιτηρίων μεταξύ δύο συγκεκριμένων πόλεων κάνοντας χρήση συγκεκριμένης πιστωτικής κάρτας. Στην παρούσα φάση ο χρήστης θα πρέπει μόνος του να εντοπίσει τον κατάλληλο δικτυακό τόπο που προσφέρει την υπηρεσία αγοράς εισιτηρίων, κατόπιν να εισάγει στα κατάλληλα πεδία τις ιδιαίτερες προτιμήσεις του και να αποφασίσει από τα αποτελέσματα που θα του επιστραφούν αν είναι ικανοποιημένος ή θα επιχειρήσει να εντοπίσει κάποια άλλη αντίστοιχη υπηρεσία που θα τον ικανοποιεί περισσότερο. Με την εισαγωγή και υιοθέτηση της OWL-S ως γλώσσα περιγραφής διαδικτυακών υπηρεσιών ωστόσο το πρόβλημα αυτό αντιμετωπίζεται

ολοκληρωτικά ως εξής: Δίνεται η δυνατότητα στους παρόχους των υπηρεσιών να περιγράφουν τις υπηρεσίες τους δίνοντας πλήθος χαρακτηριστικά αυτών και να τα δημοσιοποιούν σε κατάλληλες μονάδες εγγραφών (registries), ενώ κατάλληλα προγράμματα (agents) αναλαμβάνουν από τη μεριά του χρήστη τον εντοπισμό των υπηρεσιών αυτών που ικανοποιούν όλες τις απαιτήσεις τους. Οι μέθοδοι στις οποίες στηρίζονται τα προγράμματα αυτά, ώστε να «συμπεράνουν» ότι μια υπηρεσία ικανοποιεί ή όχι τις απαιτήσεις ενός χρήστη, παρουσιάζονται αναλυτικά σε επόμενη παράγραφο (§2.3.3).

Αυτόματη εκτέλεση διαδικτυακών υπηρεσιών

Η διαδικασία αυτή, η οποία προϋποθέτει ότι ο εντοπισμός της επιθυμητής υπηρεσίας έχει ήδη επιτευχθεί, στοχεύει στην εκτέλεση της διαδικτυακής υπηρεσίας αυτόματα χωρίς την παρεμβολή του ανθρώπινου στοιχείου. Κάνοντας χρήση των υπάρχουσών τεχνολογιών, η εκτέλεση διαδικτυακής υπηρεσίας επιτυγχάνεται στηριζόμενη στο σαφή καθορισμό των δεδομένων εισόδου της από τη μεριά του χρήστη στο πρόγραμμα που αναλαμβάνει την εκτέλεσή της. Ακόμη, η διαχείριση των μηνυμάτων απόκρισης της ενέργειας αυτής, όπως για παράδειγμα σε περίπτωση αποτυχίας εκτέλεσης της υπηρεσίας, γίνεται επίσης αποκλειστικά από το χρήστη. Στηριζόμενοι ωστόσο στις δυνατότητες που προσφέρει η OWL-S, είναι εφικτό η διαδικασία αυτή να αυτοματοποιηθεί και κατάλληλα προγράμματα (agents) να αναλάβουν την εκτέλεση της υπηρεσίας, διατηρώντας τον έλεγχο σε κάθε στάδιο και λαμβάνοντας τις κατάλληλες αποφάσεις σε κάθε περίπτωση, οι οποίες βέβαια θα ικανοποιούν πάντα τις απαιτήσεις του χρήστη. Αυτό είναι απόρροια του γεγονότος ότι οι διεπαφές που διατίθενται από την OWL-S δίνουν τη δυνατότητα αφενός να περιγράψει ο χρήστης με σημασιολογικό τρόπο τις απαιτήσεις του και αφετέρου να μπορούν να κατανοήσουν τα προγράμματα (agents) τη σημασιολογία των απαιτήσεων αυτών, ενεργώντας κατάλληλα κατά την εκτέλεση της επιθυμητής υπηρεσίας.

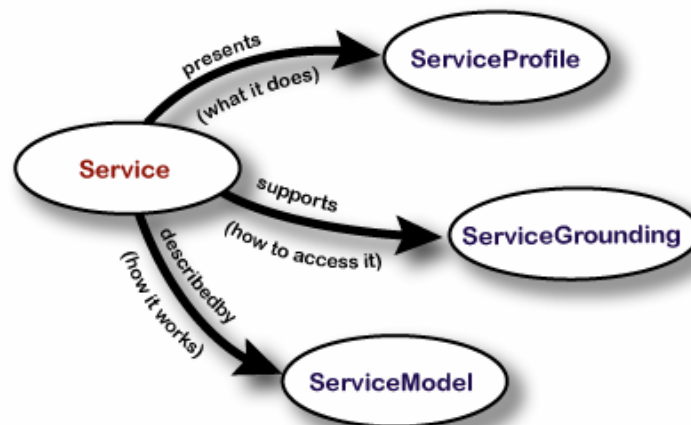
Αυτόματη σύνθεση διαδικτυακών υπηρεσιών

Η ενέργεια αυτή στοχεύει στην αυτοματοποίηση της διαδικασίας επιλογής κατάλληλων υπηρεσιών, προκειμένου να δημιουργηθεί μια σύνθετη υπηρεσία, η οποία λαμβάνει χώρα σε περισσότερα από ένα στάδια (υπηρεσίες) ελέγχοντας παράλληλα την επικοινωνία μεταξύ των υπηρεσιών αυτών. Χαρακτηριστικό παράδειγμα είναι η περίπτωση όπου ένας χρήστης επιθυμεί να ταξιδέψει σε μια χώρα κάνοντας όλες τις απαραίτητες κρατήσεις όσον αφορά αεροπορικά εισιτήρια, ξενοδοχείο που θα διαμένει ή ακόμα και αυτοκίνητο που επιθυμεί να μισθώσει κατά την παραμονή του στη χώρα αυτή. Η διαδικασία που ακολουθείται στην τρέχουσα περίοδο είναι η εκτέλεση όλων των επιμέρους ενεργειών από τη μεριά του χρήστη. Η διαδικασία αυτή όμως είναι πολύ χρονοβόρα και επίπονη όπως είναι κατανοητό, ενώ δεν είναι λίγες οι φορές που ο χρήστης συμβιβάζεται σε λύσεις που δεν τον ικανοποιούν απόλυτα λόγω έλλειψης χρόνου ή τεχνικών γνώσεων. Με την υιοθέτηση της γλώσσας OWL-S ωστόσο, δίνεται η δυνατότητα στο χρήστη να καθορίσει πολύπλοκες απαιτήσεις, ενώ

προγράμματα (agents) αναλαμβάνουν να «κατανοήσουν» όλες τις απαιτήσεις αυτές, να επιλέξουν τις κατάλληλες υπηρεσίες ή σύνθεση των οποίων τις ικανοποιεί και να αναλάβουν την επικοινωνία με τις υπηρεσίες αυτές. Αυτό επιτυγχάνεται κάνοντας χρήση κάποιων χαρακτηριστικών της OWL-S τα οποία αναλύονται στη συνέχεια, όπως είναι τα προαπαιτούμενα (prerequisites) και τα αποτελέσματα (consequences) καθώς και ενός μοντέλου αλληλεπιδράσεων ροής δεδομένων (data flow).

2.3.2. Χαρακτηριστικά OWL-S

Η γλώσσα OWL-S, όντας μια γλώσσα για οντολογίες, κληρονομεί όλα τα χαρακτηριστικά που διαθέτει μια τέτοια γλώσσα. Επομένως, διαθέτει κλάσεις (classes / concepts), ιδιότητες (properties / slots / roles), καθώς και περιορισμούς (restrictions) στις ιδιότητες αυτές. Δομική μονάδα στη γλώσσα αυτή είναι όπως είναι αναμενόμενο η υπηρεσία (Service). Κάθε διαδικτυακή υπηρεσία διαθέτει απαραίτητα ένα αντικείμενο της κλάσης αυτής (Service), ενώ όπως σκιαγραφείται και στο Σχήμα 2-7 [14] η κλάση αυτή διαθέτει 3 κλάσεις και 3 ιδιότητες που τη συνδέουν με αυτές.



Σχήμα 2-7 Μοντέλο διαδικτυακής υπηρεσίας με OWL - S

Η κλάση *Service* συνδέεται με την κλάση *ServiceProfile* μέσω της ιδιότητας (slot) *presents*. Στην *ServiceProfile* περιλαμβάνεται όλη η πληροφορία που αφορά τη λειτουργικότητα της υπηρεσίας. Χρησιμοποιείται είτε κατά τη διαδικασία εύρεσης κατάλληλης υπηρεσίας από τα προγράμματα (agents) / πελάτες όπως αναλύθηκε παραπάνω, είτε κατά τη διαδικασία που ένας πάροχος επιθυμεί να διαφημίσει τα χαρακτηριστικά της υπηρεσίας που προσφέρει. Πιο αναλυτικά, η πληροφορία που περιλαμβάνει η κλάση αυτή είναι:

- Τα στοιχεία του παρόχου της υπηρεσίας.
- Τις παραμέτρους εισόδου, εξόδου, τα προαπαιτούμενα και τα αποτελέσματα της εκτέλεσης της υπηρεσίας (Input, Output, Precondition, Effects) (IOPE). Οι παράμετροι εισόδου είναι οι παράμετροι που απαιτούνται για να εκτελεστεί η υπηρεσία. Οι παράμετροι εξόδου το αποτέλεσμα που επιστρέφει η εκτέλεση της

υπηρεσίας. Τα προαπαιτούμενα χρησιμοποιούνται προκειμένου να καθοριστεί τι είναι απαραίτητο να ισχύει για να μπορεί κάποιος να χρησιμοποιήσει την υπηρεσία αυτή, ενώ τα αποτελέσματα εκτέλεσης της υπηρεσίας αναφέρονται στις μεταβολές που λαμβάνουν χώρα μετά την εκτέλεση της υπηρεσίας. Ένα παράδειγμα για να γίνουν αυτά κατανοητά είναι η υπηρεσία αγοράς ενός αντικειμένου μέσω διαδικτύου. Σαν είσοδος της υπηρεσίας μπορεί να θεωρηθεί το συγκεκριμένο αντικείμενο καθώς και προσωπικά στοιχεία του πελάτη μεταξύ των οποίων είναι και ο αριθμός της πιστωτικής του κάρτας. Σαν έξοδος της υπηρεσίας η αγορά του αντικειμένου από τον πελάτη. Σαν προαπαιτούμενο μπορεί να θεωρηθεί ότι ο πελάτης οφείλει να διαθέτει έγκυρη πιστωτική κάρτα, ενώ σαν αποτέλεσμα της εκτέλεσης της υπηρεσίας ότι η πιστωτική κάρτα του πελάτη χρεώθηκε με το πόσο που κοστίζει το αντικείμενο που αγόρασε.

- Πρόσθετα χαρακτηριστικά της υπηρεσίας που δε σχετίζονται με τη λειτουργικότητα της υπηρεσίας, όπως είναι οι παράμετροι ποιότητας υπηρεσίας (διαθεσιμότητα (availability), προσπελασιμότητα (accessibility) κτλ.). Συχνά οι παράμετροι αυτές χαρακτηρίζονται και ως μη λειτουργικές παράμετροι για να διαχωρίζονται από τις παραμέτρους εισόδου / εξόδου στα IOPEs που ουσιαστικά περιγράφουν τη λειτουργικότητα της υπηρεσίας.

Η κλάση *Service* συνδέεται επίσης με την κλάση *ServiceModel* μέσω της ιδιότητας (slot) *describedby* όπως φαίνεται και στο Σχήμα 2-7. Στην *ServiceModel* περιλαμβάνεται όλη η απαραίτητη τεχνική πληροφορία για το πώς λειτουργεί η υπηρεσία. Επομένως, η κλάση αυτή χρησιμοποιείται κατά την εκτέλεση της υπηρεσίας για να ενημερώσει το πρόγραμμα (agent) – χρήστη για όλες τις απαραίτητες τεχνικές λεπτομέρειες, προκειμένου να είναι εφικτή η κλήση της. Πιο συγκεκριμένα η κλάση αυτή περιλαμβάνει:

- Την υλοποίηση των IOPEs. Επισημαίνεται ότι δύναται οι παράμετροι που περιλαμβάνονται στα IOPEs στην κλάση αυτή να είναι υπερσύνολο των αντίστοιχων παραμέτρων που διατίθενται στην *ServiceProfile*.
- Την κλάση *ProcessModel* ως υποκλάση της *ServiceModel*. Η κλάση αυτή μοντελοποιεί 3 διαφορετικές περιπτώσεις όσον αφορά το χαρακτηρισμό μιας υπηρεσίας, που στην ορολογία της κλάσης αυτής ονομάζεται διεργασία. Η υπηρεσία (διεργασία) χαρακτηρίζεται ως ατομική (atomic), όταν είναι άμεσα προσπελάσιμη, ενώ η διαδικασία εκτέλεσής της ολοκληρώνεται σε ένα βήμα, όπως για παράδειγμα, η υπηρεσία που επιστρέφει τον ταχυδρομικό κώδικα μιας περιοχής. Η διεργασία χαρακτηρίζεται ως απλή (simple) όταν περιλαμβάνει στοιχεία είτε για το πώς λειτουργεί μια ατομική υπηρεσία ή προσφέρει μια αφαιρετική αναπαράσταση μιας σύνθετης (composite) υπηρεσίας. Σύνθετη τέλος χαρακτηρίζεται η υπηρεσία που

περιλαμβάνει περισσότερα από ένα στάδια εκτέλεσης, ενώ οι δομικές μονάδες της περιλαμβάνουν τις ατομικές υπηρεσίες καθώς και διάφορες συνθήκες ελέγχου της μορφής If – Then – Else, Repeat – While κτλ..

Τέλος, η κλάση *Service* συνδέεται με την κλάση *ServiceGrounding* μέσω της ιδιότητας (slot) *supports*. Στην *ServiceGrounding* περιλαμβάνεται όλη η απαραίτητη πληροφορία, προκειμένου να είναι εφικτή η επικοινωνία των διαφόρων προγραμμάτων (agents) – πελατών με την υπηρεσία. Πιο αναλυτικά, η κλάση αυτή περιλαμβάνει:

- Το πρωτόκολλο επικοινωνίας που απαιτείται μεταξύ παρόχου και πελάτη για την εκτέλεση της υπηρεσίας.
- Τη μορφοποίηση του μηνύματος που θα ανταλλαχθεί μεταξύ των δύο αυτών οντοτήτων.
- Τη διεύθυνση (URI) της υπηρεσίας.
- Άλλες πληροφορίες που θεωρούνται απαραίτητες για την εκτέλεση της υπηρεσίας.

Αξίζει να επισημανθεί ότι η γλώσσα OWL-S κάνει χρήση της γλώσσας WSDL, προκειμένου να είναι εφικτή η προσπέλαση και εκτέλεση οποιασδήποτε υπηρεσίας. Αυτό οφείλεται στο γεγονός ότι ο σχεδιασμός της στην παρούσα φάση δεν προβλέπει τη δυνατότητα σύνδεσης των μεθόδων που προσφέρει μια υπηρεσία (και οι οποίες περιγράφονται με σημασιολογικό τρόπο με αυτήν) με κάποιο πρωτόκολλο επικοινωνίας (όπως είναι το SOAP), κάτι που είναι απαραίτητο για την ανταλλαγή μηνυμάτων μεταξύ παρόχου – πελάτη όπως φάνηκε και σε προηγούμενη παράγραφο (§2.1).

2.3.3. Εύρεση Διαδικτυακών Υπηρεσιών

Η περιγραφή των διαδικτυακών υπηρεσιών που προσφέρονται από διάφορους παρόχους καθώς και των αιτήσεων από υποψήφιους πελάτες με την OWL-S, έχει ως αποτέλεσμα λόγω των πολλών δυνατοτήτων που προσφέρονται από τη γλώσσα αυτή, να παρατηρείται πολύ συχνά το ακόλουθο φαινόμενο: το πλήθος των παροχών των υπηρεσιών καθώς και των απαιτήσεων από τη μεριά των πελατών να μην ταυτίζονται. Αντικείμενο της παραγράφου αυτής αποτελεί η παρουσίαση όλων των δυνατών περιπτώσεων που μπορούν να αντιμετωπιστούν από τα διάφορα προγράμματα (agents) κατά τη διαδικασία αυτόματης εύρεσης διαδικτυακών υπηρεσιών (§2.3.1), προκειμένου να συμπεράνουν ότι μια υπηρεσία ικανοποιεί συγκεκριμένες απαιτήσεις. Επισημαίνεται ότι οι περιπτώσεις αυτές στηρίζονται στα IOPEs των υπηρεσιών και πιο συγκεκριμένα στις κλάσεις των παραμέτρων εισόδου (input) και εξόδου (output) αυτών. Ακολουθούν οι δυνατές περιπτώσεις που ισχύει για τις εξόδους, ενώ το αντίστροφο ισχύει για τις εισόδους [61]:

- Ακριβή (Exact): Οι παράμετροι εξόδου της αίτησης από τη μεριά του πελάτη είναι ακριβώς ίδιες ή είναι υποκλάσεις των παραμέτρων εξόδου της υπηρεσίας.

- Μη - Πλήρη (Plugin): Οι παράμετροι εξόδου της αίτησης από τη μεριά του πελάτη περιλαμβάνονται στο σύνολό τους στις παραμέτρους εξόδου της υπηρεσίας.
- Υπερπλήρη (Subsume): Οι παράμετροι εξόδου της αίτησης από τη μεριά του πελάτη περιλαμβάνονται μόνο κατά ένα μέρος στις παραμέτρους εξόδου της υπηρεσίας. Σε αυτή την περίπτωση ο πελάτης της υπηρεσίας δε θα ικανοποιηθεί σε απόλυτο βαθμό, αφού η έξοδος της υπηρεσίας που προσφέρεται υπολείπεται της εξόδου που επιθυμεί.
- Ασύζευκτη (Disjoint): Οι παράμετροι εξόδου της αίτησης από τη μεριά του πελάτη είναι απόλυτα διαφορετικές σε σχέση με τις παραμέτρους εξόδου της υπηρεσίας. Δεν υπάρχει καμία περίπτωση να ικανοποιηθεί ο πελάτης, αφού δεν υπάρχει κάποια σχέση που να οδηγεί σε κάποιο συμπέρασμα μεταξύ των παραμέτρων αυτών.

Η διαδικασία αυτόματης εύρεσης διαδικτυακών υπηρεσιών όπως έχει ήδη αναφερθεί είναι μια από τις σημαντικότερες παροχές που υπόσχονται να προσφέρουν οι σημασιολογικές διαδικτυακές υπηρεσίες (semantic web services). Λόγω της σπουδαιότητας της συμβολής αρκετοί ερευνητές έχουν ασχοληθεί με το σχεδιασμό κατάλληλων αλγορίθμων ταιριάσματος (matchmaking), κάνοντας χρήση των προαναφερθέντων περιπτώσεων (exact, plugin, subsume, disjoint / fail) ([62], [63], [64]). Είναι εύλογο βέβαια ότι οι μεθοδολογίες αυτές εφαρμόζονται μόνο σε περίπτωση που τόσο οι υπηρεσίες όσο και οι αιτήσεις από τους πελάτες στηρίζουν την περιγραφή τους σε οντολογίες.

Κεφάλαιο 3ο

3. Οντολογίες ποιότητας υπηρεσίας (QoS)

Αντικείμενο του κεφαλαίου αυτού αποτελεί ο σχεδιασμός κατάλληλων οντολογιών (οντολογία Γλώσσα και οντολογία Λεξιλόγιο), ώστε να είναι εφικτή η περιγραφή των παραμέτρων ποιότητας των διαφόρων διαδικτυακών υπηρεσιών του σημασιολογικού δικτύου (semantic web). Επιπρόσθετα, θα καταδειχθεί η αναγκαιότητα και σπουδαιότητα παροχής διαδικτυακών υπηρεσιών καθορισμένης ποιότητας και πώς αυτή η ποιότητα μπορεί να καθοριστεί με χρήση συγκεκριμένων παραμέτρων.

Πιο αναλυτικά, στην παράγραφο 3.1 παρουσιάζονται οι λόγοι που οδήγησαν στην παροχή υπηρεσιών προσαρμοσμένων στις ιδιαίτερες προτιμήσεις / απαιτήσεις του χρήστη / πελάτη της υπηρεσίας. Στην παράγραφο 3.2 γίνεται μια εκτενής αναφορά και ανάλυση επιστημονικών εργασιών που πραγματεύονται το σχεδιασμό αντίστοιχων οντολογιών με αυτές που προτείνονται στο κεφάλαιο αυτό, προτείνοντας παράλληλα αρχιτεκτονικές εφαρμογής τους σε αυτές. Το κεφάλαιο ολοκληρώνεται στην παράγραφο 3.3 όπου παρουσιάζονται τα βήματα που ακολουθήθηκαν προκειμένου να σχεδιαστούν οι εν λόγω οντολογίες (§3.3.1), αναλύονται (οντολογία Γλώσσα (§3.3.2) και οντολογία Λεξιλόγιο (§3.3.3)) και αποτιμώνται οι οντολογίες αυτές (§3.3.4), ενώ στο τέλος παρατίθεται ένα παράδειγμα εφαρμογής τους (§3.3.5) σε μια διαδικτυακή υπηρεσία.

3.1. Ποιότητα υπηρεσίας (QoS) σε διαδικτυακές υπηρεσίες

Η ολοένα και μεγαλύτερη ανάπτυξη και εξάπλωση του διαδικτύου, έχει οδηγήσει στην ανάγκη δημιουργίας υπηρεσιών που να ικανοποιούν κάθε επιθυμία και απαίτηση. Οι υπηρεσίες αυτές, γνωστές και ως διαδικτυακές υπηρεσίες (web services) ([7], [8]) (περισσότερες πληροφορίες για τις υπηρεσίες αυτές, υπάρχουν στο προηγούμενο κεφάλαιο), αποτελούν πλέον μια πραγματικότητα με πλήθος εφαρμογές σε όλους τους τομείς, όπως για

παράδειγμα οικονομικές εφαρμογές, εμπορικές εφαρμογές, εφαρμογές που στοχεύουν στη ψυχαγωγία του χρήστη κ.α.. Η εξάπλωση αυτή σηματοδότησε την ανάγκη δημιουργίας και ανάπτυξης ενός νέου κοινού κώδικα επικοινωνίας, ο οποίος όφειλε να είναι αποδεκτός από όλες τις οντότητες που συμμετέχουν στις υπηρεσίες αυτές. Μεταξύ των νέων αρχιτεκτονικών, πρωτοκόλλων και γλωσσών αυτών επικοινωνίας είναι η SOA [10], η WSDL [12], το SOAP [11] και το UDDI [37], τα οποία συνέβαλαν σημαντικά στην εξάπλωση των διαδικτυακών υπηρεσιών. Καθώς όμως η εξάπλωσή τους γινόταν όλο και μεγαλύτερη, όπως είναι λογικό επακόλουθο, δεν άργησαν να εμφανιστούν και τα πρώτα προβλήματα στη χρήση τους. Τα σημαντικότερα από αυτά σχετίζονταν με θέματα που αφορούσαν ότι ο πάροχος της υπηρεσίας αδυνατούσε να προσφέρει αυτά που υποσχόταν [15]. Ποικίλες αιτίες μπορούν να προκαλούσαν το φαινόμενο αυτό, όπως για παράδειγμα ότι ο πάροχος είχε περισσότερες αιτήσεις / πελάτες από ότι μπορούσε να εξυπηρετήσει, ότι το σύστημά του δεν ήταν σχεδιασμένο σωστά, ώστε να υποστηρίξει πελάτες με διαφορετικά πρωτόκολλα / αρχιτεκτονικές / γλώσσες διαδικτύου, είτε ότι τελικά οι υπηρεσίες που απολάμβανε τελικά ο χρήστης ήταν πολύ φτωχές και κατά συνέπεια μη εκμεταλλεύσιμες. Ένα τέτοιο παράδειγμα είναι η υπηρεσία η οποία προσφέρει δεδομένα σε πραγματικό χρόνο τα οποία μεταβάλλονται σημαντικά με την πάροδο του χρόνου. Είναι εύλογο, ότι αν η υπηρεσία έχει μεγάλη καθυστέρηση, αυτό έχει σαν αποτέλεσμα τα δεδομένα που προσφέρει να είναι τελικά άχρηστα στο χρήστη / πελάτη της υπηρεσίας. Το φαινόμενο αυτό, αν και αρκετά σύνηθες κατά την εμφάνιση των διαδικτυακών υπηρεσιών, δεν ήταν ωστόσο το μόνο. Ένα επίσης πολύ σημαντικό πρόβλημα που δεν άργησε να εμφανιστεί και απαιτούσε άμεση αντιμετώπιση, ήταν η ασφάλεια και η αξιοπιστία των προσφερόμενων υπηρεσιών [65]. Όταν πρωτοεμφανίστηκαν οι υπηρεσίες αυτές, δεν υπήρχε τρόπος να διασφαλιστεί ο χρήστης, ούτε προφανώς τα δεδομένα που αντάλλασσε αυτός με την υπηρεσία που χρησιμοποιούσε. Αυτό βέβαια ήταν ένα τεράστιο πρόβλημα, αν αναλογιστεί κανείς ότι λόγω του ότι οι διαδικτυακές υπηρεσίες προσφέρονταν σε κάποιο κόστος, πλήθος ανθρώπων χρεώνονταν για υπηρεσίες που δε χρησιμοποιούσαν ποτέ. Πολύ σημαντικό πρόβλημα ασφαλείας αποτελεί ακόμη το γεγονός ότι κάποιος υποδουμένος κάποιον άλλον, εκμεταλλευόμενος τα κενά ασφαλείας των διαδικτυακών υπηρεσιών στα πρωταρχικά τους στάδια, είχε πρόσβαση σε αρχεία / δεδομένα, με αποτέλεσμα να τα χρησιμοποιούσε προς ίδιον όφελος. Επιπρόσθετα, λόγοι που σχετίζονταν με το διαδίκτυο και την κίνηση σε αυτό, δημιουργούσε ένα επιπλέον πρόβλημα, αφού πολύ συχνά ο χρήστης αδυνατούσε να χρησιμοποιήσει την υπηρεσία, όχι λόγω αδυναμίας του παρόχου της υπηρεσίας αλλά λόγω της υποδομής του δικτύου και των αποτυχιών (failures) αυτού.

Εκτός από τα προβλήματα που ήδη αναφέρθηκαν, η ολοένα μεγαλύτερη ανάπτυξη και εξάπλωση των διαδικτυακών υπηρεσιών, συνοδεύτηκε από τη δημιουργία αντίστοιχου πλήθους παρόχων, με τον κάθε έναν από αυτούς να προσφέρει ίδιες ή παραπλήσιες υπηρεσίες

[15]. Επομένως, για την ίδια υπηρεσία, θα έπρεπε να βρεθούν σαφή πλέον κριτήρια επιλογής για το χρήστη, στα οποία στηριζόμενος να επιλέξει τελικά κάποιο συγκεκριμένο πάροχο.

Όλοι αυτοί οι λόγοι οδήγησαν στην εισαγωγή της ποιότητας υπηρεσίας QoS (Quality of Service). Με τον όρο ποιότητα υπηρεσίας, αναφερόμαστε σε όλες αυτές τις παραμέτρους που μπορούν να χαρακτηρίσουν μια υπηρεσία (όπως για παράδειγμα, διαθεσιμότητα, χρόνος απόκρισης, ασφάλεια της υπηρεσίας) και που δε σχετίζεται με τη λειτουργικότητά της. Οι παράμετροι που χαρακτηρίζουν την ποιότητα μιας υπηρεσίας συχνά χαρακτηρίζονται και ως μη – λειτουργικές παράμετροι, επισημαίνοντας με αυτόν τον τρόπο το γεγονός, ότι δε σχετίζονται με το τι προσφέρει λειτουργικά μια υπηρεσία, αλλά αντίθετα στοχεύουν στο να κάνουν μια υπηρεσία λιγότερο ή περισσότερο ελκυστική σε σχέση με άλλες με την ίδια ακριβώς λειτουργικότητα. Για παράδειγμα, πολλοί πάροχοι προσφέρουν την υπηρεσία βίντεο κατ' απαίτηση (Video On Demand) αλλά με διαφορετικό ρυθμό δεδομένων, κόστος, χρόνο απόκρισης, μεταβολή στην καθυστέρηση των δεδομένων (jitter) κ.α.. Η λειτουργικότητα της υπηρεσίας είναι η παροχή σε πραγματικό χρόνο βίντεο, ενώ αντίθετα οι μη-λειτουργικές παράμετροι, βάσει των οποίων ένας χρήστης θα επιλέξει κάποιον συγκεκριμένο πάροχο είναι ο ρυθμός δεδομένων, το κόστος κτλ..

Επομένως, ο πάροχος όφειλε, προκειμένου να αντιμετωπίσει τα προβλήματα αυτά και να αυξήσει το πελατολόγιό του και κατ' επέκταση και τα κέρδη του, να γίνει πιο ανταγωνιστικός, δημιουργώντας πιο ελκυστικές υπηρεσίες, οι οποίες θα έδιναν λύση στα προβλήματα που επισημάνθηκαν.

Λύσεις που υιοθετήθηκαν από πλήθος παρόχων αποτελούν η εξισορρόπηση φορτίου (load balancing) [66] και ο σχεδιασμός κλιμακωτών συστημάτων, τα οποία δίνουν τη δυνατότητα εξυπηρέτησης πολλών πελατών, ενώ παράλληλα προσαρμόζονται κατάλληλα στην αύξηση της ζήτησης. Ακόμη, με την εισαγωγή μεθόδων κρυπτογράφησης και πιστοποίησης επιτυγχάνεται διασφάλιση τόσο του πελάτη / χρήστη της υπηρεσίας όσο και των δεδομένων που αυτός ανταλλάσσει κατά τη χρήση της. Άλλωστε, με τη δημιουργία πιο σύνθετων διαδικτυακών υπηρεσιών, υπάρχουν ακόμη μεγαλύτερες απαιτήσεις που πρέπει να λαμβάνονται υπόψη κατά το σχεδιασμό τους, αφού μια υπηρεσία πλέον μπορεί να περιλαμβάνει τη χρήση άλλων υπηρεσιών ή να ολοκληρώνεται σε πολλαπλά στάδια, μέσω της διαδικασίας των συναλλαγών (transactions). Επομένως, θα πρέπει οι συναλλαγές αυτές, να υπακούουν σε κάποιους κανόνες, προκειμένου να εξασφαλίζει μια υπηρεσία την ακεραιότητα των δεδομένων της. Οι αρχές αυτές, που αναφέρονται εν συντομία και ως ACID (Atomicity, Consistency, Isolation και Durability) [67], ουσιαστικά πιστοποιούν ότι όλες οι απαραίτητες ενέργειες που μεσολαβούν και συμμετέχουν για την επιτυχή εκτέλεση μιας υπηρεσίας, αντιμετωπίζονται σε μια λογική μονάδα. Με τον όρο ατομικότητα (atomicity) δηλώνεται ότι όλες οι ενέργειες πρέπει να ολοκληρωθούν με επιτυχία, ώστε να θεωρείται η συναλλαγή επιτυχής. Με τον όρο συνοχή (consistency) εξασφαλίζεται προστασία στα

δεδομένα της συναλλαγής ως προς τη συνοχή τους, όταν σε κάποιο ενδιάμεσο στάδιο, υπάρχει μια αλλαγή σε αυτά. Απομόνωση (isolation) επιτυγχάνεται όταν συναλλαγές (transactions) οι οποίες εκτελούνται ταυτόχρονα, δεν επηρεάζει η μία την άλλη μέχρι την ολοκλήρωσή τους. Τέλος, ο τέταρτος όρος (διάρκεια (durability)) σηματοδοτεί ότι οι αλλαγές που γίνονται μετά το πέρας των συναλλαγών δε χάνονται ποτέ. Οι αλλαγές που γίνονται κατά τη διάρκεια των συναλλαγών αυτών γίνονται μόνιμες, μόνο στην περίπτωση που ολοκληρωθούν με επιτυχία όλα τα στάδια / ενέργειες που μεσολαβούν. Σε διαφορετική περίπτωση το σύστημα επανέρχεται στην κατάσταση που ήταν, προτού ξεκινήσει η εκτέλεσή τους.

Είναι προφανές από τα παραπάνω, ότι η χρήση υπηρεσιών που δεν παρέχουν καμία εγγύηση για το αν τελικά μπορούν να προσφέρουν αυτό που διαφημίζουν, διαρκώς θα περιορίζεται και μάλιστα θα επικεντρώνεται σε υπηρεσίες ελάχιστονης σημασίας. Νέοι ορίζοντες έχουν ανοίξει στις διαδικτυακές υπηρεσίες που προσφέρουν πλέον στον τελικό χρήστη πέραν της λειτουργικότητάς τους, πρόσθετες παροχές που εξασφαλίζουν ότι οι προσφερόμενες υπηρεσίες ικανοποιούν επιπροσθέτως ένα δεδομένο επίπεδο ποιότητας. Συνάπτονται σχέσεις μεταξύ χρήστη και παρόχου υπηρεσίας (Service Level Agreements (SLAs)) [68], που μεταξύ άλλων οριοθετούν τους όρους και το επίπεδο ποιότητας της προσφερόμενης υπηρεσίας. Όλα αυτά στοχεύουν στην εξυπηρέτηση του πελάτη, απολαμβάνοντας προσωποποιημένες και προσαρμοσμένες στις ιδιαίτερες προτιμήσεις και απαιτήσεις του υπηρεσίες, καταβάλλοντας βέβαια πάντα το ανάλογο αντίτιμο, ενώ από τη μεριά του παρόχου η προσφορά πιο ανταγωνιστικών και ελκυστικών υπηρεσιών για τον τελικό χρήστη αποσκοπούν στην ενίσχυση των κερδών του.

3.2. Σχετική βιβλιογραφία

Στην παράγραφο αυτή θα αναλυθούν εργασίες της παγκόσμιας βιβλιογραφίας που αναλύουν τεχνικές σχεδιασμού οντολογιών και πιο ειδικά οντολογιών για παραμέτρους ποιότητας υπηρεσιών. Πρέπει να επισημανθεί ότι οι εργασίες αυτές χρησιμοποιούν τις οντολογίες που προτείνουν, προκειμένου να υποστηρίξουν κάποια προτεινόμενη αρχιτεκτονική για διαδικτυακές υπηρεσίες. Ωστόσο, επειδή το αντικείμενο του κεφαλαίου επικεντρώνεται κυρίως σε οντολογίες, ο σχολιασμός των εργασιών θα περιοριστεί στις προτεινόμενες οντολογίες και όχι στην αρχιτεκτονική ή στους αλγορίθμους επιλογής - ταιριάσματος (matchmaking) [32] που παρουσιάζονται. Επιπρόσθετα, επειδή κάποιες εργασίες (όπως για παράδειγμα η [69]) οριοθετούν το πρόβλημα της έλλειψης παροχής ποιότητας υπηρεσίας με τρόπο ώστε να αποτελούν εργασία αναφοράς για πολλές άλλες, κρίνεται σκόπιμο να συμπεριληφθούν στην ανάλυση αυτή.

Η πιο ενδιαφέρουσα εργασία από αυτές που μελετήθηκαν είναι των Maximilien και Singh [70]. Η εργασία αυτή προτείνει μια αρχιτεκτονική βασισμένη σε πράκτορες (agents).

Αρχικά επισημαίνει την ανυπαρξία διαδικτυακών υπηρεσιών οι οποίες προσφέρουν καθορισμένο επίπεδο ποιότητας υπηρεσίας με δυναμικό τρόπο. Κατόπιν, υποδεικνύει για ποιο λόγο οι υπάρχουσες αρχιτεκτονικές αδυνατούν να το υλοποιήσουν αυτό και προτείνουν λύση βασισμένη σε πράκτορες, που θα δώσει τη δυνατότητα στους παρόχους των υπηρεσιών να διαφημίζουν τις υπηρεσίες τους, οι οποίες προσφέρουν ένα δεδομένο επίπεδο ποιότητας. Παράλληλα μέσω της προτεινόμενης αρχιτεκτονικής δίνεται η δυνατότητα στο χρήστη της υπηρεσίας δυναμικά, να επιλέγει υπηρεσίες όχι μόνο βάσει της λειτουργικότητάς της αλλά ικανοποιώντας τις ιδιαίτερες προτιμήσεις / απαιτήσεις του.

Πιο συγκεκριμένα, η αρχιτεκτονική που προτείνεται στηρίζεται στις υπάρχουσες τεχνολογίες και αρχιτεκτονικές για διαδικτυακές υπηρεσίες (WSDL [12] και UDDI [37]). Κατόπιν, προσθέτουν μια διεπαφή (interface) – πράκτορα στην κάθε υπηρεσία του παρόχου, στις μονάδες εγγραφών (registries) UDDI και στον κάθε χρήστη, με αποτέλεσμα όλη η επικοινωνία να ελέγχεται από αυτούς τους πράκτορες. Η δυναμικότητα της προτεινόμενης αρχιτεκτονικής και η δυνατότητα περιγραφής των παραμέτρων ποιότητας των διαδικτυακών υπηρεσιών γίνεται με τη χρήση οντολογιών. Υπάρχουν δύο κατηγορίες οντολογιών: Η οντολογία υπηρεσία που ουσιαστικά συσχετίζει την κάθε υπηρεσία με τις παραμέτρους ποιότητάς της και η οντολογία QoS που περιγράφει τις παραμέτρους ποιότητας σε 3 επίπεδα. Το πρώτο επίπεδο και το πιο αφαιρετικό (και το οποίο ταυτίζεται με την οντολογία Γλώσσα που προτείνεται στο κεφάλαιο αυτό) περιγράφει γενικά χαρακτηριστικά (attributes) και σχέσεις μεταξύ μιας παραμέτρου ποιότητας με μια άλλη. Τα σημαντικότερα από αυτά είναι το χαρακτηριστικό Quality που ουσιαστικά είναι ο γενικότερος όρος αυτής της οντολογίας και παριστάνει μια οποιαδήποτε παράμετρο, το χαρακτηριστικό QRelationship που μοντελοποιεί τη σχέση μιας παραμέτρου με μια άλλη και το χαρακτηριστικό AggregateQuality το οποίο υποδεικνύει ότι κάποια παράμετρος προκύπτει σαν άθροισμα άλλων παραμέτρων. Όσον αφορά το μεσαίο επίπεδο (και το οποίο ταυτίζεται με την οντολογία Λεξιλόγιο που προτείνεται στο κεφάλαιο αυτό), αυτό περιέχει τις διάφορες παραμέτρους (όπως για παράδειγμα χρόνος απόκρισης, διαθεσιμότητα κ.α.) και σχέσεις μεταξύ αυτών (όπως για παράδειγμα ότι οι παράμετροι πιστοποίηση και κωδικοποίηση ανήκουν στη γενική κατηγορία ασφάλεια). Το τρίτο επίπεδο δεν αναλύεται κάπου στην εργασία αυτή. Τα δύο επίπεδα ωστόσο που αναλύονται είναι αρκετά πλήρη. Ένα μειονέκτημα της οντολογίας του πρώτου επιπέδου είναι ότι δεν υπάρχει τρόπος περιγραφής της τιμής και των μονάδων της κάθε παραμέτρου. Όσον αφορά την οντολογία του μεσαίου επιπέδου, αυτή εμφανίζεται αρκετά πλήρης, που ωστόσο απουσιάζουν και εδώ κάποιοι όροι (όπως για παράδειγμα μεταβολή καθυστέρησης (jitter) ή προσπελασιμότητα (accessibility)), παράμετροι που απαντώνται συχνά σε διαδικτυακές υπηρεσίες.

Μια επίσης πολύ ενδιαφέρουσα εργασία είναι αυτή των Zhou, Chia και Lee [64]. Η εργασία αυτή προτείνει μια οντολογία με παραμέτρους ποιότητας, στηριζόμενη στη γλώσσα

DAML-S (Darpa Agent MarkUp Language - Service) (πρόγονος της OWL-S) [13]. Όπως υποστηρίζουν οι συγγραφείς, η εύρεση και επιλογή διαδικτυακών υπηρεσιών με χρήση παραμέτρων ποιότητας υπηρεσίας, ενώ αποτελεί αναγκαιότητα, δεν υποστηρίζεται από κάποια υπάρχουσα γλώσσα – ούτε και από τη DAML-S – για διαδικτυακές υπηρεσίες. Επιπρόσθετα, στην εργασία αυτή προτείνεται ένας αλγόριθμος επιλογής – ταιριάσματος (matchmaking) [32] διαδικτυακών υπηρεσιών.

Πιο συγκεκριμένα, στην εργασία αυτή επεκτείνεται η γλώσσα DAML-S, η οποία χρησιμοποιείται για περιγραφή διαδικτυακών υπηρεσιών με χρήση οντολογιών, προκειμένου να γίνουν οι υπηρεσίες όχι μόνο κατανοητές αλλά και διαχειρίσιμες από τον υπολογιστή. Η επέκταση που γίνεται, αφορά κυρίως όρους που περιγράφουν τις παραμέτρους ποιότητας των υπηρεσιών. Η νέα γλώσσα που προτείνεται είναι η DAML-QoS και η οποία εκτός από την προαναφερθείσα προσθήκη, είναι - σύμφωνα με τους συγγραφείς - πιο κατανοητή από τους υπολογιστές, πιο εύκολα διαχειρίσιμη και επεκτάσιμη.

Η προτεινόμενη οντολογία, η οποία δημιουργήθηκε με σκοπό να χρησιμοποιηθεί κυρίως στην επιλογή διαδικτυακών υπηρεσιών, αποτελείται από 3 επίπεδα: Το επίπεδο του προφίλ το οποίο θα χρησιμοποιηθεί από τον αλγόριθμο επιλογής – ταιριάσματος, το επίπεδο του ορισμού ιδιοτήτων το οποίο θα ορίζει τις ιδιότητες και τις σχέσεις μεταξύ τους, ώστε να είναι επεξεργάσιμες από τους υπολογιστές, και το μετρικό επίπεδο το οποίο χρησιμοποιείται για ό,τι έχει σχέση με μετρήσεις. Πιο αναλυτικά, το πρώτο επίπεδο δύναται εναλλακτικά να απαρτίζεται από 3 οντολογίες: Από την οντολογία του παρόχου υπηρεσιών που διαφημίζει την υπηρεσία του (ProviderQoS), από την οντολογία του πελάτη που αναζητά την υπηρεσία (InquiryQoS) ή από την οντολογία πρότυπο που αποθηκεύεται προσωρινά για να χρησιμοποιηθεί μελλοντικά τροποποιημένη (TemplateQoS). Ωστόσο, για να υπάρχει ένας κοινός κώδικας επικοινωνίας μεταξύ των οντολογιών αυτών, θα πρέπει αυτές να στηρίζονται στο δεύτερο επίπεδο (επίπεδο ορισμού ιδιοτήτων). Το επίπεδο αυτό βρίσκεται σε πλήρη αντιστοιχία με αυτό που προτείνεται και στη DAML-S [13], επομένως έχει 4 υποεπίπεδα: είσοδο (QoS input), πυρήνα (QoS core), προαπαιτήσεις (QoS preconditions) και συνέπειες (QoS affects). Το τρίτο επίπεδο υποστηρίζει μεταξύ άλλων και σύνθετες μονάδες μέτρησης καθώς και σχέσεις μετατροπής από τη μια μονάδα μέτρησης στην άλλη.

Προκειμένου να επιδείξουν τις δυνατότητες της προτεινόμενης οντολογίας και να εφαρμόσουν τον αλγόριθμο ταιριάσματος, προτείνουν μια υπηρεσία με 4 παραμέτρους ποιότητας: χρόνος απόκρισης, κόστος, αξιοπιστία και ρυθμός δεδομένων. Ο προτεινόμενος αλγόριθμος ταιριάσματος υποστηρίζει τις περιπτώσεις: υπερπλήρη (subsume), ακριβή, (exact), μη-πλήρη (plugin) και ασύζευκτη (disjoint) [32], ανάλογα με το βαθμό ταιριάσματος της διαφημιζόμενης υπηρεσίας με την αίτηση του χρήστη (στοιχεία για τις περιπτώσεις αυτές υπάρχουν στην §2.3.3). Μειονέκτημα της προτεινόμενης οντολογίας αποτελεί το γεγονός ότι δημιουργήθηκε για να χρησιμοποιηθεί αυστηρά στην επιλογή υπηρεσιών με αποτέλεσμα να

παρουσιάζει αρκετές ελλείψεις και να μη μπορεί να χρησιμοποιηθεί ως μια γενική οντολογία Γλώσσα, ικανή να εφαρμοστεί και σε άλλες αρχιτεκτονικές. Επιπρόσθετα, το γεγονός ότι δεν προτείνεται κάποια οντολογία Λεξιλόγιο, οδηγεί στο συμπέρασμα ότι θα στηριχθούν σε άλλες υλοποιήσεις.

Μια εξίσου ενδιαφέρουσα εργασία είναι αυτή που παρουσιάζεται στην [71]. Οι συγγραφείς αρχικά υποδεικνύουν την ανάγκη δημιουργίας μιας αρχιτεκτονικής που προσφέρει τη δυνατότητα σύνθεσης διαδικτυακών υπηρεσιών. Όπως υποστηρίζουν, οι ολοένα και αυξανόμενες ανάγκες σε διαδικτυακές υπηρεσίες οι οποίες προσφέρουν ένα δεδομένο επίπεδο ποιότητας και η ανάγκη για αυτοματοποιημένη διαδικασία σύνθεσης αυτών των υπηρεσιών, καθιστά τη δημιουργία τέτοιας αρχιτεκτονικής μείζονος σημασίας. Η προτεινόμενη αρχιτεκτονική (WebQ) προσφέρει τη δυνατότητα σύνθεσης διαδικτυακών υπηρεσιών δυναμικά ως μέρος μιας εργασίας ροής (workflow) [72], ενώ βρίσκει εφαρμογή σε πλήθος τομέων / περιοχών όπως για παράδειγμα ηλεκτρονικό εμπόριο και ιατρική.

Πιο αναλυτικά, η προτεινόμενη αρχιτεκτονική αποτελείται από 4 στάδια: Στο πρώτο στάδιο, αρχικοποιείται η εργασία ροής στηριζόμενη στη γλώσσα DAML-S [13]. Στη συνέχεια καθορίζονται οι παράμετροι ποιότητας υπηρεσίας που θα χρησιμοποιηθούν. Αυτές χωρίζονται σε 3 κατηγορίες: Στις παραμέτρους που αφορούν τη δεδομένη διαδικτυακή υπηρεσία μόνο, στις παραμέτρους που αφορούν το δίκτυο καθώς και στις γενικές παραμέτρους που δεν εντάσσονται στις υπόλοιπες δύο κατηγορίες. Στο τρίτο στάδιο, επιλέγονται όλες οι υπηρεσίες που θα χρησιμοποιηθούν σε αυτή την εργασία ροής, αποδίδοντας σε κάθε μία από αυτές έναν παράγοντα Ω , που ουσιαστικά παριστάνει το βαθμό ταιριάσματος της δεδομένης υπηρεσίας για τη δεδομένη εργασία ροής. Στο τέταρτο και τελευταίο στάδιο, που είναι και το πιο σημαντικό, γίνεται η δυναμική σύνθεση και εκτέλεση της εργασίας ροής με χρήση των διαδικτυακών αυτών υπηρεσιών, ενώ παράλληλα υπάρχει και η δυνατότητα παρακολούθησης του παράγοντα Ω , ώστε να είναι εφικτή η επιλογή κάποιας άλλης υπηρεσίας, όταν αυτή γίνει πιο ελκυστική.

Όσον αφορά τις παραμέτρους που χρησιμοποιούνται στο δεύτερο στάδιο, αυτές είναι: (i) παράμετροι δικτύου: διαθεσιμότητα, ασφάλεια, προσπελασιμότητα και συμβατότητα, (ii) γενικές παράμετροι: καθυστέρηση, ρυθμός δεδομένων, αξιοπιστία, κόστος, ενώ όσον αφορά τις παραμέτρους που αφορούν τη δεδομένη υπηρεσία (iii) αυτό που επισημαίνεται στην εργασία είναι ότι προσμετρούνται κατά τη διαδικασία σύνθεσης και επιλογής σαν το άθροισμα του γινομένου της κάθε παραμέτρου με το ποσοστό ταιριάσματος της δεδομένης υπηρεσίας για τη δεδομένη ροή εργασίας. Μειονέκτημα και αυτής της εργασίας είναι ότι εμφανίζεται αρκετά ελλιπής η οντολογία Γλώσσα, ενώ η προτεινόμενη οντολογία Λεξιλόγιο, περιορίζεται στην απαρίθμηση κάποιων λιγοστών παραμέτρων ποιότητας υπηρεσίας.

Μια άλλη εργασία που χρήζει προσοχής είναι αυτή των Sánchez-Macián, Bellido και Pastor [73]. Στην εργασία αυτή, αρχικά επιχειρείται μια αποτίμηση των υπάρχουσών

αρχιτεκτονικών στηριζόμενων σε οντολογίες. Κατόπιν, προτείνεται μια νέα οντολογία, η οποία στοχεύει στη μοντελοποίηση και διαχείριση της αντίληψης που έχει ο χρήστης για την ποιότητα υπηρεσίας που του προσφέρεται. Η εργασία ολοκληρώνεται με δύο πιθανές αρχιτεκτονικές, οι οποίες στοχεύουν στο να συλλέξουν τις επιθυμίες του χρήστη βάσει της προτεινόμενης νέας οντολογίας.

Πιο αναλυτικά, η εργασία επισημαίνει το πρόβλημα που υπάρχει στις υπάρχουσες προσεγγίσεις, οι οποίες αγνοούν το βαθμό αντίληψης που έχει ο χρήστης για την ποιότητα της υπηρεσίας που απολαμβάνει. Ένας τρόπος ένταξης της πολύ σημαντικής αυτής παραμέτρου υλοποιείται υιοθετώντας μεθοδολογίες - κάτι που γίνεται και στην εργασία αυτή - όπως η ServQual [74]. Η μεθοδολογία αυτή στηρίζεται ουσιαστικά σε ένα ερωτηματολόγιο με ερωτήσεις που είναι σημαντικές για όλες τις υπηρεσίες, όπως για παράδειγμα το πόσο αντιληπτή είναι, πόσο ασφαλής, πόσο αξιόπιστη κ.α.. Η προτεινόμενη οντολογία (ServQual) θα χρησιμοποιηθεί για να ερωτηθεί ο χρήστης, για να συλλέξει τη γνώμη του και για να ταξινομήσει / επεξεργαστεί την απάντησή του. Η οντολογία αυτή αποτελεί μέρος μιας αρχιτεκτονικής, που περιλαμβάνει μεταξύ άλλων άλλες δύο οντολογίες απαραίτητες για την υλοποίηση του στόχου. Η πρώτη είναι μια οντολογία Γλώσσα, η οποία περιλαμβάνει όρους όπως αποτίμηση της υπηρεσίας που προσφέρεται από τον πάροχο, ημερομηνία αποτίμησης της υπηρεσίας, ή αν η υπηρεσία αποτιμάται στηριζόμενη σε αντικειμενικά ή υποκειμενικά κριτήρια. Η δεύτερη οντολογία που αποτελεί τη βάση στήριξης της οντολογίας ServQual, είναι η οντολογία ερωτηματολόγιο που περιέχει όρους όπως ερώτηση, απάντηση, τύπος ερώτησης κ.α.. Πρέπει να επισημανθεί εδώ, ότι ο τρόπος περιγραφής της οντολογίας Γλώσσας, όπως αυτός γίνεται στην παρούσα εργασία, δεν είναι σαφής, με αποτέλεσμα να θεωρείται η οντολογία αυτή αρκετά ελλιπής. Στηριζόμενοι λοιπόν στις οντολογίες αυτές, προτείνουν την ServQual, η οποία περιλαμβάνει όρους όπως UserDimension, ο οποίος μοντελοποιεί ουσιαστικά το ποσοστό αντίληψης της υπηρεσίας από το χρήστη καθώς και το τι περιμένει από αυτήν, ή ο όρος ServQualAttributeAverage, ο οποίος αποτιμά τη μέση τιμή των απαντήσεων του χρήστη για μια δεδομένη παράμετρο κ.α.. Η εργασία αυτή παρά τις ελλείψεις της στην οντολογία Γλώσσα και παραβλέποντας το γεγονός ότι δεν προτείνει κάποια οντολογία Λεξιλόγιο, ωστόσο προτείνει οντολογίες που μοντελοποιούν κάτι εξαιρετικά σημαντικό για τις διαδικτυακές υπηρεσίες: το βαθμό αντίληψης του χρήστη για το επίπεδο της ποιότητας της υπηρεσίας που του προσφέρεται από τον πάροχο. Παράμετρος που πρέπει να λαμβάνεται οπωσδήποτε υπόψη από αυτόν, εφόσον στοχεύει στην ικανοποίηση των πελατών του, ενισχύοντας κατ' αυτόν τον τρόπο την ανταγωνιστικότητά του και κατά συνέπεια και τα κέρδη του.

Την αδυναμία παροχής καθορισμένου επιπέδου ποιότητας από τις υπάρχουσες διαδικτυακές υπηρεσίες είναι το αντικείμενο μελέτης και της εργασίας που παρουσιάζεται στην [75]. Όπως και στην προηγούμενη εργασία, έτσι και σε αυτή καινοτομία αποτελεί το

γεγονός ότι προσεγγίζει το πρόβλημα από τη μεριά του χρήστη, δίνοντας τη δυνατότητα σε αυτόν να παρακολουθεί σε πραγματικό χρόνο τις παραμέτρους ποιότητας υπηρεσίας που προσφέρονται από τους διάφορους παρόχους υπηρεσιών και στηριζόμενος στα μεγέθη που υπολογίζονται να επιλέγει την υπηρεσία που ικανοποιεί περισσότερο τις τρέχουσες ανάγκες του.

Η αρχιτεκτονική που προτείνεται στην εργασία αυτή, στηρίζεται σε ένα σχήμα (schema) XML (XSD) για διαδικτυακές υπηρεσίες, το οποίο χρησιμοποιείται τόσο από τον πάροχο της υπηρεσίας όσο και από τον πελάτη της υπηρεσίας. Με τη χρήση αυτού, δίνεται η δυνατότητα στον πάροχο να ορίζει και να διαφημίζει δυναμικά τις υπηρεσίες του, ενώ ο πελάτης έχει τη δυνατότητα εκτός από το να αιτείται υπηρεσίες, να παρακολουθεί τις παραμέτρους ποιότητας υπηρεσίας που του προσφέρονται σε πραγματικό χρόνο, όπως άλλωστε έχει ήδη αναφερθεί. Το ταίριασμα (matchmaking) μεταξύ των προσφερόμενων και αιτούμενων υπηρεσιών επιτυγχάνεται με τη χρήση ενός μεσίτη (broker) για διαδικτυακές υπηρεσίες που ενεργοποιείται από τη μεριά του πελάτη. Ο μεσίτης αυτός επικοινωνεί με τις υπάρχουσες μονάδες εγγραφών (registries) στο UDDI [37] και επιλέγει βάσει των κριτηρίων του χρήστη την καταλληλότερη υπηρεσία γι' αυτόν, δίνοντας προτεραιότητα στο κόστος της υπηρεσίας που θα επιλεγεί τελικά.

Το μοντέλο του προτεινόμενου σχήματος (schema) περιλαμβάνει όρους που θα μπορούσαν να θεωρηθούν σαν τμήματα οντολογίας Γλώσσας (όπως για παράδειγμα οι όροι `qoSDefinitionInfo`, `ContractAndMonitoring` κ.α.) ή σαν τμήματα οντολογίας Λεξιλόγιο (όπως για παράδειγμα οι όροι `χρόνος επεξεργασίας`, `ασφάλεια` κ.α.). Ειδικότερα, όσον αφορά το τμήμα οντολογίας Λεξιλόγιο, προτείνεται μια ιεράρχηση των παραμέτρων αυτών σύμφωνα με το αν αφορούν τις παραμέτρους ποιότητας του παρόχου (όπως για παράδειγμα `διαθεσιμότητα`, `αξιοπιστία` κ.α.), αν αφορούν την υποδομή του δικτύου (όπως για παράδειγμα `καθυστέρηση`, `αριθμός πακέτων που χάνονται` κ.α.), αν αφορούν θέματα ασφάλειας και συναλλαγών (transactions), ενώ έχει προβλεφθεί και η περίπτωση της επέκτασης του προτεινόμενου σχήματος με τον ορισμό ενός κενού πεδίου που προορίζεται για μελλοντική χρήση. Μειονέκτημα της εργασίας αυτής αποτελεί το γεγονός ότι δεν προτείνεται κάποια οντολογία αλλά ένα σχήμα (schema) XML, το οποίο θα πρέπει να προσαρμοστεί και να τροποποιηθεί κατάλληλα, προκειμένου να είναι εφικτή η συνεργασία του με γλώσσες για διαδικτυακές υπηρεσίες, όπως για παράδειγμα η OWL-S [14]. Ένα άλλο μειονέκτημα του προτεινόμενου σχήματος είναι ότι δεν εξηγούνται επαρκώς οι όροι που το απαρτίζουν.

Η τελευταία εργασία που θα αναλυθεί είναι η [69], η οποία μελετά το πρόβλημα της έλλειψης παροχής ποιότητας στις υπάρχουσες διαδικτυακές υπηρεσίες. Η εργασία αυτή, αν και δεν προτείνει κάποια οντολογία Λεξιλόγιο ή οντολογία Γλώσσα, ωστόσο προτείνει κάποιες ενδιαφέρουσες λύσεις για την αντιμετώπιση του προβλήματος αυτού. Προτείνει ένα μοντέλο για επιλογή διαδικτυακών υπηρεσιών, που στηρίζεται αφενός στο τι προσφέρει η

κάθε υπηρεσία και αφετέρου στην ποιότητά της, όπως αυτή διαφημίζεται από τον πάροχο. Επιπρόσθετα, απαριθμεί και αναλύει τους περισσότερους όρους ποιότητας υπηρεσίας που συναντώνται στις διαδικτυακές υπηρεσίες, προσφέροντας παράλληλα ένα μοντέλο ταξινόμησης των όρων αυτών.

Πιο συγκεκριμένα, επεκτείνεται το υπάρχον μοντέλο UDDI [37] με τη χρήση ενός νέου κόμβου πιστοποίησης της ποιότητας υπηρεσίας που προσφέρει ένας πάροχος. Ρόλος του κόμβου αυτού είναι να ελέγχει και να πιστοποιεί την ποιότητα υπηρεσίας που προσφέρει ένας πάροχος, όπως αυτή διαφημίζεται στις μονάδες εγγραφών (registries) του μοντέλου UDDI. Με τη χρήση του κόμβου αυτού, επιτυγχάνεται εύρεση υπηρεσιών που ταιριάζουν περισσότερο στις προτιμήσεις / απαιτήσεις του χρήστη, αφού η υπηρεσία που τελικά θα επιλεγεί, ικανοποιεί όχι μόνο το είδος της υπηρεσίας που επιθυμεί αλλά και το επίπεδο ποιότητας που την επιθυμεί. Κατόπιν, η εργασία αναλύει και ταξινομεί τους πιο συχνά χρησιμοποιούμενους όρους, που καθορίζουν την ποιότητα μιας υπηρεσίας. Η ταξινόμηση περιλαμβάνει 4 ομάδες. Η πρώτη ομάδα περιλαμβάνει όρους που σχετίζονται με την ποιότητα της προσφερόμενης υπηρεσίας κατά την εκτέλεσή της, όπως για παράδειγμα χωρητικότητα, διαθεσιμότητα κ.α.. Η δεύτερη ομάδα περιλαμβάνει όρους που σχετίζονται με συναλλαγές (transactions) και πιο συγκεκριμένα με την ακεραιότητα (integrity) των δεδομένων. Η τρίτη ομάδα περιλαμβάνει όρους που αφορούν είτε τα πρωτόκολλα που υποστηρίζει η υπηρεσία ή γενικά όρους που σχετίζονται με κανονισμούς που πρέπει να είναι συμβατή η υπηρεσία είτε όρους που σχετίζονται με το κόστος της υπηρεσίας. Η τέταρτη και τελευταία ομάδα περιλαμβάνει όρους που σχετίζονται με την ασφάλεια της υπηρεσίας, όπως για παράδειγμα πιστοποίηση, κρυπτογράφηση δεδομένων κ.α..

3.3. Σχεδιασμός οντολογιών παραμέτρων ποιότητας

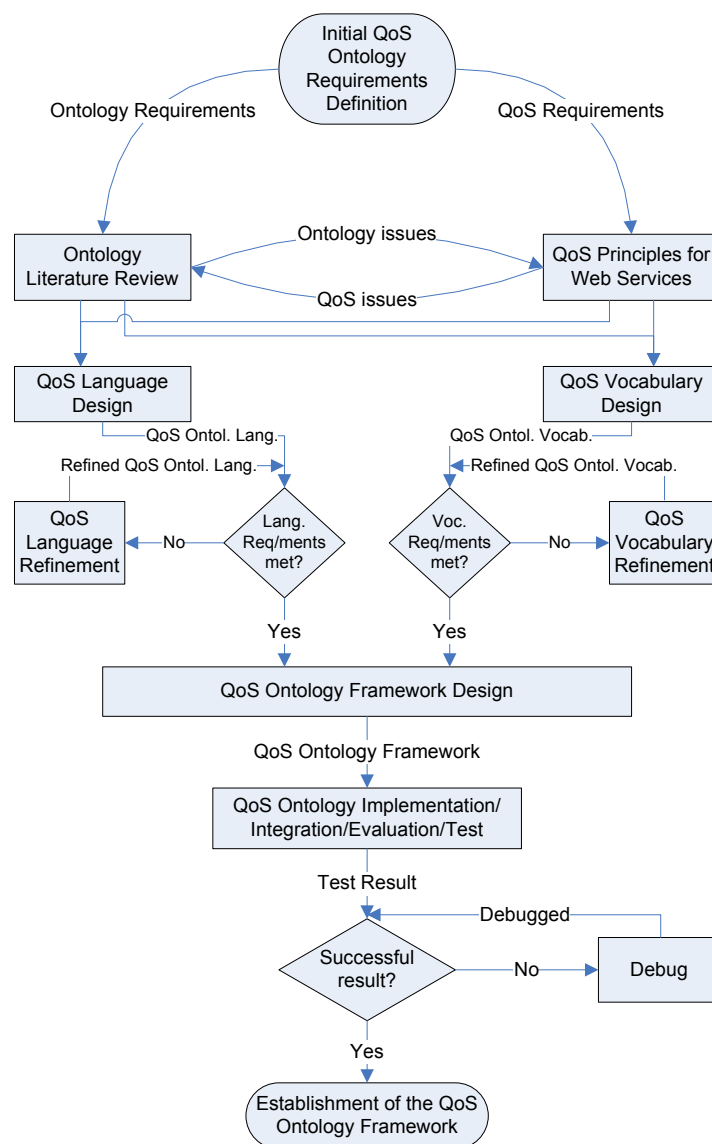
Σε αυτήν την παράγραφο θα προταθούν δύο οντολογίες: Η οντολογία Γλώσσα (§3.3.2) και η οντολογία Λεξιλόγιο (§3.3.3). Αρχικά, θα παρουσιαστούν τα βήματα που ακολουθήθηκαν, προκειμένου να σχεδιαστούν οι οντολογίες αυτές. Στη συνέχεια, ακολουθεί η ανάλυσή και η αποτίμησή τους, ενώ η παράγραφος ολοκληρώνεται με ένα παράδειγμα εφαρμογής των οντολογιών αυτών σε μια διαδικτυακή υπηρεσία, παραθέτοντας τα αντίστοιχα κομμάτια κώδικα. Επισημαίνεται, ότι σαν εργαλείο σχεδιασμού των οντολογιών αυτών χρησιμοποιήθηκε το Protégé [76], ενώ ο πηγαίος κώδικας των οντολογιών αυτών παρατίθεται στη συνολική μορφή του στο Παράρτημα Α.

3.3.1. Βήματα σχεδιασμού οντολογιών

Ακρογωνιαίος λίθος ενός σωστού σχεδιασμού αποτελεί η ορθή και πλήρης καταγραφή όλων των απαιτήσεων του συστήματος που πρέπει να σχεδιαστεί. Ακολουθώντας και υιοθετώντας την ίδια τακτική και στο σχεδιασμό των οντολογιών που προτείνονται στην

παράγραφο αυτή, το πρώτο βήμα που υλοποιείται είναι ο εντοπισμός και η καταγραφή των απαιτήσεων που πρέπει να ικανοποιούν οι οντολογίες αυτές. Προκειμένου να επιτευχθεί αυτό, κρίθηκε απαραίτητο να γίνει σε πρώτη φάση ένας διαχωρισμός της συνολικής διαδικασίας σχεδιασμού σε δύο συμπληρωματικές διαδικασίες, όπως αναπαριστάται και στο σχήμα που ακολουθεί (Σχήμα 3-1). Η πρώτη διαδικασία περιλαμβάνει την εκτεταμένη μελέτη ανάλογων εργασιών της παγκόσμιας βιβλιογραφίας, ενώ στόχος της δεύτερης διαδικασίας είναι η καταγραφή των απαιτήσεων που πρέπει να ικανοποιούν οι οντολογίες αυτές, υπό το πρίσμα και όπως αυτές καταδεικνύονται από τις αρχιτεκτονικές των διαφόρων υπαρχουσών διαδικτυακών υπηρεσιών. Απόρροια της εξέλιξης και αλληλεπίδρασης αυτών των δύο διαδικασιών ήταν μια πολύ σημαντική απόφαση: η δημιουργία 2 οντολογιών (οντολογίας Γλώσσας και οντολογίας Λεξιλόγιο), η οποία συνοδεύτηκε από έναν πρωτογενή σχεδιασμό τους. Επισημαίνεται ότι ο σχεδιασμός αυτός στηρίχθηκε σε κάποιους βασικούς κανόνες που πρέπει να υπηρετούν οι οντολογίες αυτές, οι οποίοι μεταξύ άλλων είναι: σαφήνεια, επεκτασιμότητα, ευελιξία, πληρότητα και κατανόηση τόσο για τον απλό χρήστη όσο και για το σχεδιαστή διαδικτυακών υπηρεσιών / αρχιτεκτονικών που θα χρησιμοποιήσει τις οντολογίες αυτές (περισσότερες πληροφορίες για τους κανόνες αυτούς υπάρχουν στην παράγραφο §3.3.4 που γίνεται αποτίμηση των προτεινόμενων οντολογιών). Η διαδικασία συνεχίζει με τη βελτίωση των οντολογιών που σχεδιάστηκαν αρχικά, μελετώντας και διορθώνοντας τις όποιες αδυναμίες και ασυνέπειές τους σε σχέση με τις απαιτήσεις που πρέπει να πληρούν, όπως αυτές καταγράφηκαν στο προηγούμενο βήμα. Αξίζει να αναφερθεί ότι αυτή είναι μια επαναληπτική και αρκετά επίπονη διαδικασία λόγω των σχέσεων και αλληλεπιδράσεων που έχουν μεταξύ τους οι δύο αυτές οντολογίες, απαιτώντας υπομονή, επιμονή και συνδυαστικές ικανότητες. Η ολοκλήρωση του βήματος αυτού οδήγησε στη δημιουργία ενός πλαισίου (framework) σχεδιασμού για οντολογίες ποιότητας υπηρεσίας. Στο πλαίσιο αυτό οι εν λόγω οντολογίες ενοποιήθηκαν, ενώ οι όποιες ασυμβατότητες και αστοχίες υπήρχαν, αντιμετωπίστηκαν όπως προαναφέρθηκε, προκειμένου να επιτευχθεί άρρηκτη και λειτουργική σχέση συνεργασίας μεταξύ τους. Με την επίτευξη της δημιουργίας του πλαισίου αυτού, το μόνο που έχει απομείνει προκειμένου να ολοκληρωθεί η συνολική διαδικασία, είναι το στάδιο υλοποίησης των δύο οντολογιών. Επομένως, σε αυτό το βήμα πρέπει να επιλεγθεί αφενός η γλώσσα που θα χρησιμοποιηθεί για να περιγράψει τις δύο αυτές οντολογίες και αφετέρου το εργαλείο ανάπτυξης. Η υλοποίησή τους, όπως έχει ήδη αναφερθεί άλλωστε, στηρίχθηκε στη γλώσσα OWL [5] (§2.2.4) η οποία αποτελεί την πιο διαδεδομένη, πλήρη και διαρκώς εξελισσόμενη γλώσσα για οντολογίες, ενώ σαν εργαλείο ανάπτυξης χρησιμοποιήθηκε το Protégé [76], το οποίο αποτελεί το πιο δημοφιλές και αξιόπιστο εργαλείο λογισμικού για οντολογίες. Η συνολική διαδικασία τερματίζεται, αφού ακολουθηθεί μια επαναληπτική διαδικασία ελέγχου λαθών με αυτό το εργαλείο ανάπτυξης, καθορίζοντας την τελική μορφή των οντολογιών αυτών.

Τα βήματα που αναλύθηκαν, σκιαγραφούνται στο σχήμα που ακολουθεί. Επισημαίνεται ότι σε όλα τα σχήματα του κεφαλαίου αυτού, επιλέχθηκε η αγγλική γλώσσα, για λόγους πλήρους συμβατότητας και κατανόησης αφενός με τη γλώσσα OWL [5] και αφετέρου με το Παράρτημα Α.



Σχήμα 3-1 Βήματα σχεδιασμού οντολογιών ποιότητας υπηρεσίας

3.3.2. Οντολογία Γλώσσα

Στην παράγραφο αυτή θα αναλυθεί η οντολογία Γλώσσα [17]. Η οντολογία αυτή (Σχήμα 3-2) αποτελεί ένα γενικό μοντέλο αναπαράστασης παραμέτρων ποιότητας υπηρεσίας, των σχέσεων που μπορούν να υπάρχουν μεταξύ τους, ενώ παράλληλα δίνεται η δυνατότητα μέσω της προτεινόμενης μοντελοποίησης να γίνει πλήρης καταγραφή των τιμών και των μονάδων της κάθε μιας από αυτές. Στην οντολογία αυτή οι περισσότεροι όροι σχετίζονται με την κλάση QoS Παράμετρο (QoS Parameter), που παριστάνει το γενικότερο όρο με τον οποίο μπορεί να αναφερθεί κανείς σε μια παράμετρο ποιότητας υπηρεσίας. Επιπρόσθετα, κάποιες

κλάσεις συνδέονται με άλλες κλάσεις ή είναι υποκλάσεις κάποιων άλλων κλάσεων, όπως για παράδειγμα η κλάση Στατιστικών (Statistics) είναι υποκλάση της κλάσης Μετρικής (Metric). Περισσότερες πληροφορίες για τις κλάσεις, τις υποκλάσεις και τις σχέσεις που μπορούν να έχουν αυτές μεταξύ τους υπάρχουν στο προηγούμενο κεφάλαιο, που αναλύονται τα χαρακτηριστικά της γλώσσας OWL [5] (§2.2.4) στην οποία έχει στηριχθεί, όπως έχει ήδη αναφερθεί, η ανάπτυξη της οντολογίας Γλώσσα. Αξίζει να επισημανθεί ωστόσο, ότι για τη σχέση μεταξύ κλάσεων χρησιμοποιούνται ιδιότητες αντικειμένων (object properties), ενώ όταν μια κλάση έχει πεδίο τιμών που δεν περιλαμβάνει κλάση τότε χρησιμοποιούνται ιδιότητες τύπου δεδομένων (datatype properties).

Οι κλάσεις που απαρτίζουν την οντολογία Γλώσσα είναι:

- QoS Παράμετρος (QoS Parameter): Η κλάση αυτή αποτελεί το γενικότερο όρο με τον οποίο μοντελοποιείται μια μη – λειτουργική παράμετρος μιας διαδικτυακής υπηρεσίας (§3.1). Αυτές οι παράμετροι δύναται να είναι μετρήσιμα μεγέθη ή να έχουν σχέσεις με άλλες μη – λειτουργικές παραμέτρους, όπως θα φανεί και στη συνέχεια.
- Μετρική (Metric): Η κλάση αυτή χρησιμοποιείται για να εκχωρήσει σε μια παράμετρο ποιότητας υπηρεσίας τιμή και μονάδες. Συσχετίζεται με την κλάση QoS Παράμετρο με την ιδιότητα αντικειμένου (object property) hasMetric. Όπως φαίνεται και στο σχήμα που ακολουθεί (Σχήμα 3-2), το κάθε αντικείμενο της κλάσης Μετρικής έχει: (i) μια ιδιότητα τύπου δεδομένων (MetricType) που χρησιμοποιείται για να καθορίσει τον τύπο της, όπως για παράδειγμα Ακέραιος (Int), Συμβολοακολουθία (String), Λογικός (Boolean) κτλ., (ii) μια ιδιότητα τύπου δεδομένων (Value) που χρησιμοποιείται για να αποθηκεύσει την τιμή της μεταβλητής αυτής και (iii) μια ιδιότητα αντικειμένου (hasUnit), η οποία χρησιμοποιείται για να συσχετίσει την κλάση Μετρική με την κλάση Μονάδα (Unit). Ο ρόλος της κλάσης Μονάδας είναι διττός: Αφενός να παρέχει τη δυνατότητα καταγραφής της μονάδας της εκάστοτε παραμέτρου ποιότητας υπηρεσίας (π.χ. cm) και αφετέρου να προσφέρει τη δυνατότητα μετατροπής από ένα σύστημα μονάδων σε κάποιο άλλο (π.χ. από cm σε m), το οποίο επιτυγχάνεται με τη χρήση της βοηθητικής κλάσης Τύπος Μετατροπής (ConversionFormula). Όλες αυτές οι κλάσεις συνδέονται μεταξύ τους με τις αντίστοιχες ιδιότητες αντικειμένου convertsTo και hasConversionFormula όπως σκιαγραφείται και στο Σχήμα 3-2. Αξίζει να επισημανθεί, ότι εφόσον μια παράμετρος δύναται να είναι μη μετρήσιμο μέγεθος (όπως για παράδειγμα η διαθεσιμότητα), είναι λογικό επακόλουθο σε αυτές τις περιπτώσεις να μη γίνεται χρήση της κλάσης Μονάδας (Unit). Η κλάση Μετρική υποστηρίζει ακόμη τη δυνατότητα διατήρησης στατιστικών στοιχείων για μια παράμετρο, με τη χρήση της

υποκλάσης Στατιστικών (Statistics) η οποία δύναται να υποστηρίζει πλήθος στατιστικών συναρτήσεων.

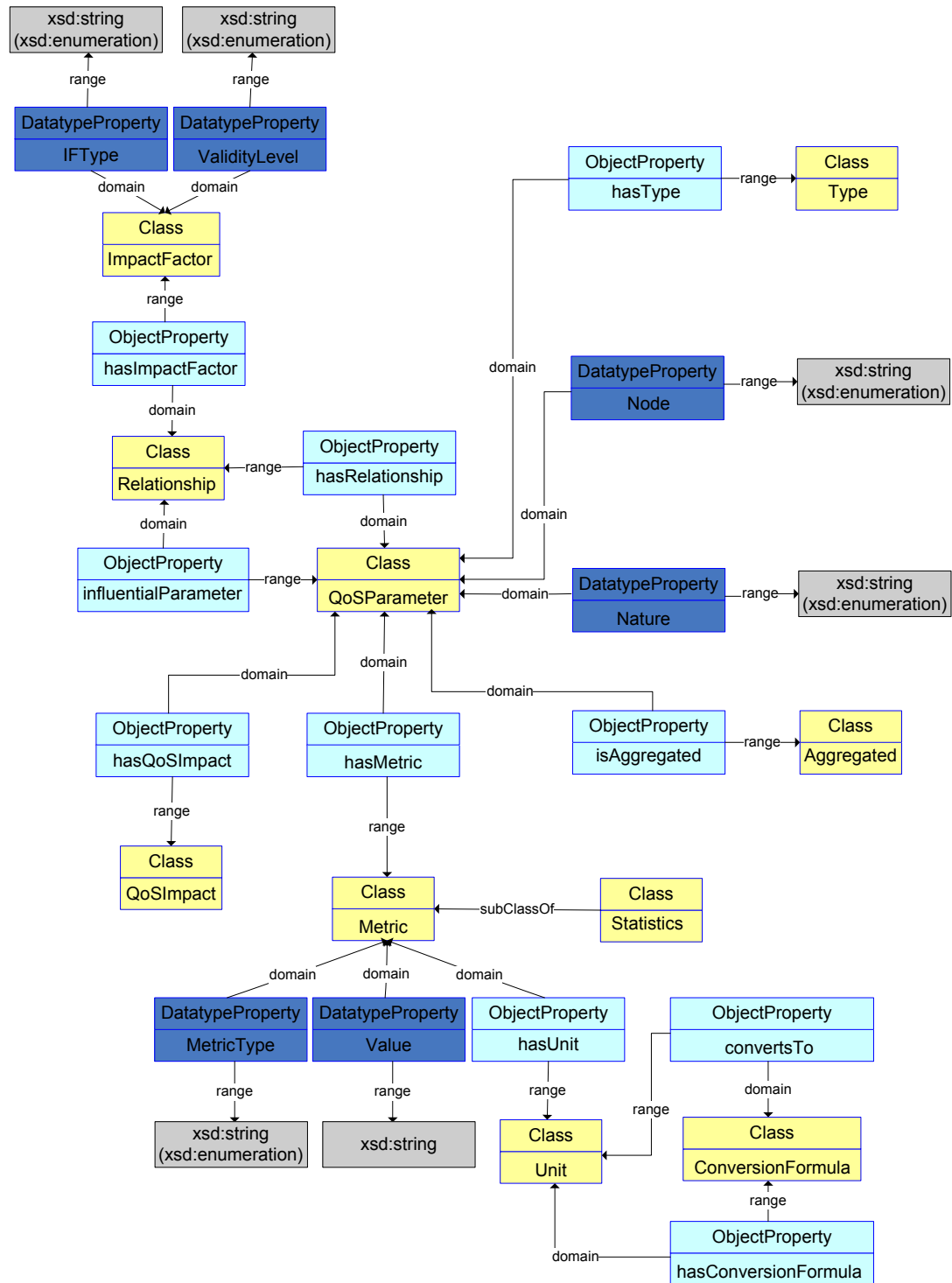
- QoS Επίδραση (QoSImpact): Η κλάση QoS Επίδραση χρησιμοποιείται για να αναπαραστήσει το βαθμό επίδρασης της τιμής μιας παραμέτρου ποιότητας υπηρεσίας στη συνολική ποιότητα μιας διαδικτυακής υπηρεσίας, όπως αυτή γίνεται αντιληπτή από το χρήστη. Χαρακτηριστικό παράδειγμα με το οποίο καθίσταται αυτό σαφές αποτελεί η περίπτωση της μείωσης του χρόνου εξυπηρέτησης μιας αίτησης. Η μείωση αυτή είναι προφανές ότι αναμένεται να αυξήσει το βαθμό χρήσης της υπηρεσίας αυτής από το χρήστη. Επομένως, η κλάση αυτή προσφέρει τη δυνατότητα εκτίμησης του βαθμού ικανοποίησης του χρήστη υπό το πρίσμα της ποιότητας της υπηρεσίας που απολαμβάνει. Κατά συνέπεια, ο ρόλος της είναι πολύ σημαντικός και ξεχωριστός, αφού στοχεύει στη μοντελοποίηση του βαθμού αντίληψης του χρήστη για την ποιότητα της υπηρεσίας που του προσφέρεται. Παράμετρος που απαντάται και σε εργασίες της παγκόσμιας βιβλιογραφίας και αναλύθηκαν προηγουμένως (§3.2) όπως στην [73].
- Τύπος (Type): Η κλάση αυτή παριστάνει τη συγκεκριμένη κλάση / κατηγορία της οντολογίας Λεξιλόγιο (§3.3.3) όπου μια παράμετρος ποιότητας υπηρεσίας ανήκει, όπως για παράδειγμα ασφάλεια, επίδοση, κ.α. (περισσότερες πληροφορίες για τις παραμέτρους και τις κατηγορίες αυτές υπάρχουν στην επόμενη παράγραφο). Η κλάση αυτή συσχετίζεται με την κλάση QoS Παράμετρο με την ιδιότητα αντικειμένου hasType, με πεδίο ορισμού (domain) την κλάση QoS Παράμετρο και πεδίο τιμών (range) την κλάση Τύπος.
- Φύση (Nature): Αυτή η ιδιότητα τύπου δεδομένων (datatype property) χρησιμοποιείται για να αναπαραστήσει τη δυναμική ή στατική «φύση» μιας παραμέτρου ποιότητας υπηρεσίας. Στατική χαρακτηρίζεται η «φύση» μιας παραμέτρου, όταν έχει καθοριστεί η τιμή της εκ των προτέρων και παραμένει αμετάβλητη καθ' όλη τη διάρκεια που διατίθεται η υπηρεσία. Ένα τέτοιο παράδειγμα αποτελεί το είδος των πρωτοκόλλων ασφαλείας ή των αλγορίθμων κρυπτογράφησης που υποστηρίζονται και χρησιμοποιούνται κατά την ανταλλαγή των δεδομένων μεταξύ παρόχου και χρήστη της υπηρεσίας. Αντίθετα, δυναμική χαρακτηρίζεται η «φύση» μιας παραμέτρου, όταν οι τιμές της δύναται να μεταβάλλονται κατά τη διάρκεια εκτέλεσης / διάθεσης της υπηρεσίας. Παράδειγμα δυναμικής παραμέτρου είναι ο χρόνος απόκρισης της υπηρεσίας (περισσότερες πληροφορίες για τις παραμέτρους αυτές υπάρχουν στην επόμενη παράγραφο). Επισημαίνεται ότι οι τιμές του τύπου δεδομένων Φύση καθορίζονται από τον πάροχο της υπηρεσίας, ο οποίος είναι υπεύθυνος και για την τακτική ενημέρωση των τιμών της, ώστε η πληροφόρηση

του χρήστη της υπηρεσίας που διαθέτει τέτοιες παραμέτρους να είναι σε απόλυτο βαθμό έγκυρη. Ακόμη, όπως φαίνεται στο σχήμα της οντολογίας Γλώσσας που ακολουθεί (Σχήμα 3-2), οι τιμές του τύπου αυτού είναι απαρίθμηση συμβολοακολουθιών (xsd: string - xsd:enumeration) με πεδίο τιμών «Δυναμική» και «Στατική».

- Αθροιστική (Aggregated): Η κλάση αυτή χρησιμοποιείται όταν μια παράμετρος ποιότητας υπηρεσίας είναι το αποτέλεσμα της σύνθεσης / άθροισης δύο ή περισσότερων παραμέτρων. Χαρακτηριστικό παράδειγμα αποτελεί η περίπτωση του χρόνου απόκρισης της υπηρεσίας που είναι το αποτέλεσμα της άθροισης της καθυστέρησης μετάδοσης της απόκρισης που οφείλεται σε δικτυακούς παράγοντες και του χρόνου που απαιτείται για την επεξεργασία της αίτησης από τον εξυπηρετητή (περισσότερες πληροφορίες για τις παραμέτρους αυτές υπάρχουν στην επόμενη παράγραφο).
- Κόμβος (Node): Αυτή η ιδιότητα τύπου δεδομένων χρησιμοποιείται για να αναγνωρίσει και να διαχωρίσει το είδος του δικτυακού κόμβου, ο οποίος μπορεί να επιδράσει στην τιμή μιας παραμέτρου ποιότητας υπηρεσίας. Επομένως, οι πιθανές τιμές αυτού του τύπου δεδομένων είναι: «Εξυπηρετητής» (Server), «Πελάτης» (Client) ή και τα δύο, ενώ είναι προφανές ότι ο τύπος των δεδομένων αυτών είναι απαρίθμηση συμβολοακολουθιών (xsd: string - xsd:enumeration).
- Συγγενική (Relationship): Η εισαγωγή της κλάσης αυτής στην οντολογία Γλώσσα στοχεύει στο να συσχετίσει μια παράμετρο ποιότητας υπηρεσίας με μια άλλη. Όπως σκιαγραφείται και στο σχήμα που ακολουθεί (Σχήμα 3-2), η κλάση Συγγενική συσχετίζεται με την κλάση QoS Παράμετρο με την ιδιότητα αντικειμένου hasRelationship. Επιπρόσθετα, προκειμένου να μοντελοποιηθεί η περίπτωση της αλληλοσυσχέτισης 2 παραμέτρων ποιότητας υπηρεσίας με γνώμονα ποια από τις δύο παραμέτρους έχει επίδραση στην άλλη, κρίθηκε αναγκαίο η εισαγωγή της ιδιότητας αντικειμένου influentialParameter. Άλλωστε με την προσέγγιση αυτή και τη χρήση αυτής της ιδιότητας αντικειμένου, μοντελοποιούνται και οι περιπτώσεις εκείνες όπου οι σχέσεις αλληλεξάρτησης μεταξύ των 2 παραμέτρων είναι ασύμμετρες. Χαρακτηριστικό παράδειγμα μιας τέτοιας περίπτωσης είναι η σχέση μεταξύ της διαθεσιμότητας και της προσπελασιμότητας ενός παρόχου υπηρεσιών, αφού όσο μεγαλύτερη προσπελασιμότητα έχει ο πάροχος, τόσο μεγαλύτερη διαθεσιμότητα έχει, χωρίς να ισχύει απαραίτητα και το αντίστροφο, μια και η προσπελασιμότητα εξαρτάται και από άλλους παράγοντες, όπως για παράδειγμα τη χωρητικότητα του παρόχου (περισσότερες πληροφορίες για τις παραμέτρους αυτές υπάρχουν στην επόμενη παράγραφο). Εξάλλου, η σχέση μεταξύ δύο παραμέτρων ποιότητας

υπηρεσίας δύναται να είναι είτε ανάλογη (Proportional) είτε αντιστρόφως ανάλογη (Inversely Proportional), κάτι που μοντελοποιείται με τη χρήση της ιδιότητας τύπου δεδομένων IFType. Ένα παράδειγμα ανάλογης σχέσης μεταξύ 2 παραμέτρων είναι η σχέση μεταξύ προσπελασιμότητας και χωρητικότητας, αφού όσο αυξάνεται η χωρητικότητα ενός συστήματος τόσο αυξάνεται και η προσπελασιμότητά του. Ένα παράδειγμα αντιστρόφως ανάλογης σχέσης είναι μεταξύ του χρόνου απόκρισης μιας υπηρεσίας και του ρυθμού δεδομένων (throughput), αφού όσο μεγαλώνει ο ρυθμός μετάδοσης των δεδομένων μεταξύ παρόχου και χρήστη της υπηρεσίας τόσο ελαττώνεται ο χρόνος απόκρισης της υπηρεσίας. Επομένως, οι τιμές του τύπου αυτού είναι απαρίθμηση συμβολοακολουθιών (xsd: string - xsd:enumeration) με πεδίο τιμών «Ανάλογη» και «Αντιστρόφως Ανάλογη». Ακόμη, η σχέση μεταξύ 2 παραμέτρων δύναται να είναι «Ισχυρή», «Μέτρια» και «Αδύναμη», που εκφράζει την «ένταση» της επίδρασης της μιας παραμέτρου στην άλλη και μοντελοποιείται με την εισαγωγή της ιδιότητας τύπου δεδομένων ValidityLevel με πεδίο τιμών τις τιμές που μόλις αναφέρθηκαν και τύπο απαρίθμησης συμβολοακολουθιών (xsd: string - xsd:enumeration). Η εισαγωγή των δύο αυτών τύπων δεδομένων (IFType και ValidityLevel) στην οντολογία Γλώσσα επιτυγχάνεται με τη χρήση της κλάσης Παράγοντας Επίδρασης (ImpactFactor), η οποία συσχετίζεται με τη κλάση Συγγενική (Relationship) μέσω της ιδιότητας αντικειμένου hasImpactFactor.

Η γραφική αναπαράσταση των κλάσεων που αναφέρθηκαν καθώς και των μεταξύ τους σχέσεων σκιαγραφείται στο ακόλουθο σχήμα:



Σχήμα 3-2 Οντολογία Γλώσσα

3.3.3. Οντολογία Λεξιλόγιο

Στην παράγραφο αυτή θα αναλυθεί η οντολογία Λεξιλόγιο [16]. Η οντολογία αυτή περιλαμβάνει τις κλάσεις των παραμέτρων ποιότητας υπηρεσίας που συναντώνται πιο συχνά σε διαδικτυακές υπηρεσίες, ενώ παράλληλα προσφέρει έναν τρόπο ιεράρχησής τους. Η ανάλυση των παραμέτρων θα στηριχθεί κυρίως στις εργασίες [65], [69] και [77]. Στη συνέχεια ακολουθεί ένας συγκεντρωτικός πίνακας, που περιλαμβάνει τις μονάδες της κάθε παραμέτρου (π.χ. μονάδα χρόνου), το πεδίο τιμών της (π.χ. μεγαλύτερη του 0) και το είδος των τιμών της (π.χ. ακέραια), ενώ η ανάλυση ολοκληρώνεται με την παράθεση του διαγράμματος που αναπαριστά την οντολογία Λεξιλόγιο (Σχήμα 3-3).

Οι κλάσεις που απαρτίζουν την οντολογία Λεξιλόγιο είναι:

- Κλιμάκωση (scalability): Η κλάση αυτή χρησιμοποιείται για να περιγράψει την ικανότητα του παρόχου / εξυπηρετητή να αυξήσει τη χωρητικότητά του, ώστε να μπορεί να εξυπηρετήσει περισσότερες αιτήσεις. Ακόμη, κλιμακωτό θεωρείται το σύστημα το οποίο μπορεί να επεξεργαστεί περισσότερες αιτήσεις, λειτουργίες ή συναλλαγές (transactions) όταν καθίσταται ανάγκη. Το πεδίο τιμών της είναι μεταξύ 0 και 1.
- Διαθεσιμότητα (availability): Διαθεσιμότητα ορίζεται σαν η πιθανότητα ο πάροχος / εξυπηρετητής της υπηρεσίας να είναι ενεργός και να μπορεί να εξυπηρετήσει νέες αιτήσεις. Οι τιμές που παίρνει η κλάση αυτή κυμαίνονται από 0 μέχρι 1. Όσο πιο πολύ προσεγγίζει την μονάδα, τόσο πιο διαθέσιμος θεωρείται ο πάροχος, κάτι που είναι προφανώς επιθυμητό.
- Προσπελασιμότητα (accessibility): Προσπελασιμότητα ορίζεται ως η δυνατότητα του παρόχου / εξυπηρετητή να εξυπηρετεί νέες αιτήσεις. Η προσπελασιμότητα διαφέρει από τη διαθεσιμότητα διότι ένας εξυπηρετητής μπορεί να είναι διαθέσιμος, που σημαίνει ότι είναι ενεργός, αλλά μπορεί να μην είναι προσπελάσιμος, που σημαίνει ότι δε μπορεί να εξυπηρετήσει νέες αιτήσεις. Αυτό είναι ένα φαινόμενο που προκαλείται όταν ένας εξυπηρετητής δέχεται μεγάλο αριθμό αιτήσεων και δεν παρουσιάζει κλιμάκωση το σύστημά του. Επομένως, για να παρουσιάζει ένας εξυπηρετητής μεγάλη προσπελασιμότητα, θα πρέπει να μελετηθεί η κλιμάκωση που προσφέρει το σύστημά του. Όπως και στη διαθεσιμότητα, οι τιμές της κλάσεως αυτής είναι μεταξύ 0 και 1.
- Χωρητικότητα (capacity): Η χωρητικότητα καθορίζει το μέγιστο αριθμό ταυτόχρονων αιτήσεων που ένας πάροχος / εξυπηρετητής μπορεί να διαχειριστεί παράλληλα. Είναι προφανές ότι όσο μεγαλύτερη χωρητικότητα διαθέτει ένας εξυπηρετητής, τόσες περισσότερες αιτήσεις μπορεί να εξυπηρετήσει. Επομένως, όσο

μεγαλύτερη χωρητικότητα έχει ένας εξυπηρετητής, τόσο μεγαλύτερη προσπελασιμότητα έχει. Επειδή η χωρητικότητα εκφράζει αριθμό αιτήσεων, το πεδίο τιμών της είναι φυσικός αριθμός μεγαλύτερος της μονάδας.

Αξιοπιστία (reliability):

Αξιοπιστία ορίζεται σαν η πιθανότητα η κλήση και εκτέλεση μιας υπηρεσίας να ολοκληρωθεί με επιτυχία. Είναι προφανές επομένως ότι η αξιοπιστία συνδέεται στενά με τη διαθεσιμότητα, αφού όσο πιο διαθέσιμος είναι ένας εξυπηρετητής / πάροχος, τόσο πιο αξιόπιστος είναι. Μεγέθη με τα οποία υπολογίζεται η αξιοπιστία ενός παρόχου είναι:

- Μέση τιμή επαναφοράς MTTR (Mean Time To Repair): Το MTTR ορίζεται σαν το μέσο χρόνο που απαιτείται από τον εξυπηρετητή για να μπορεί να λειτουργήσει πάλι κανονικά, αφού μεσολαβήσει διάστημα προσωρινής διακοπής του. Η κλάση αυτή παίρνει τιμές μεταξύ 0 και 1.
- Μέση τιμή μεταξύ αποτυχιών MTBF (Mean Time Between Failures): Το MTBF ορίζεται σαν το χρονικό διάστημα που μεσολαβεί μεταξύ δυο προσωρινών διακοπών του εξυπηρετητή, ενώ το πεδίο τιμών της είναι μεταξύ 0 και 1.

Επίδοση (performance)

Η επίδοση περιλαμβάνει τις κλάσεις εκείνες που παρέχουν πληροφορίες για τη δικτυακή σύνδεση μεταξύ παρόχου και πελάτη, δίνοντας τη δυνατότητα μεταξύ άλλων της παρακολούθησης και καταγραφής του χρονικού διαστήματος που απαιτείται για την εξυπηρέτηση μιας αίτησης. Οι κλάσεις αυτές είναι:

- Ρυθμός δεδομένων (throughput): Ο ρυθμός δεδομένων παριστάνει τον αριθμό των αιτήσεων που εξυπηρετήθηκαν από τον πάροχο σε ένα δεδομένο χρονικό διάστημα. Συνήθως οι χρήστες / πελάτες ενδιαφέρονται για υψηλές τιμές της κλάσεως αυτής, ειδικά όταν χρησιμοποιούν υπηρεσία που στέλνει / λαμβάνει μεγάλο όγκο δεδομένων. Επομένως, όταν ένας χρήστης χρησιμοποιεί μια εφαρμογή που στηρίζεται σε τέτοιες υπηρεσίες, τότε καθίσταται απαραίτητο να υπάρχουν υψηλές τιμές στο ρυθμό δεδομένων. Εξάλλου, μετρήσεις που στοχεύουν στον υπολογισμό της συνολικής ποιότητας μιας υπηρεσίας συνήθως περιλαμβάνουν το μέγιστο ρυθμό δεδομένων ή μεθόδους που υπολογίζουν πώς μεταβάλλεται ο ρυθμός ως συνάρτηση της πυκνότητας φορτίου. Αυτό δηλώνει ότι η κλάση αυτή είναι ένας δυναμικός όρος και εξαρτάται τόσο από τις γενικές παραμέτρους δικτύου όσο και από την κατάσταση που επικρατεί τη δεδομένη χρονική στιγμή. Μεγάλος ρυθμός δεδομένων συνήθως συνοδεύεται και από μικρή συνολικά πυκνότητα κίνησης, όπως και το αντίστροφο

βεβαίως. Ένα από τα σημαντικότερα θέματα στην καταγραφή των τιμών της είναι το πότε και σε ποιο σημείο πρέπει να γίνεται αυτό, προκειμένου οι μετρήσεις να είναι όσο το δυνατόν πιο αντικειμενικές. Το πεδίο τιμών της είναι πραγματικός αριθμός, μεγαλύτερος του μηδενός.

- Καθυστέρηση (delay): Ως καθυστέρηση ορίζεται το χρονικό διάστημα που μεσολαβεί από τη στιγμή που αποστέλλεται ένα πακέτο από τον πελάτη / χρήστη μέχρι τη στιγμή που αυτό φτάνει στον πάροχο, αλλά και το αντίστροφο, δηλαδή από τον πάροχο προς τον πελάτη / χρήστη. Για την πλειονότητα των υπηρεσιών, η καθυστέρηση δε θα πρέπει να υπερβαίνει κάποια συγκεκριμένα όρια. Όπως και στο ρυθμό δεδομένων, η καθυστέρηση που εμφανίζεται στον πελάτη / χρήστη και η καθυστέρηση που εμφανίζεται στον πάροχο μπορεί να είναι διαφορετική. Ο συνολικός χρόνος καθυστέρησης (round-trip time) είναι το άθροισμα αυτών των επιμέρους καθυστερήσεων και είναι ένα μέγεθος αρκετά σημαντικό, αφού οι περισσότερες επικοινωνίες στις διαδικτυακές υπηρεσίες είναι του τύπου αίτηση / απόκριση. Επιπρόσθετα, όπως και στο ρυθμό δεδομένων, επειδή η καθυστέρηση είναι ένα μέγεθος μεταβλητό με το χρόνο, είναι πολύ δύσκολο να μετρηθεί με αντικειμενικό τρόπο. Το πεδίο τιμών της είναι πραγματικός αριθμός, μεγαλύτερος του μηδενός.
- Χρόνος απόκρισης (response time): Ο χρόνος απόκρισης είναι ο συνολικός χρόνος που απαιτείται για να εκτελέσει ένας πελάτης μια υπηρεσία. Ο χρόνος αυτός υπολογίζεται από τη στιγμή που ξεκινά η επικοινωνία του πελάτη με τον πάροχο και τερματίζεται όταν ο πελάτης της υπηρεσίας λάβει και το τελευταίο byte της απάντησης. Ωστόσο, ορισμένες φορές ως χρόνος απόκρισης ορίζεται ο «βέβαιος» (guaranteed) χρόνος που απαιτείται για την εξυπηρέτηση μιας αίτησης. Σε αυτή την περίπτωση και ανάλογα με την υλοποίηση, ο χρόνος απόκρισης μπορεί να αναφέρεται στο μέγιστο ή τον ελάχιστο χρόνο που απαιτείται για την εξυπηρέτηση. Το πεδίο τιμών της κλάσεως αυτής είναι πραγματικός αριθμός, μεγαλύτερος του μηδενός.
- Μεταβολή καθυστέρησης (jitter): Η κλάση αυτή αναφέρεται στη μεταβλητότητα της καθυστέρησης και αυξάνεται όσο αυξάνεται η κίνηση / φορτίο στο δίκτυο. Η μεταβολή καθυστέρησης μπορεί να πάρει ιδιαίτερα υψηλές τιμές - κάτι που είναι βέβαια μη επιθυμητό - όταν πακέτα μεταφέρονται μεταξύ διαφορετικών διαδρομών. Το φαινόμενο αυτό παρατηρείται όταν οι υπεύθυνοι δρομολογητές του δικτύου έχουν υπερφορτωθεί. Όπως και στις προηγούμενες κλάσεις (ρυθμό δεδομένων και καθυστέρηση), η τιμή της είναι διαφορετική προς τις δύο κατευθύνσεις (από τον

πελάτη προς τον πάροχο και το αντίστροφο), ενώ το πεδίο τιμών της είναι πραγματικός αριθμός, μεγαλύτερος του μηδενός.

- Ρυθμός λαθών (error rate): Ο ρυθμός λαθών προσδιορίζει τη συχνότητα με την οποία χάνονται πακέτα. Είναι μια δυναμική κλάση που εξαρτάται από τα κατώτερα στρώματα δικτύου, τα χαρακτηριστικά των κόμβων και το φορτίο / κίνηση του δικτύου. Οι τιμές της κυμαίνονται μεταξύ 0 και 1.

Συνολικά επομένως για να εμφανίζει ένα σύστημα «υψηλή» επίδοση (performance), είναι επιθυμητό να υπάρχει μεγάλος ρυθμός δεδομένων, μικρή καθυστέρηση, μικρός χρόνος απόκρισης, μικρή μεταβολή καθυστέρησης και μικρός ρυθμός λαθών.

- Κόστος (cost): Η κλάση αυτή παριστάνει το συνολικό κόστος που προκύπτει από τη χρήση της υπηρεσίας. Αποτελεί καθοριστικό παράγοντα για την επιλογή της υπηρεσίας, καθιστώντας την κατ' αυτόν τον τρόπο μια από τις σημαντικότερες κλάσεις κατά τον υπολογισμό της συνολικής ποιότητας της προσφερόμενης υπηρεσίας. Το κόστος δύναται να μοντελοποιηθεί με διάφορους τρόπους, όπως για παράδειγμα κόστος υπηρεσίας / αίτηση, κόστος υπηρεσίας / όγκο δεδομένων, κόστος υπηρεσίας / μονάδα χρόνου κ.α.. Το πεδίο τιμών της είναι πραγματικός αριθμός, μεγαλύτερος ή ίσος με το μηδέν.

Διαμόρφωση (configuration):

Η διαμόρφωση των υπηρεσιών αφορά κυρίως το ρυθμό που ενημερώνονται οι διεπαφές (interfaces) των υπηρεσιών και τον αριθμό των προτύπων / κανόνων που υποστηρίζει η υπηρεσία. Η διαμόρφωση υπολογίζεται από τις ακόλουθες κλάσεις:

- Σταθερότητα (stability): Η σταθερότητα παριστάνει τη συχνότητα των αλλαγών / μετατροπών όσον αφορά τις διεπαφές της υπηρεσίας. Ακόμη, η σταθερότητα μπορεί να αναφέρεται και στη συχνότητα που αλλάζει η υλοποίηση της υπηρεσίας. Οι τιμές της κυμαίνονται μεταξύ 0 και 1.
- Υποστηριζόμενα πρότυπα (supported standards): Η κλάση αυτή περιλαμβάνει τα πρότυπα τα οποία υποστηρίζει και είναι συμβατή η προσφερόμενη υπηρεσία. Τέτοια πρότυπα συναντώνται σε πληθώρα στις διαδικτυακές υπηρεσίες, όπως για παράδειγμα UDDI [37], WSDL [12], SOAP [11] κ.α.. Επομένως όσα περισσότερα τέτοια πρότυπα υποστηρίζει μια υπηρεσία τόσο περισσότερο «συμβατή» με άλλες υπηρεσίες και εφαρμογές καθίσταται. Οι τιμές της κλάσεως αυτής είναι μια απαρίθμηση από συμβολοακολουθίες (strings) των υποστηριζόμενων προτύπων.

- **Ρυθμιστικότητα (regulatory):** Η ρυθμιστικότητα αναφέρεται στην πιθανότητα που έχει μια υπηρεσία να είναι συμβατή με έναν τυχαίο κανόνα. Οι τιμές της κυμαίνονται μεταξύ 0 και 1.
- **Ακεραιότητα (integrity):** Η ακεραιότητα καθορίζει τη δυνατότητα μιας υπηρεσίας να παρέχει ακεραιότητα στα δεδομένα της κατά την εκτέλεσή της, όταν αυτή συμβαίνει σε πολλαπλά επίπεδα / στάδια. Προκειμένου να είναι αυτό εφικτό, θα πρέπει όλα αυτά τα επίπεδα / στάδια να μοντελοποιούνται σαν μια απλή μονάδα και σαν απλή μονάδα να διαχειρίζονται. Σε περίπτωση που όλα τα επίπεδα εκτελεστούν με επιτυχία, τότε όλες οι αλλαγές του κάθε επιμέρους σταδίου γίνονται μόνιμες, αλλιώς το σύστημα επανέρχεται στην κατάσταση που ήταν αρχικά. Η ακεραιότητα παίρνει τιμές αληθής ή ψευδής, ανάλογα με το αν την υποστηρίζει η προσφερόμενη υπηρεσία ή όχι.

Ασφάλεια (security)

Οι κλάσεις της κατηγορίας αυτής περιγράφουν το βαθμό ασφάλειας της εκάστοτε υπηρεσίας. Όπως αναφέρθηκε και στην παράγραφο 3.1, η ασφάλεια αποτελεί θέμα μείζονος σημασίας για τις διαδικτυακές υπηρεσίες. Οι κλάσεις που καθορίζουν το βαθμό ασφάλειας μιας υπηρεσίας είναι οι ακόλουθες:

- **Εμπιστευτικότητα (confidentiality):** Η παρουσία της κλάσεως αυτής σε μια διαδικτυακή υπηρεσία βεβαιώνει ότι στην υπηρεσία αυτή, μόνο πιστοποιημένοι φορείς έχουν τη δυνατότητα να τροποποιούν τα δεδομένα που ανταλλάσσονται. Επομένως, η κλάση αυτή παίρνει τιμές αληθής ή ψευδής, ανάλογα με το αν υποστηρίζεται από την υπηρεσία ή όχι, πρόκειται δηλαδή για μια λογική (true / false) μεταβλητή.
- **Ελεγχσιμότητα (auditability):** Όταν η κλάση αυτή είναι παρούσα σε μια υπηρεσία, αυτό δηλώνει ότι υπάρχει δυνατότητα πρόσβασης στο ιστορικό της υπηρεσίας, με αποτέλεσμα να είναι εφικτή η παρακολούθηση της συμπεριφοράς της υπηρεσίας κατά την εκτέλεσή της στο παρελθόν. Πρόκειται για μια λογική μεταβλητή επομένως που λαμβάνει τιμές αληθής ή ψευδής. Σε ορισμένες υπηρεσίες παρουσιάζεται η κλάση ανιχνευσιμότητα (traceability) αντί της ελεγχσιμότητας, μοντελοποιώντας όμως ουσιαστικά την ίδια λειτουργικότητα.
- **Πιστοποίηση (authentication):** Η πιστοποίηση αποτελεί έναν από τους πιο σημαντικούς όρους σε μια υπηρεσία, δεδομένου ότι αναφέρεται σε ένα από τα πιο ουσιαστικά θέματα που απαρτίζουν την εκτέλεση μιας υπηρεσίας. Η πιστοποίηση λαμβάνει χώρα αμέσως μετά την κλήση της υπηρεσίας από τον πελάτη / χρήστη και

είναι η διαδικασία κατά την οποία ο πάροχος των υπηρεσιών ζητά από τον πελάτη / χρήστη να πιστοποιήσει την ταυτότητά του. Σε περίπτωση που η πιστοποίηση είναι επιτυχής, ο χρήστης μπορεί να αρχίσει να χρησιμοποιεί την υπηρεσία. Οι μηχανισμοί πιστοποίησης (όπως για παράδειγμα βάσει κωδικού χρήσης, κουπονιών (tokens) κ.α.) που ακολουθούνται από τον πάροχο των υπηρεσιών αποτελεί και το πεδίο τιμών της κλάσεως αυτής. Επομένως, περιλαμβάνει μια απαρίθμηση από συμβολοακολουθίες (strings).

- Εξουσιοδότηση (authorization): Ως εξουσιοδότηση ορίζεται εν γένει η διαδικασία πιστοποίησης ότι ένας χρήστης έχει το δικαίωμα να εκτελέσει μια συγκεκριμένη ενέργεια. Επίσης, ως εξουσιοδότηση ορίζεται και η διαδικασία που αποφασίζει αν ένας χρήστης επιτρέπεται να έχει πρόσβαση σε μια διαδικασία αφού έχει πιστοποιηθεί από το σύστημα. Γενικά οι μηχανισμοί εξουσιοδότησης χρησιμοποιούνται για να πιστοποιήσουν τα δικαιώματα πρόσβασης σε «προστατευμένες» υπηρεσίες. Επομένως, για να χρησιμοποιήσει κάποιος χρήστης τέτοιες υπηρεσίες και να μπορεί να υποβληθεί σε διαδικασία πιστοποίησης, θα πρέπει να διαθέτει τα κατάλληλα πιστοποιητικά. Τα πιστοποιητικά που υποστηρίζονται από τη διαδικτυακή υπηρεσία αποτελούν και σε αυτή την περίπτωση το πεδίο τιμών της, δηλαδή το πεδίο τιμών της αποτελείται από μια απαρίθμηση από συμβολοακολουθίες.
- Κρυπτογράφηση δεδομένων (data encryption): Η κρυπτογράφηση των δεδομένων αναφέρεται στις μεθόδους που η υπηρεσία κρυπτογραφεί τα δεδομένα κατά την εκτέλεσή της, δηλαδή στους αλγόριθμους κρυπτογράφησης που έχουν χρησιμοποιηθεί. Διαφοροποιείται σε σχέση με τις δύο προηγούμενες κλάσεις στο γεγονός ότι αναφέρεται στη χρονική στιγμή όπου η υπηρεσία χρησιμοποιείται / εκτελείται και όχι στα στάδια που μεσολαβούν προτού ξεκινήσει η εκτέλεσή της. Η διαδικασία της κρυπτογράφησης των δεδομένων αρχικοποιείται συνήθως από το χρήστη / πελάτη, ο οποίος αποστέλλει μια αίτηση στον πάροχο της υπηρεσίας αιτούμενος για κρυπτογραφημένη επικοινωνία καθορίζοντας παράλληλα τον επιθυμητό αλγόριθμο κρυπτογράφησης. Ο πάροχος με τη σειρά του αποκρίνεται στο αίτημα του πελάτη του είτε καταφατικά, σε περίπτωση που υποστηρίζει το συγκεκριμένο αλγόριθμο, είτε αρνητικά σε περίπτωση που δεν υποστηρίζει το συγκεκριμένο αλγόριθμο ή δεν υποστηρίζει κρυπτογραφημένη επικοινωνία γενικότερα. Πεδίο τιμών της κλάσεως αυτής είναι μια απαρίθμηση από συμβολοακολουθίες που παρουσιάζουν τους αλγόριθμους κρυπτογράφησης που υποστηρίζονται από τον πάροχο.

- Μη αποκήρυξη (non repudiation): Τα υπολογιστικά συστήματα θα πρέπει να διαθέτουν έναν τρόπο να αποτρέπουν τη δυνατότητα της αποκήρυξης μιας συναλλαγής (transaction), η οποία ολοκληρώθηκε με επιτυχία, από τη φυσική οντότητα που την αιτήθηκε. Προκειμένου να διασφαλιστεί η «ψηφιακή» εμπιστοσύνη (digital trust), όσοι χρησιμοποιούν τέτοια συστήματα οφείλουν να διαθέτουν ψηφιακές υπογραφές (digital signatures), οι οποίες όχι μόνο πιστοποιούν την ταυτότητα του χρήστη αλλά παράλληλα δημιουργούν χρονικά στιγμιότυπα των συναλλαγών του χρήστη με το σύστημα, ώστε να μη μπορεί να υποστηρίξει μετά το πέρας της συναλλαγής ότι η συναλλαγή δεν είναι νόμιμη ή να αποποιηθεί τη νομιμότητα της πιστοποίησής του. Η κλάση αυτή είναι είτε αληθής σε περίπτωση που υποστηρίζεται από την υπηρεσία είτε ψευδής σε περίπτωση που δεν υποστηρίζεται. Επομένως πρόκειται για μια λογική μεταβλητή.

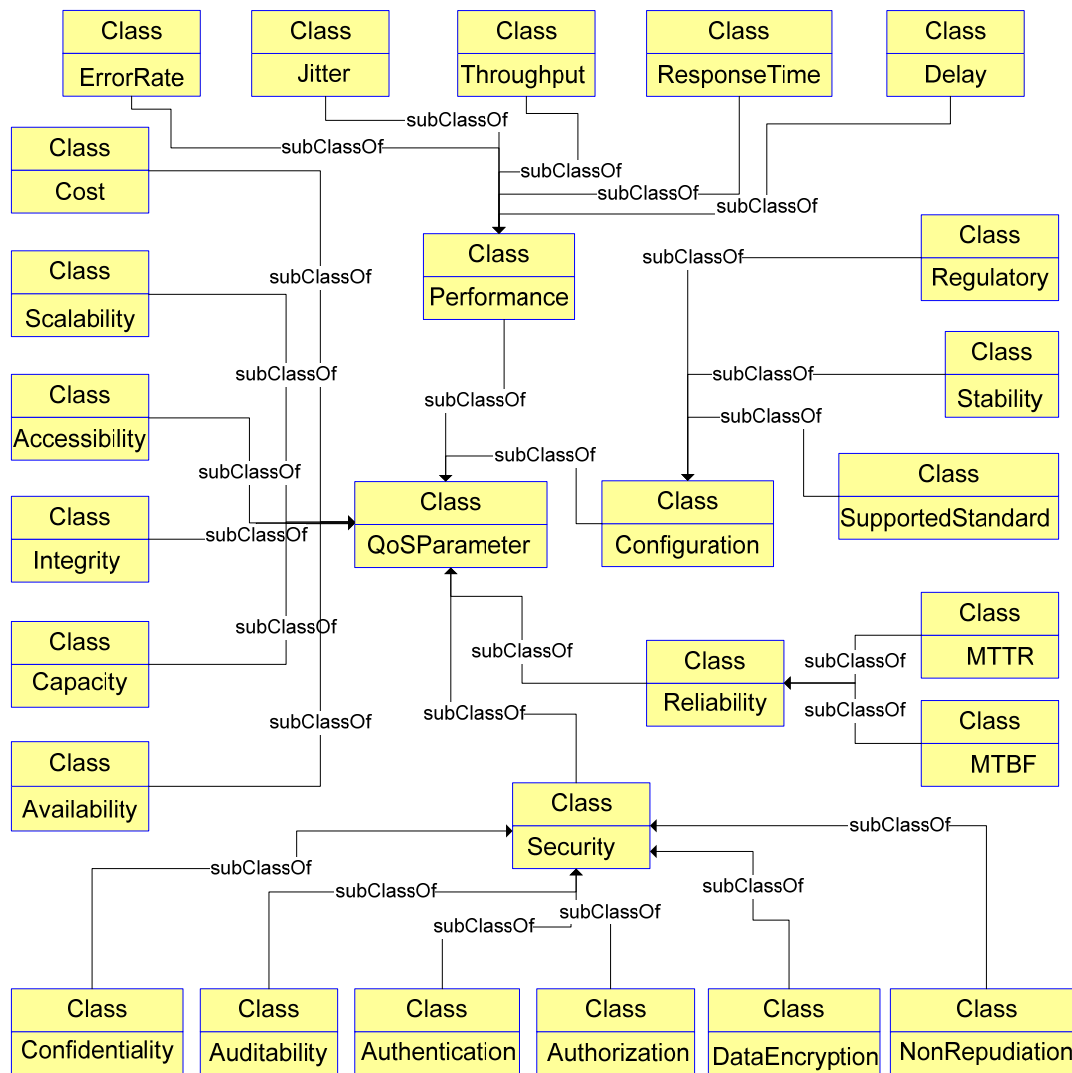
Ο πίνακας που ακολουθεί περιέχει συγκεντρωτικά όλες αυτές τις κλάσεις, συνοδευόμενες από τον τύπο της μεταβλητής που θα χρησιμοποιηθεί για να τις προσομοιώσει σε ένα πραγματικό σύστημα, τις μονάδες της μεταβλητής και το πεδίο τιμών της. Επισημαίνεται, ότι αν και σε κάποιες κλάσεις δεν έχει λογικό νόημα οι ακραίες τιμές των πεδίων τιμών τους (όπως για παράδειγμα διαθεσιμότητα 0), ωστόσο κρίνεται σκόπιμο να συμπεριληφθούν και οι τιμές αυτές, καλύπτοντας με αυτόν τον τρόπο και τις πιο εξεζητημένες περιπτώσεις που μπορούν να αντιμετωπιστούν.

Κλάση	Τύπος δεδομένων	Μονάδα	Πεδίο τιμών
Κλιμάκωση	Δεκαδικός	-	Πραγματικός $\in [0,1]$
Διαθεσιμότητα	Δεκαδικός	-	Πραγματικός $\in [0,1]$
Προσπελασιμότητα	Δεκαδικός	-	Πραγματικός $\in [0,1]$
Χωρητικότητα	Ακέραιος	-	Ακέραιος > 0
MTTR	Δεκαδικός	Χρόνου	Πραγματικός > 0
MTBF	Δεκαδικός	Χρόνου	Πραγματικός > 0
Ρυθμός δεδομένων	Δεκαδικός	Εύρους ζώνης	Πραγματικός > 0
Καθυστέρηση	Δεκαδικός	Χρόνου	Πραγματικός > 0
Χρόνος απόκρισης	Δεκαδικός	Χρόνου	Πραγματικός > 0
Μεταβολή καθυστέρησης	Δεκαδικός	Χρόνου	Πραγματικός > 0
Ρυθμός λαθών	Δεκαδικός	Χρόνου	Πραγματικός $\in [0,1]$
Κόστος	Δεκαδικός	Χρήματος	Πραγματικός ≥ 0
Σταθερότητα	Δεκαδικός	-	Πραγματικός $\in [0,1]$

Υποστηριζόμενα πρότυπα	Συμβολοακολουθία	-	Απαρίθμηση συμβολοακολουθιών
Ρυθμιστικότητα	Δεκαδικός	-	Πραγματικός $\in [0,1]$
Ακεραιότητα	Λογικός	-	Αληθής / Ψευδής
Εμπιστευτικότητα	Λογικός	-	Αληθής / Ψευδής
Ελεγχιμότητα	Λογικός	-	Αληθής / Ψευδής
Πιστοποίηση	Συμβολοακολουθία	-	Απαρίθμηση συμβολοακολουθιών
Εξουσιοδότηση	Συμβολοακολουθία	-	Απαρίθμηση συμβολοακολουθιών
Κρυπτογράφηση δεδομένων	Συμβολοακολουθία	-	Απαρίθμηση συμβολοακολουθιών
Μη αποκήρυξη	Λογικός	-	Αληθής / Ψευδής

Πίνακας 3-1 Μονάδες μέτρησης παραμέτρων ποιότητας

Η γραφική αναπαράσταση των όσων αναφέρθηκαν σκιαγραφείται στο σχήμα που ακολουθεί, που αποτελεί την οντολογία Λεξιλόγιο. Στην οντολογία αυτή οι περισσότεροι όροι που αναλύθηκαν εκτενώς προηγουμένως, μοντελοποιούνται σαν υποκλάσεις της γενικής κλάσης QoS Παράμετρος (QoS parameter), που παριστάνει το γενικότερο όρο με τον οποίο μπορεί να αναφερθεί κάποιος σε μια παράμετρο ποιότητας υπηρεσίας και που βρίσκεται σε απόλυτη ταύτιση με την αντίστοιχη κλάση της οντολογίας Γλώσσας (§3.3.2). Επιπρόσθετα, κάποιες κλάσεις είναι υποκλάσεις κάποιων άλλων όρων, όπως για παράδειγμα η κρυπτογράφηση δεδομένων (Data Encryption) είναι υποκλάση της κλάσης ασφάλειας (Security). Περισσότερες πληροφορίες για τις κλάσεις, τις υποκλάσεις και τις σχέσεις που μπορούν να έχουν αυτές μεταξύ τους υπάρχουν στο προηγούμενο κεφάλαιο που αναλύονται τα χαρακτηριστικά της γλώσσας OWL [5] (§2.2.4) στην οποία έχει στηριχθεί, όπως έχει ήδη αναφερθεί, η ανάπτυξη της οντολογίας Λεξιλόγιο.



Σχήμα 3-3 Οντολογία Λεξιλόγιο

3.3.4. Αποτίμηση οντολογιών

Αντικείμενο της παραγράφου αυτής αποτελεί η αποτίμηση του σχεδιασμού των οντολογιών που παρουσιάστηκαν στο κεφάλαιο αυτό. Επισημαίνεται ότι η εύρεση ολοκληρωμένων μεθόδων αποτίμησης των εν λόγω οντολογιών αλλά και των οντολογιών γενικότερα είναι μια πολύ δύσκολη και επίπονη διαδικασία. Αυτό οφείλεται στο γεγονός ότι οι οντολογίες διαθέτουν πολλαπλά και ιδιαίτερα χαρακτηριστικά όπως έχει ήδη αναφερθεί (§2.2.4), με αποτέλεσμα να μην είναι εφικτή η δημιουργία κριτηρίων που χρήζουν καθολικής αποδοχής και αποτιμούν με ακρίβεια οποιαδήποτε οντολογία. Πρόσθετος παράγοντας που επιβαρύνει την ενέργεια αυτή είναι ότι στην παρούσα φάση έχουν γίνει μόνο σποραδικές προσπάθειες δημιουργίας - σχεδιασμού οντολογιών και επομένως η δημιουργία σαφών και επαρκών κριτηρίων αποτίμησής τους να αποτελεί θέμα ελάχιστονης σημασίας. Ωστόσο μια από τις πιο δημοφιλής μεθόδους αποτίμησης οντολογιών είναι η μέθοδος του χρυσού προτύπου (Golden Standard) [78] κατά την οποία συγκρίνεται η οντολογία με κάποια

οντολογία πρότυπο. Δυστυχώς, αυτή η μέθοδος δε μπορεί να υιοθετηθεί και στην παρούσα περίπτωση μια και μέχρι τη στιγμή της συγγραφής της παρούσας διατριβής δεν υπάρχουν διαθέσιμα τέτοια πρότυπα. Μια άλλη εξίσου δημοφιλής μέθοδος είναι η χρήση της οντολογίας σε κάποια εφαρμογή, όπου η ορθότητα του σχεδιασμού της οντολογίας κρίνεται βάσει των αποτελεσμάτων (σε θέματα επίδοσης, προσαρμοστικότητας κτλ.) που έχει η χρήση της στην εφαρμογή αυτή. Κάτι τέτοιο όμως είναι αρκετά ελλιπές και λανθασμένο, αφού η εξαγωγή συμπερασμάτων θα βασίζεται αποκλειστικά από την εφαρμογή στην οποία θα χρησιμοποιηθεί η εν λόγω οντολογία. Επιπρόσθετα, η μέθοδος αυτή δεν προσφέρει τη δυνατότητα σύγκρισης με άλλες οντολογίες, αφού είναι πολύ δύσκολη η εύρεση αντίστοιχων οντολογιών που θα ικανοποιούν και αυτές τις ανάγκες της συγκεκριμένης εφαρμογής [78]. Για τους λόγους αυτούς κρίθηκε σκόπιμο να μην επιλεγεί κάποια από τις μεθόδους αυτές, αλλά αντίθετα να στηριχθεί η αποτίμηση σε κάποια κριτήρια που να είναι -στο μέγιστο βαθμό που είναι αυτό εφικτό - εφαρμόσιμα, αντικειμενικά και να χρήζουν ευρείας αποδοχής από την επιστημονική κοινότητα.

Τα κριτήρια που θα χρησιμοποιηθούν έχουν δημοσιευθεί στην [79] αποτελώντας εργασία αναφοράς για πολλές άλλες αντίστοιχες εργασίες, και αποτιμούν την ποιότητα μιας οντολογίας βάσει 4 κατηγοριών: συντακτικά, σημασιολογικά, πραγματικά και κοινωνικά. Η κάθε κατηγορία περιλαμβάνει τα κριτήρια τα οποία θα εφαρμοστούν και στις προτεινόμενες οντολογίες. Αναλυτικά τα κριτήρια της κάθε κατηγορίας είναι:

Συντακτικά: Η κατηγορία αυτή αποτιμά την ποιότητα της οντολογίας βάσει των συντακτικών κανόνων που αυτή ακολουθεί. Δύο είναι τα κριτήρια που χρησιμοποιούνται:

- Η τήρηση των συντακτικών κανόνων της γλώσσας στην οποία έχει στηριχθεί η ανάπτυξη της υπό εξέταση οντολογίας (lawfulness): Το κριτήριο αυτό ικανοποιείται σε απόλυτο βαθμό και από τις 2 προτεινόμενες οντολογίες, αφού η υλοποίησή τους έγινε - όπως έχει ήδη αναφερθεί - στο εργαλείο Protégé [76] που προσφέρει αυτόματο έλεγχο συντακτικών λαθών.
- Το ποσοστό χρήσης των χαρακτηριστικών που προσφέρει η γλώσσα στην οποία έχει στηριχθεί η ανάπτυξη της υπό εξέταση οντολογίας (richness): Το κριτήριο αυτό στοχεύει στο να αποτιμήσει το ποσοστό εκμετάλλευσης των χαρακτηριστικών αυτών, με τις οντολογίες που χρησιμοποιούν περισσότερα χαρακτηριστικά να προσφέρουν όπως είναι εύλογο και περισσότερες δυνατότητες στο χρήστη. Οι προτεινόμενες οντολογίες έχουν αναπτυχθεί στη γλώσσα OWL [5] όπως έχει ήδη αναφερθεί (§2.2.4). Όπως φαίνεται και στο Παράρτημα Α που παρατίθεται ο πηγαίος κώδικας των οντολογιών αυτών, οι οντολογίες αυτές χρησιμοποιούν τα σημαντικότερα χαρακτηριστικά της γλώσσας όπως κλάσεις (classes) και υποκλάσεις (subclasses), ιδιότητες τύπου δεδομένων (datatype properties) και αντικειμένων

(object properties), εφαρμόζονται σχέσεις μεταξύ τους τύπου domain / range μεταβλητού αριθμού (cardinality), ενώ δημιουργούνται και διάφορα αντικείμενα (instances). Επομένως μπορεί να υποστηριχθεί με ασφάλεια ότι ο σχεδιασμός των οντολογιών αυτών δε θα περιορίσει το χρήστη τους αλλά αντίθετα θα του προσφέρει πολλαπλές δυνατότητες εκμετάλλευσής τους.

Σημασιολογικά: Η κατηγορία αυτή αποτιμά την ποιότητα της οντολογίας βάσει της σημασίας της έννοιας των όρων που χρησιμοποιούνται σε αυτήν. Τρία είναι τα κριτήρια που χρησιμοποιούνται:

- Η σημασία των λέξεων που χρησιμοποιούνται για τις κλάσεις και τις ιδιότητες από τις οποίες απαρτίζεται η υπό εξέταση οντολογία (interpretability): Ουσιαστικά το κριτήριο αυτό αναφέρεται στο ποσοστό των λέξεων που χρησιμοποιούνται στα ονόματα κλάσεων και ιδιοτήτων και οι οποίες αποτελούν μέρος της αγγλικής γλώσσας. Το κριτήριο αυτό ικανοποιείται σε μεγάλο βαθμό και από τις δύο οντολογίες με λίγες εξαιρέσεις - όπως για παράδειγμα στην οντολογία Γλώσσα η ιδιότητα τύπου δεδομένων *IFTtype* ή οι σύνθετες λέξεις που χρησιμοποιούνται σαν ονόματα όπως η περίπτωση της κλάσης *ImpactFactor* - αφού η ονοματολογία τους στηρίζεται αποκλειστικά σε αγγλικές λέξεις.
- Η συνοχή των όρων (κλάσεις, ιδιότητες) από τους οποίους απαρτίζεται η υπό εξέταση οντολογία (consistency): Περιπτώσεις μη συνοχής παρατηρούνται όταν για παράδειγμα μια κλάση A είναι ταυτόχρονα υποκλάση (subclass) και υπερκλάση (superclass) μιας άλλης κλάσης B ή όταν η ιδιότητα A μιας κλάσης B είναι ταυτόχρονα υποκλάση (subclass) αυτής. Το κριτήριο αυτό ικανοποιείται σε απόλυτο βαθμό και από τις δύο οντολογίες, αφού δεν παρατηρούνται φαινόμενα μη συνοχής.
- Η σαφήνεια των ονομάτων που χρησιμοποιούνται για τις κλάσεις και τις ιδιότητες από τις οποίες απαρτίζεται η υπό εξέταση οντολογία (clarity): Το κριτήριο αυτό είναι επέκταση του πρώτου κριτηρίου της κατηγορίας αυτής (interpretability) και περιλαμβάνει και φράσεις που χρησιμοποιούνται σαν ονόματα στις κλάσεις και τις ιδιότητες αυτές, ενώ επίσης εξετάζει τη χρήση πολύσημων ή μονόσημων λέξεων. Όπως είναι εύλογο, είναι επιθυμητό η χρήση φράσεων στα ονόματα κλάσεων ή ιδιοτήτων να έχουν «νόημα», ενώ οι λέξεις να είναι μονόσημες. Εξετάζοντας τα ονόματα που έχουν χρησιμοποιηθεί στις προτεινόμενες οντολογίες, εύκολα διαπιστώνει κανείς ότι στις περιπτώσεις εκείνες που το όνομα είναι σύνθετο οι λέξεις που το αποτελούν είναι λέξεις της αγγλικής γλώσσας, όπως για παράδειγμα στην οντολογία Γλώσσα η ιδιότητα

τύπου δεδομένων *hasRelationship*. Επιπρόσθετα, οι λέξεις αυτές είναι στην πλειονότητά τους μονόσημες. Κατά συνέπεια και το κριτήριο αυτό ικανοποιείται σε απόλυτο σχεδόν βαθμό από τις προτεινόμενες οντολογίες.

Πραγματικά: Η κατηγορία αυτή αποτιμά την ποιότητα της οντολογίας βάσει της χρησιμότητάς της στους χρήστες / πράκτορες (agents) που θα τη χρησιμοποιήσουν ανεξάρτητα από τη συμβατότητά της με τα κριτήρια σύνταξης ή σημασιολογίας που αναφέρθηκαν προηγουμένως. Τρία είναι τα κριτήρια που χρησιμοποιούνται:

- Το πλήθος των κλάσεων και των ιδιοτήτων από τις οποίες απαρτίζεται η υπό εξέταση οντολογία (comprehensiveness): Ο αριθμός αυτός εξαρτάται σε πολύ μεγάλο βαθμό από την πολυπλοκότητα της επιστημονικής περιοχής που περιγράφει η οντολογία. Σε απόλυτους αριθμούς η οντολογία Γλώσσα περιλαμβάνει 10 κλάσεις και 16 ιδιότητες, ενώ η οντολογία Λεξιλόγιο περιλαμβάνει 27 κλάσεις. Συγκρίνοντας τους αριθμούς αυτούς με αντίστοιχες εργασίες που μελετήθηκαν στην παράγραφο της σχετικής βιβλιογραφίας (§3.2), διαπιστώνεται εύκολα ότι οι οντολογίες αυτές περιλαμβάνουν περισσότερες κλάσεις και ιδιότητες από οποιαδήποτε άλλη οντολογία έχει προταθεί. Επομένως μπορεί να υποστηριχθεί με ασφάλεια ότι η περιγραφή της ποιότητας των διαδικτυακών υπηρεσιών με τις οντολογίες αυτές είναι πληρέστερη από κάθε άλλη αντίστοιχη ερευνητική εργασία.
- Η ακρίβεια και αληθοφάνεια των συμπερασμάτων που εξάγονται μελετώντας τις σχέσεις των όρων (κλάσεις / ιδιότητες) από τις οποίες απαρτίζεται η υπό εξέταση οντολογία (accuracy): Στόχος του κριτηρίου αυτού είναι ο εντοπισμός εκείνων των σχέσεων που οδηγούν σε εσφαλμένα συμπεράσματα. Χαρακτηριστικό παράδειγμα η περίπτωση της σχέσεως μεταξύ τμήματος (department) και μελών προσωπικού (staff). Η δήλωση ότι το μέλος προσωπικού είναι υποκλάση του τμήματος είναι ανακριβής, αφού στην πραγματικότητα είναι μέρος αυτού. Επισημαίνεται ότι ο εντοπισμός αυτός είναι πολύ δύσκολη και επίπονη διαδικασία, ιδιαίτερα σε περιπτώσεις πολύπλοκων οντολογιών που αποτελούνται από πολλούς όρους. Ο σχεδιασμός ωστόσο των προτεινόμενων οντολογιών δεν παρουσιάζει τέτοιες αστοχίες, αφού η μοντελοποίησή τους στηρίχθηκε αποκλειστικά σε όρους και σχέσεις που ισχύουν στην πραγματικότητα.
- Το ποσοστό ωφέλιμης και εκμεταλλεύσιμης πληροφορίας που προσφέρει η υπό εξέταση οντολογία σε κάποια εφαρμογή που τη χρησιμοποιεί (relevance): Οι προτεινόμενες οντολογίες έχουν χρησιμοποιηθεί από διάφορες εφαρμογές στα πλαίσια ενός ερευνητικού προγράμματος [80] εκτενώς και εξαντλητικά. Τα

συμπεράσματα που εξάχθηκαν ήταν σε απόλυτο βαθμό ικανοποιητικά, αφού οι εφαρμογές αυτές έκαναν χρήση των περισσοτέρων όρων (κλάσεις / ιδιότητες) που περιλαμβάνονται στις οντολογίες αυτές, χωρίς να παρατηρηθεί ποτέ το φαινόμενο απουσίας / έλλειψης κάποιου όρου. Κατά συνέπεια και το κριτήριο αυτό ικανοποιείται απόλυτα από τις προτεινόμενες οντολογίες.

Κοινωνικά: Η κατηγορία αυτή αποτιμά την ποιότητα της οντολογίας βάσει του ποσοστού χρήσης της οντολογίας αυτής από άλλες οντολογίες. Δύο είναι τα κριτήρια που χρησιμοποιούνται:

- Ο αριθμός των αναφορών που γίνονται στην υπό εξέταση οντολογία από άλλες οντολογίες (authority): Ο προσδιορισμός του αριθμού αυτού δεν είναι εφικτό να γίνει στην παρούσα περίπτωση, αφού δεν υπάρχει τρόπος εντοπισμού όλων των οντολογιών που χρησιμοποιούν τις προτεινόμενες οντολογίες, παρά μόνο οι οντολογίες που χρησιμοποιούνται από τις εφαρμογές που αναφέρθηκαν προηγουμένως [80]. Στις οντολογίες των εφαρμογών αυτών λοιπόν οι προτεινόμενες οντολογίες γίνονται αναφορά 7 φορές.
- Ο συνολικός αριθμός προσπελάσεων της υπό εξέταση οντολογίας (history): Και σε αυτή την περίπτωση δεν είναι εφικτός ο προσδιορισμός του αριθμού αυτού και μάλιστα ούτε και για τις οντολογίες των εφαρμογών του ερευνητικού προγράμματος [80], αφού δεν υπάρχει κάποιος μετρητής που να παρέχει την πληροφορία αυτή.

Αθροιστικά λοιπόν οι προτεινόμενες οντολογίες ικανοποιούν σχεδόν σε απόλυτο βαθμό όλα τα κριτήρια που προτείνονται στην [78], οδηγώντας ανεπιφύλακτα στο συμπέρασμα του ορθού και ολοκληρωμένου σχεδιασμού τους. Αυτό το συμπέρασμα ενισχύεται και από την υιοθέτηση – αναφορά των οντολογιών αυτών από πλήθος άλλων επιστημονικών εργασιών ([20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31]) .

3.3.5. Παράδειγμα εφαρμογής οντολογιών

Στην παράγραφο αυτή παρατίθεται ένα παράδειγμα εφαρμογής των οντολογιών αυτών «εμπνευσμένο» από το περιβάλλον νοημοσύνης (ambient intelligence) [81], το οποίο αναδεικνύει στο μέγιστο δυνατό βαθμό τις δυνατότητές τους υιοθετώντας και χρησιμοποιώντας τεχνικές και όρους του σημασιολογικού δικτύου [16].

Μια υπηρεσία εφαρμογής τους είναι η ακόλουθη: *Η Μαρία είναι μια ηλικιωμένη γυναίκα η οποία απαιτείται να έχει συχνή επαφή με το γιατρό της. Χρησιμοποιώντας ένα κατάλληλο ιατρικό μηχανήμα, της δίνεται η δυνατότητα να επικοινωνεί με το γιατρό της όσο συχνά το επιθυμεί ή προκύψει ανάγκη. Η ιδιαιτερότητα της επικοινωνίας επικεντρώνεται στο*

γεγονός ότι η Μαρία χρησιμοποιώντας την υποδομή του περιβάλλοντος νοημοσύνης (*ambient intelligence*), της δίνεται η δυνατότητα της «εικονικής» επαφής της με το γιατρό της κατά τη διάρκεια της οποίας μεταφέρονται εικόνες, ήχος και δεδομένα. Ο γιατρός μπορεί να βρίσκεται στο ιατρείο του, στο σπίτι του ή σε κάποιο άλλο χώρο και να εξετάζει την ασθενή του, να κάνει διάγνωση ή να της δίνει συμβουλές που πρέπει να ακολουθήσει, σαν να βρισκόταν εκείνη δίπλα του.

Η υπηρεσία που περιγράφηκε προηγουμένως θα μπορούσε να χαρακτηριστεί ως μια υπηρεσία «Τηλεϊατρικής με πολυμέσα» (*Multimedia Telemedicine*). Η υπηρεσία αυτή περιλαμβάνει πολλές παραμέτρους ποιότητας υπηρεσίας, προκειμένου η επικοινωνία να είναι εφικτή, αποτελεσματική και αξιόπιστη και προς τις δύο κατευθύνσεις. Μελετώντας μια τέτοια υπηρεσία μαζί με τα χαρακτηριστικά που εκείνη πρέπει να διαθέτει από τη μεριά του παρόχου, καταλήγει κανείς στο συμπέρασμα ότι ιδιαίτερη προσοχή πρέπει να δοθεί αρχικά σε παραμέτρους που σχετίζονται με την επίδοση του δικτύου, μια και υπάρχει ανάγκη μεταφοράς βίντεο σε πραγματικό χρόνο. Επομένως, ο ρυθμός δεδομένων πρέπει να έχει υψηλή τιμή, ενώ η καθυστέρηση, ο χρόνος απόκρισης, η μεταβολή καθυστέρησης και ο ρυθμός λαθών πρέπει να έχουν χαμηλές τιμές. Ακόμη, λόγω της ιδιαιτερότητας της υπηρεσίας, είναι εύλογο ότι πρέπει να είναι διαθέσιμη και προσπελάσιμη στο μέγιστο δυνατό βαθμό, ώστε να είναι εφικτή η χρήση της σε κάθε περίπτωση, αφού είναι μια υπηρεσία που βρίσκει εφαρμογή και σε επείγοντα περιστατικά. Εξάλλου, η κλιμάκωση και η χωρητικότητα του παρόχου πρέπει να είναι σε ένα επίπεδο που θα επιτρέπουν τη δυνατότητα προσαρμογής και εξυπηρέτησης πολλαπλών αιτήσεων / ασθενών, όταν καθίσταται ανάγκη. Το κόστος τέτοιων υπηρεσιών δύναται να προσδιορίζεται είτε ανά χρήση της υπηρεσίας, είτε ανά λεπτό (περίπτωση που επιλέχθηκε στην παρούσα περίπτωση, όπως θα φανεί και στα κομμάτια κώδικα που ακολουθούν). Επιπρόσθετα, πρέπει να ληφθούν υπόψη παράγοντες που σχετίζονται με τα πρότυπα που υποστηρίζει η υπηρεσία, τα οποία βέβαια πρέπει να επιλεγθούν κατάλληλα ώστε να είναι ευρείας αποδοχής, διευκολύνοντας κατ' αυτόν τον τρόπο τη χρήση της υπηρεσίας. Τέλος, λόγω της «ευαισθησίας» των δεδομένων που ανταλλάσσονται και λαμβάνοντας υπόψη το ιατρικό απόρρητο που πρέπει να τηρείται, η υπηρεσία θα πρέπει να υποστηρίζει τη δυνατότητα κρυπτογραφημένης επικοινωνίας μεταξύ γιατρού και ασθενούς.

Ακολουθούν κομμάτια κώδικα τόσο της οντολογίας Γλώσσας όσο και της οντολογίας Λεξιλόγιο που παριστάνουν την πληροφορία της υπηρεσίας «Τηλεϊατρικής με πολυμέσα» που μόλις αναλύθηκε υπό το πρίσμα των παραμέτρων ποιότητάς της. Υπενθυμίζεται ότι σαν εργαλείο ανάπτυξης των οντολογιών αυτών χρησιμοποιήθηκε το Protégé [76]. Εξάλλου, όσον αφορά την οντολογία Γλώσσα, επιλέχθηκε η παράθεση μιας μόνο παραμέτρου, του ρυθμού δεδομένων (*throughput*). Το κριτήριο επιλογής της συγκεκριμένης παραμέτρου είναι ότι επιτρέπει τη χρήση των περισσότερων κλάσεων και σχέσεων, όπως αυτές ορίζονται στην

παράγραφο 3.3.2. Όσον αφορά την οντολογία Λεξιλόγιο, αυτή περιλαμβάνει το σύνολο των παραμέτρων ποιότητας που μόλις αναλύθηκαν, οι οποίες συνοδεύονται από τις μονάδες τους, όπου αυτές υπάρχουν. Επισημαίνεται ότι για εργονομικούς λόγους κρίθηκε σκόπιμο να μη συμπεριληφθούν τα αρχεία των οντολογιών στη συνολική μορφή τους, αλλά μόνο τα κομμάτια εκείνα που σχετίζονται με την προσφερόμενη υπηρεσία.

Οντολογία Γλώσσα:

```
<QoSParameter rdf:ID="TelemedThroughput">
  <hasQoSImpact>
    <QoSImpact rdf:ID="HighImpact"/>
  </hasQoSImpact>
  <Node rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Server</Node>
  <Nature rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Dynamic</Nature>
  <hasType>
    <Type rdf:ID="TelemedPerformance"/>
  </hasType>
  <hasRelationship>
    <Relationship rdf:ID="TelemedThroughputvsTelemedErrorRate">
      <influentialParameter>
        <QoSParameter rdf:ID="TelemedErrorRate"/>
      </influentialParameter>
      <hasImpactFactor>
        <ImpactFactor rdf:ID="InvStrong">
          <IFType rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
          >InverselyProportional</IFType>
          <validityLevel rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
          >Strong</validityLevel>
        </ImpactFactor>
      </hasImpactFactor>
    </Relationship>
  </hasRelationship>
  <hasRelationship>
    <Relationship rdf:ID="TelemedThroughputvsTelemedCapacity">
      <hasImpactFactor>
        <ImpactFactor rdf:ID="PropStrong">
          <validityLevel rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
```

```

    >Strong</validityLevel>
    <IFType rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Proportional</IFType>
  </ImpactFactor>
</hasImpactFactor>
<influentialParameter>
  <QoSParameter rdf:ID="TelemedCapacity"/>
</influentialParameter>
</Relationship>
</hasRelationship>
<hasRelationship>
  <Relationship rdf:ID="TelemedThroughputvsTelemedSupportedStandard1">
    <influentialParameter rdf:resource="#TelemedSupportedStandard1"/>
    <hasImpactFactor>
      <ImpactFactor rdf:ID="InvWeak">
        <validityLevel rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >Weak</validityLevel>
        <IFType rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >InverselyProportional</IFType>
      </ImpactFactor>
    </hasImpactFactor>
  </Relationship>
</hasRelationship>
<hasRelationship>
  <Relationship rdf:ID="TelemedThroughputvsTelemedResponseTime">
    <influentialParameter rdf:resource="#TelemedResponseTime"/>
    <hasImpactFactor rdf:resource="#InvStrong"/>
  </Relationship>
</hasRelationship>
<hasMetric>
  <Metric rdf:ID="TelemedThroughputMetric">
    <hasUnit>
      <Unit rdf:ID="TelemedThroughputUnit">
        <hasConversionFormula>
          <ConversionFormula rdf:ID="bpsToKbps">
            <convertsTo rdf:resource="#TelemedThroughputUnit"/>
          </ConversionFormula>
        </hasConversionFormula>
      </Unit>
    </hasUnit>
  </Metric>
</hasMetric>

```

```

</Metric>
</hasMetric>
<hasRelationship>
  <Relationship rdf:ID="TelemedThroughputvsTelemedScalability">
    <influentialParameter rdf:resource="#TelemedScalability"/>
    <hasImpactFactor rdf:resource="#PropStrong"/>
  </Relationship>
</hasRelationship>
<hasRelationship>
  <Relationship rdf:ID="TelemedThroughputvsTelemedAccessibility">
    <influentialParameter rdf:resource="#TelemedAccessibility"/>
    <hasImpactFactor rdf:resource="#PropStrong"/>
  </Relationship>
</hasRelationship>
</QoSParameter>

```

Σχήμα 3-4 Παράδειγμα εφαρμογής της οντολογίας Γλώσσας σε μια υπηρεσία «Τηλεϊατρικής με πολυμέσα»

Στο παραπάνω τμήμα κώδικα, περιέχονται διάφορες πληροφορίες τόσο για το ρυθμό δεδομένων όσο και για τις σχέσεις που έχει η παράμετρος αυτή με άλλες παραμέτρους ποιότητας. Εξετάζοντας λοιπόν αυτό το τμήμα κώδικα, διαπιστώνει κανείς ότι η συγκεκριμένη μοντελοποίηση του ρυθμού δεδομένων έχει τα ακόλουθα χαρακτηριστικά: Είναι μια «Δυναμική» (Dynamic) παράμετρος που η τιμή της εξαρτάται από τον «Εξυπηρετητή» (Server), ανήκει στην κατηγορία Επίδοση (Performance), ο βαθμός επίδρασής της, όπως αυτός γίνεται αντιληπτός από το χρήστη, στη συνολική ποιότητα μιας διαδικτυακής υπηρεσίας είναι πολύ μεγάλος (HighImpact), ενώ σχετίζεται με διάφορες σχέσεις με τις άλλες παραμέτρους. Για παράδειγμα, η σχέση συσχέτισης μεταξύ του ρυθμού δεδομένων και της χωρητικότητας είναι «Ισχυρή» (Strong) και «Ανάλογη» (Proportional), ενώ η σχέση συσχέτισης του ρυθμού δεδομένων με τα υποστηριζόμενα πρότυπα είναι «Αδύναμη» (Weak) και «Αντιστρόφως Ανάλογη» (Inversely Proportional). Τέλος, η μοντελοποίηση αυτή υποστηρίζει τη δυνατότητα μετατροπής των μονάδων του ρυθμού δεδομένων από bps (bits per second) σε Kbps (Kilobits per second).

Οντολογία Λεξιλόγιο:

```
<Cost rdf:ID="TelemedCost">
  <TelemedicineService:QC_Metric rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Euro</TelemedicineService:QC_Metric>
  <TelemedicineService:QC_Value rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >0.20</TelemedicineService:QC_Value>
</Cost>

<Scalability rdf:ID="TelemedScalability">
  <TelemedicineService:QC_Value rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >0.80</TelemedicineService:QC_Value>
</Scalability>

<ResponseTime rdf:ID="TelemedResponseTime">
  <TelemedicineService:QC_Value rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >0.01</TelemedicineService:QC_Value>
  <TelemedicineService:QC_Metric rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >sec</TelemedicineService:QC_Metric>
</ResponseTime>

<MTBF rdf:ID="TelemedMTBF">
  <TelemedicineService:QC_Value rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >36000000</TelemedicineService:QC_Value>
  <TelemedicineService:QC_Metric rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >sec</TelemedicineService:QC_Metric>
</MTBF>

<SupportedStandard rdf:ID="TelemedSupportedStandard1">
  <TelemedicineService:QC_Value rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >UDDI 3.0</TelemedicineService:QC_Value>
</SupportedStandard>

<owl:DataRange>
  <owl:oneOf rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
</owl:DataRange>

<MTTR rdf:ID="TelemedMTTR">
  <TelemedicineService:QC_Metric rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >sec</TelemedicineService:QC_Metric>
  <TelemedicineService:QC_Value rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >1800</TelemedicineService:QC_Value>
</MTTR>

<DataEncryption rdf:ID="TelemedDataEncryption">
  <TelemedicineService:QC_Value rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >AES-128</TelemedicineService:QC_Value>
```

```

</DataEncryption>
<NonRepudiation rdf:ID="TelemedNonRepudiation">
  <TelemedicineService:QC_Value rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >1</TelemedicineService:QC_Value>
</NonRepudiation>
<Jitter rdf:ID="TelemedJitter">
  <TelemedicineService:QC_Metric rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >msec</TelemedicineService:QC_Metric>
  <TelemedicineService:QC_Value rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >1</TelemedicineService:QC_Value>
</Jitter>
<Authorization rdf:ID="TelemedAuthorization">
  <TelemedicineService:QC_Value rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >SSL</TelemedicineService:QC_Value>
</Authorization>
<Authentication rdf:ID="TelemedAuthentication">
  <TelemedicineService:QC_Value rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Password</TelemedicineService:QC_Value>
</Authentication>
<ErrorRate rdf:ID="TelemedErrorRate">
  <TelemedicineService:QC_Value rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >exp(-5)</TelemedicineService:QC_Value>
</ErrorRate>
<Delay rdf:ID="TelemedDelay">
  <TelemedicineService:QC_Metric rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >msec</TelemedicineService:QC_Metric>
  <TelemedicineService:QC_Value rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >300</TelemedicineService:QC_Value>
</Delay>
<Throughput rdf:ID="TelemedThroughput">
  <TelemedicineService:QC_Value rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >384</TelemedicineService:QC_Value>
  <TelemedicineService:QC_Metric rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Kbps</TelemedicineService:QC_Metric>
</Throughput>
<Auditability rdf:ID="TelemedAuditability">
  <TelemedicineService:QC_Value rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >1</TelemedicineService:QC_Value>
</Auditability>
<Accessibility rdf:ID="TelemedAccessibility">

```

```

<TelemedicineService:QC_Value rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>0.99000</TelemedicineService:QC_Value>
</Accessibility>
<Capacity rdf:ID="TelemedCapacity">
  <TelemedicineService:QC_Value rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >200</TelemedicineService:QC_Value>
</Capacity>
<SupportedStandard rdf:ID="TelemedSupportedStandard2">
  <TelemedicineService:QC_Value rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >WSDL 1.1</TelemedicineService:QC_Value>
</SupportedStandard>
<Availability rdf:ID="TelemedAvailability">
  <TelemedicineService:QC_Value rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >0.99995</TelemedicineService:QC_Value>
</Availability>
<Confidentiality rdf:ID="TelemedConfidentiality">
  <TelemedicineService:QC_Value rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >1</TelemedicineService:QC_Value>
</Confidentiality>

```

Σχήμα 3-5 Παράδειγμα εφαρμογής της οντολογίας Λεξιλόγιο σε μια υπηρεσία «Τηλεϊατρικής με πολυμέσα»

Στο παραπάνω τμήμα κώδικα περιέχονται διάφορες παράμετροι ποιότητας υπηρεσίας συνοδευόμενες από τις μονάδες τους, όπου αυτές υπάρχουν. Εξετάζοντας λοιπόν αυτό το τμήμα κώδικα, διαπιστώνει κανείς ότι μεταξύ άλλων η συγκεκριμένη υπηρεσία έχει χρέωση 0.20€, υποστηρίζει κρυπτογράφηση με αλγόριθμο κρυπτογράφησης τον AES-128, έχει πολύ υψηλή διαθεσιμότητα (0.99995) και προσπελασιμότητα (0.99000), ενώ ο ρυθμός δεδομένων που προσφέρεται η υπηρεσία είναι της τάξης των 384Kbps.

=====

Κεφάλαιο 4ο

4. Αλγόριθμοι και τεχνικές επίλυσης προβλημάτων που ανήκουν στο knapsack family

=====

Το κεφάλαιο αυτό αποτελεί μια θεωρητική εισαγωγή στους αλγορίθμους. Αν και δεν υπάρχει επίσημος ορισμός για το τι είναι ένας αλγόριθμος, ωστόσο ένας από τους πιο σαφείς ορισμούς είναι ο [82]: «...αλγόριθμος είναι μια πλήρως καθορισμένη υπολογιστική διαδικασία, η οποία ακολουθείται για την επίλυση ενός προβλήματος και η οποία δέχεται μια ή περισσότερες τιμές σαν είσοδο και παράγει μια ή περισσότερες τιμές σαν έξοδο...». Λόγω της ευρύτητας φάσματος των περιοχών που καλύπτει το αντικείμενο αυτό, η παρουσίαση θα περιοριστεί στις περιοχές εκείνες, οι οποίες θα αποτελέσουν το απαραίτητο θεωρητικό υπόβαθρο για το κεφάλαιο που ακολουθεί, στο οποίο προτείνονται αλγόριθμοι επιλογής υπηρεσιών. Ο ενδιαφερόμενος αναγνώστης μπορεί να ανατρέξει μεταξύ άλλων στις εργασίες [82], [83] και [84] για εκτεταμένη μελέτη του ερευνητικού πεδίου των αλγορίθμων.

Αναλυτικά, στην παράγραφο 4.1 παρουσιάζονται κάποια θεωρητικά στοιχεία που αφορούν την πολυπλοκότητα των αλγορίθμων, κατατάσσοντας τα προβλήματα που καλούνται να επιλύσουν σε διάφορες κατηγορίες (P, NP, NP-Complete κ.α.). Στην παράγραφο 4.2 μελετάται μια οικογένεια προβλημάτων που αποτελεί αντικείμενο έρευνας για πολλές δεκαετίες: η οικογένεια προβλημάτων του σακιδίου (knapsack family). Το κεφάλαιο ολοκληρώνεται με την ανάλυση των δύο πιο συχνά αντιμετωπίσιμων μορφών της οικογένειας αυτής: το 0-1 Knapsack Problem (§4.3) και το Multiple Choice Knapsack Problem (MCKP) (§4.4). Επισημαίνεται ότι στις μεθόδους που ακολουθούνται για την επίλυση των δύο αυτών προβλημάτων στηρίζονται και οι προτεινόμενοι αλγόριθμοι του επόμενου κεφαλαίου.

4.1. Πολυπλοκότητα Αλγορίθμων

Η μελέτη της πολυπλοκότητας ενός αλγορίθμου αναφέρεται στον υπολογισμό των πόρων (resources) – συνήθως χρόνου – που απαιτεί ο αλγόριθμος προκειμένου να επιλύσει ένα πρόβλημα [82]. Ο υπολογισμός αυτός επιτυγχάνεται συγκρίνοντας το χρόνο που απαιτείται να τερματίσει ο προτεινόμενος αλγόριθμος με το χρόνο που απαιτείται να τερματίσει στη χειρότερη περίπτωση μια συνάρτηση, όπου τερματισμός θεωρείται η εκτέλεση όλων των απαραίτητων πράξεων, ώστε για μια δεδομένη είσοδο να παραχθεί η έξοδος – λύση του προβλήματος. Πιο αναλυτικά, αν υποθεθεί ότι ο αλγόριθμος που επιλύει το πρόβλημα μοντελοποιείται από τη συνάρτηση $f(n)$, ενώ η συνάρτηση σύγκρισης είναι η $g(n)$ όπου n φυσικός αριθμός, τότε οι ακόλουθοι συμβολισμοί χαρακτηρίζουν την πολυπλοκότητα ενός αλγορίθμου [82]:

- $\Theta(g(n))$ όταν υπάρχουν θετικοί ακέραιοι c_1 , c_2 και n_0 ώστε να ισχύει η σχέση $0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)$ για κάθε $n \geq n_0$.
- $O(g(n))$ όταν υπάρχουν θετικοί ακέραιοι c και n_0 ώστε να ισχύει η σχέση $0 \leq f(n) \leq c g(n)$ για κάθε $n \geq n_0$. Ο συμβολισμός αυτός είναι ο πιο συνήθης για τον χαρακτηρισμό της πολυπλοκότητας ενός αλγορίθμου.
- $\Omega(g(n))$ όταν υπάρχουν θετικοί ακέραιοι c και n_0 ώστε να ισχύει η σχέση $0 \leq c g(n) \leq f(n)$ για κάθε $n \geq n_0$.

Σε ορισμένες περιπτώσεις χρησιμοποιούνται και οι συμβολισμοί $o(g(n))$ - $\omega(g(n))$ που διαφέρουν σε σχέση με τους $O(g(n))$ - $\Omega(g(n))$ αντίστοιχα στο ότι οι ανισότητες ισχύουν για όλες τις σταθερές $c > 0$ και όχι για κάποιες θετικές σταθερές c μόνο. Όπως διακρίνεται εύκολα από τους παραπάνω συμβολισμούς, η πολυπλοκότητα ενός αλγορίθμου προσδιορίζεται με τη δημιουργία κατάλληλων φραγμάτων. Δύο είναι οι μεγάλες κατηγορίες συναρτήσεων που χρησιμοποιούνται σαν συναρτήσεις σύγκρισης: (i) Οι πολυωνυμικές συναρτήσεις $g(n)$, όπου το πρόβλημα επιλύεται σε υπολογιστικό χρόνο και (ii) οι εκθετικές συναρτήσεις $g(n)$, όπου το πρόβλημα επιλύεται σε υπολογιστικό χρόνο για μικρές μόνο εισόδους n . Για να καταστεί αυτό περισσότερο σαφές, στον παρακάτω πίνακα γίνεται μια σύγκριση του χρόνου που απαιτείται από μια πολυωνυμική συνάρτηση ($g(n) = n^3$) και μια εκθετική συνάρτηση ($g(n) = 3^n$), καθώς αυξάνεται η είσοδος (n):

$g(n)$	$n = 10$	$n = 20$	$n = 30$	$n = 40$	$n = 50$
n^3	0.001sec	0.008sec	0.027sec	0.064sec	0.125sec
3^n	0.059sec	>58min	>6.5years	>3.85*10 ³ cent.	>0.227*10 ⁹ cent.

Πίνακας 4-1 Σύγκριση χρόνου εκτέλεσης πολυωνυμικής (n^3) και εκθετικής (3^n) συνάρτησης

Όπως φαίνεται και από τον παραπάνω πίνακα, η εκτέλεση της εκθετικής συνάρτησης ακόμα και για μικρές εισόδους δεν είναι εφικτό να πραγματοποιηθεί σε πεπερασμένο χρόνο σε αντίθεση με την πολυωνυμική συνάρτηση. Η παρατήρηση αυτή, που ισχύει καθολικά για τις συναρτήσεις αυτές, οδήγησε στο συμπέρασμα ότι είναι επιθυμητό ο αλγόριθμος να έχει φράγμα πολυωνυμική συνάρτηση ή αλλιώς να έχει πολυωνυμική και όχι εκθετική πολυπλοκότητα [85], [86]. Αξίζει να αναφερθεί ότι δεν είναι όλα τα προβλήματα επιλύσιμα, όπως έχει αποδειχθεί από τον Turing [87]. Από τον Turing επίσης έχει προταθεί η ντετερμινιστική μηχανή (DTM) (Deterministic Turing Machine), η οποία χρησιμοποιείται ως υπολογιστικό μοντέλο για τη μελέτη της πολυπλοκότητας ενός αλγόριθμου. Εν συντομία, η μηχανή αυτή περιλαμβάνει μια κασέτα εγγραφής και διαθέτει μια συνάρτηση, η οποία δοθέντος της τρέχουσας κατάστασης της μηχανής υπολογίζει το επόμενο βήμα που πρέπει να ακολουθηθεί. Για παράδειγμα αν η μηχανή βρίσκεται στην κατάσταση (state) 4 με το γράμμα X, τότε η συνάρτηση αυτή καθορίζει το επόμενο βήμα, το οποίο για παράδειγμα μπορεί να είναι να κινηθεί η κεφαλή της κασέτας μια θέση προς τα αριστερά, να γραφτεί το γράμμα O και να μεταβεί στην κατάσταση 2. Ένας αλγόριθμος λοιπόν θεωρείται ότι είναι πολυωνυμικής πολυπλοκότητας DTM όταν χρησιμοποιώντας τη μηχανή αυτή, η μηχανή αυτή για κάθε είσοδο υπολογίζει την επιθυμητή έξοδο μετά από έναν αριθμό βημάτων που το πλήθος τους φράσσεται από μια πολυωνυμική συνάρτηση. Η μηχανή αυτή χρησιμοποιείται για το διαχωρισμό των προβλημάτων απόφασης (πρόβλημα απόφασης: πρόβλημα το οποίο έχει σαν απάντηση ένα ναι ή ένα όχι, όπως για παράδειγμα αν ένας αριθμός είναι πρώτος) σε δύο μεγάλες κλάσεις: Στην κλάση P (deterministic Polynomial time), όπου τα προβλήματα αυτά επιλύονται από μια μηχανή DTM και στην κλάση NP (Non-deterministic Polynomial time) τα οποία επιλύονται από μια μηχανή NTM (Non-deterministic Turing Machine). Η μηχανή αυτή διαφέρει από την DTM στο ότι μπορεί να εκτελέσει περισσότερες από μια ανεξάρτητες ακολουθίες πράξεων ταυτόχρονα. Οι δύο αυτές κλάσεις έχουν αποτελέσει αντικείμενο έρευνας και προβληματισμού για τους μαθηματικούς για πολλές δεκαετίες χωρίς να έχουν καταλήξει ακόμα στο συμπέρασμα αν η κλάση P είναι ισοδύναμη ή υποκλάση της κλάσης NP , αν και υπάρχουν σαφείς ενδείξεις που ενισχύουν το δεύτερο ισχυρισμό [88]. Αξίζει να επισημανθεί ότι τα προβλήματα απόφασης NP έχουν την ιδιότητα ότι δοθέντος μιας λύσεως, μπορεί να εξακριβωθεί σε πολυωνυμικό χρόνο η ορθότητα της λύσεως. Για παράδειγμα στο πρόβλημα του Αθροίσματος Υποσύνολου (subset sum problem) [82] - στο οποίο σαν είσοδος θεωρείται πεπερασμένο πλήθος ακέραιων αριθμών και ζητείται η επιλογή

υποσυνόλου από αυτούς, ώστε το άθροισμά τους να είναι 0 - μπορεί να ελεγχθεί εύκολα αν ένα προτεινόμενο υποσύνολο έχει άθροισμα 0 ή όχι, ωστόσο δεν υπάρχουν αλγόριθμοι που να λύνουν το πρόβλημα αυτό όταν το πλήθος των ακεραίων της εισόδου είναι πολύ μεγάλο. Το σημαντικότερο πρόβλημα της κλάσης αυτής είναι το ικανοποιήσιμο πρόβλημα (Satisfiability problem), το οποίο διαθέτει την ακόλουθη ιδιότητα: Κάθε πρόβλημα της κλάσης NP μπορεί να αναχθεί σε πολυωνυμικό χρόνο σε αυτό το πρόβλημα. Αυτό βέβαια σημαίνει ότι αν βρεθεί λύση για το πρόβλημα αυτό σε πολυωνυμικό χρόνο, τότε και όλα τα προβλήματα της κλάσης αυτής θα επιδέχονται παρόμοια λύση, καθιστώντας το υπό αυτήν την έννοια ως το δυσκολότερο πρόβλημα της κατηγορίας αυτής [83]. Η σημαντικότερη κατηγορία προβλημάτων που περιλαμβάνεται στην κλάση NP είναι τα NP - πλήρη (NP - Complete) προβλήματα [88]. Η κατηγορία αυτή περιλαμβάνει τα δυσκολότερα προβλήματα της κλάσης αυτής, τα οποία χαρακτηρίζονται από τις ιδιότητες ότι (i) κάθε πρόβλημα της κατηγορίας αυτής πρέπει να ανήκει στην κλάση NP και (ii) κάθε πρόβλημα της κλάσης αυτής μπορεί να αναχθεί σε αυτά. Επομένως διαθέτουν τα ίδια χαρακτηριστικά με το ικανοποιήσιμο πρόβλημα (Satisfiability problem). Μια άλλη πολύ σημαντική κατηγορία προβλημάτων είναι τα NP - Δύσκολα (NP - Hard) προβλήματα [83], [89]. Η κατηγορία αυτή χωρίζεται περαιτέρω σε δύο υποκατηγορίες: (i) Στα ασθενή NP - Δύσκολα (weak NP - Hard), για τα οποία υπάρχουν ψευδο - πολυωνυμικοί αλγόριθμοι (pseudo-polynomial), δηλαδή αλγόριθμοι πολυωνυμικού χρόνου όταν η είσοδος δεν είναι εκθετικού μεγέθους και (ii) Στα ισχυρά NP - Δύσκολα (strong NP - Hard) για τα οποία υπάρχουν ψευδο - πολυωνυμικοί αλγόριθμοι, μόνο στην περίπτωση που ισχύει $P = NP$. Επισημαίνεται ότι στην κατηγορία αυτή περιλαμβάνονται προβλήματα που δεν είναι απαραίτητο να περιλαμβάνονται στην κλάση NP και που διαθέτουν βαθμό δυσκολίας μεγαλύτερο ή ίσο με αυτόν της κατηγορίας NP - Πλήρη.

4.2. Οικογένεια προβλημάτων σακιδίου

Αντικείμενο της παραγράφου αυτής είναι η παρουσίαση των σημαντικότερων προβλημάτων της οικογένειας του σακιδίου (knapsack family). Για τον ενδιαφερόμενο αναγνώστη ολοκληρωμένη αναφορά και ανάλυση όλων των προβλημάτων της οικογένειας αυτής υπάρχουν μεταξύ άλλων στις εργασίες [90], [91] και [92].

Η οικογένεια προβλημάτων του σακιδίου ανήκει στην κατηγορία των προβλημάτων NP - Δύσκολα (NP - Hard) [83]. Επομένως, σύμφωνα με την προηγούμενη παράγραφο και εφόσον δεν έχει αποδειχθεί ακόμα η σχέση $P = NP$, δεν υπάρχουν αλγόριθμοι που να επιλύουν τα προβλήματα αυτά σε πολυωνυμικό χρόνο, ενώ η μοναδική μέθοδος που δίνει ακριβή λύση είναι η απαρίθμηση (enumeration) όλων των δυνατών περιπτώσεων. Η μέθοδος αυτή βέβαια δεν είναι πρακτική και εφαρμόσιμη όταν η είσοδος αποτελείται από μεγάλο πλήθος όρων, κάτι που οδήγησε στη δημιουργία προσεγγιστικών αλγορίθμων ([93], [94], [95]). Ωστόσο, όταν το πρόβλημα που καλείται να αντιμετωπιστεί έχει είσοδο με

περιορισμένο αριθμό όρων - και κατά συνέπεια μπορεί να βρεθεί ακριβής λύση με απαρίθμηση σε πραγματικό χρόνο - οι δημοφιλέστερες μέθοδοι που εφαρμόζονται στηρίζονται στο *δυναμικό προγραμματισμό* (*Dynamic Programming*) ([96], [97], [98]) και στην *επέκταση και οριοθέτηση* (*Branch and Bound*) ([99], [100], [101]). Οι μέθοδοι αυτές διαθέτουν το πλεονέκτημα ότι δεν κάνουν πλήρη απαρίθμηση όλων των δυνατών περιπτώσεων, αλλά αντίθετα στηριζόμενες σε κάποια τεχνάσματα και καθορίζοντας κάποια όρια, επιτυγχάνουν να περιορίσουν το πλήθος των εξεταζόμενων - πιθανών λύσεων και κατά συνέπεια το χρόνο εύρεσης βέλτιστης λύσης. Για παράδειγμα η τεχνική της *επέκτασης και οριοθέτησης* δημιουργεί δέντρα (trees), όπου σε κάθε κόμβο (node - leaf) υπολογίζεται βάσει συγκεκριμένων τεχνικών άνω και κάτω όρια (upper and lower bounds). Αν λοιπόν αναζητείται στο πρόβλημα η ελάχιστη τιμή κάποιας αντικειμενικής συνάρτησης (objective function) και το κάτω όριο σε έναν κόμβο A υπερβαίνει την τιμή κάποιου άνω ορίου ενός άλλου κόμβου B, τότε όπως είναι εύλογο ο κόμβος A καθώς και τα «παιδιά» του αποκλείεται να συμμετέχουν στον υπολογισμό της ελάχιστης - βέλτιστης τιμής της αντικειμενικής αυτής συνάρτησης. Επομένως μπορούν με ασφάλεια να αγνοηθούν και κατά συνέπεια να μη δαπανηθεί χρόνος στην εξέταση των προσφερόμενων λύσεων από τα «παιδιά» αυτά.

Συνοπτικά, η οικογένεια προβλημάτων του σακιδίου περιλαμβάνει προβλήματα στα οποία καλείται να επιλεγεί από μια ομάδα αντικειμένων που βρίσκονται μέσα σε ένα ή περισσότερα σακίδια μια υποομάδα αυτών, προκειμένου να βελτιστοποιηθεί (μεγιστοποιηθεί ή ελαχιστοποιηθεί) μια αντικειμενική συνάρτηση (objective function), ενώ παράλληλα θα ικανοποιούνται κάποιοι περιορισμοί. Αναλυτικά τα προβλήματα αυτά είναι:

Το πιο δημοφιλές πρόβλημα της οικογένειας είναι το *0-1 Knapsack Problem* ή *πρόβλημα του «κλέφτη»* όπως συνηθίζεται να λέγεται. Στο πρόβλημα αυτό θεωρείται ότι ένας κλέφτης εισέρχεται σε ένα διαμέρισμα με ένα σάκο χωρητικότητας c . Στο διαμέρισμα αυτό υπάρχουν N πλήθος αντικειμένων, διαφορετικής αξίας και βάρους το κάθε ένα από αυτά. Αυτό που ζητείται είναι η επιλογή εκείνων των αντικειμένων, τα οποία θα αποφέρουν το μεγαλύτερο κέρδος για τον κλέφτη. Η μαθηματική μοντελοποίηση του προβλήματος αυτού είναι η ακόλουθη:

$$\text{μεγιστοποίηση} \quad \sum_{i=1}^N p_i * x_i$$

$$\text{ώστε} \quad \sum_{i=1}^N w_i * x_i \leq c$$

$$x_i \in \{0,1\}$$

$$i = 1, 2, \dots, N$$

όπου ο δείκτης i παριστάνει το εκάστοτε αντικείμενο. Η μεταβλητή p_i παριστάνει την αξία του κάθε αντικειμένου, ενώ η μεταβλητή w_i παριστάνει το βάρος του. Η μεταβλητή x_i υποδεικνύει αν το αντικείμενο i επιλέγεται τελικά από τον κλέφτη και εισάγεται στο σάκο του ή όχι. Σε περίπτωση που επιλέγεται, παίρνει την τιμή 1, αλλιώς την τιμή 0. Εξάλλου, όπως έχει ήδη αναφερθεί, στις μεθόδους που εφαρμόζονται στην επίλυση του προβλήματος αυτού στηρίζονται και οι προτεινόμενοι αλγόριθμοι επιλογής υπηρεσιών του επόμενου κεφαλαίου. Για το λόγο αυτό κρίνεται σκόπιμο να αναλυθούν οι μέθοδοι αυτές στην επόμενη παράγραφο (§4.3).

Σε περίπτωση όπου το κάθε αντικείμενο μπορεί να συμπεριληφθεί περισσότερες από μια φορές, τότε το πρόβλημα μετασχηματίζεται στο *Bounded Knapsack Problem*. Η μαθηματική μοντελοποίηση του προβλήματος αυτού είναι η ακόλουθη:

$$\begin{aligned} \text{μεγιστοποίηση} \quad & \sum_{i=1}^N p_i * x_i \\ \text{ώστε} \quad & \sum_{i=1}^N w_i * x_i \leq c \\ & x_i \in \{0, 1, \dots, m_i\} \quad i = 1, 2, \dots, N \end{aligned}$$

Η μόνη διαφορά με το 0-1 KP, είναι οι τιμές που παίρνει η μεταβλητή x_i , οι οποίες επισημαίνεται ότι είναι διαφορετικές ανά αντικείμενο.

Όταν δεν υπάρχει όριο στο πλήθος των φορών που μπορεί να εισαχθεί κάποιο αντικείμενο στο σακίδιο, τότε το πρόβλημα μετασχηματίζεται στο *Unbounded Knapsack Problem*. Η μαθηματική μοντελοποίηση του προβλήματος αυτού είναι η ακόλουθη:

$$\begin{aligned} \text{μεγιστοποίηση} \quad & \sum_{i=1}^N p_i * x_i \\ \text{ώστε} \quad & \sum_{i=1}^N w_i * x_i \leq c \\ & x_i \geq 0 \quad i = 1, 2, \dots, N \end{aligned}$$

ενώ είναι προφανές ότι επειδή αναφερόμαστε σε πλήθος αντικειμένων, η μεταβλητή x_i παίρνει μόνο ακέραιες τιμές.

Στην οικογένεια των προβλημάτων του σακιδίου ανήκει και το *Multiple Choice Knapsack Problem*, στο οποίο υπάρχουν πολλαπλοί σάκοι (N) που διαθέτουν αντικείμενα διαφορετικής αξίας και βάρους και ζητείται η επιλογή ενός ακριβώς αντικειμένου από κάθε έναν από τους σάκους αυτούς, προκειμένου να μεγιστοποιηθεί η συνολική αξία, ενώ το

συνολικό βάρος δε θα πρέπει να ξεπερνάει μια προκαθορισμένη τιμή. Η μαθηματική διατύπωση του προβλήματος αυτού είναι:

$$\begin{aligned}
 &\text{μεγιστοποίηση} \quad \sum_{i=1}^N \sum_{j=1}^{M_i} p_{ij} * x_{ij} \\
 &\text{ώστε} \quad \sum_{i=1}^N \sum_{j=1}^{M_i} w_{ij} * x_{ij} \leq c \\
 &\quad \sum_{j=1}^{M_i} x_{ij} = 1 \quad i = 1, 2, \dots, N \\
 &\quad x_{ij} \in \{0, 1\} \quad i = 1, 2, \dots, N, \quad j \in M_i
 \end{aligned}$$

όπου ο δείκτης i παριστάνει τον εκάστοτε σάκο, ενώ ο δείκτης j παριστάνει τα αντικείμενα που βρίσκονται στο σάκο αυτό. Επισημαίνεται ότι ο αριθμός των αντικειμένων που βρίσκονται στον κάθε σάκο είναι διαφορετικός (δηλαδή η τιμή της παραμέτρου M_i είναι διαφορετική για κάθε i). Εξάλλου όπως και πριν η μεταβλητή p_{ij} παριστάνει την αξία του αντικειμένου j που βρίσκεται στο σακίδιο i , ενώ η μεταβλητή w_{ij} παριστάνει το βάρος του. Η μεταβλητή x_{ij} υποδεικνύει αν το αντικείμενο j που βρίσκεται στο σακίδιο i επιλέγεται ή όχι. Σε περίπτωση που επιλέγεται, παίρνει την τιμή 1, αλλιώς την τιμή 0. Τέλος, το άθροισμα του συνολικού βάρους των επιλεγμένων αντικειμένων δε θα πρέπει να υπερβαίνει μια προκαθορισμένη τιμή c . Το πρόβλημα αυτό έχει μελετηθεί ίσως περισσότερο από κάθε άλλο πρόβλημα της οικογένειας αυτής, ενώ υπενθυμίζεται ότι στους αλγορίθμους που επιλύουν το πρόβλημα αυτό έχουν στηριχθεί και οι προτεινόμενοι αλγόριθμοι του επόμενου κεφαλαίου. Κρίνεται σκόπιμο επομένως να αναλυθούν οι αλγόριθμοι αυτοί σε επόμενη παράγραφο (§4.4).

Σε περίπτωση που η αξία του κάθε αντικειμένου ισούται με το βάρος του, τότε το πρόβλημα ονομάζεται *Άθροισμα – Υποσυνόλου* (*Subset – Sum*). Αντικείμενο του προβλήματος αυτού είναι η κατάλληλη επιλογή αντικειμένων, ώστε να προσεγγιστεί στο μεγαλύτερο δυνατό βαθμό ο περιορισμός της χωρητικότητας c . Η μαθηματική μοντελοποίησή του είναι η ακόλουθη:

$$\begin{aligned}
 &\text{μεγιστοποίηση} \quad \sum_{i=1}^N w_i * x_i \\
 &\text{ώστε} \quad \sum_{i=1}^N w_i * x_i \leq c
 \end{aligned}$$

$$x_i \in \{0,1\} \quad i = 1,2,\dots, N$$

Ένα άλλο πολύ δημοφιλές πρόβλημα της οικογένειας αυτής είναι το *πρόβλημα του «ταμιά»* (*Change-making*) όπως συνηθίζεται να λέγεται. Στο πρόβλημα αυτό θεωρείται ότι υπάρχουν N διαφορετικά νομίσματα – αντικείμενα, διαφορετικής αξίας το καθένα. Αυτό που ζητείται είναι η ελαχιστοποίηση του συνολικού αριθμού των νομισμάτων (αντικειμένων) που πρέπει να επιλεγθούν, ώστε το άθροισμά της αξίας τους να ισούται με μια προκαθορισμένη τιμή c . Η λύση του προβλήματος αυτού δηλαδή «προσομοιώνει» τη συμπεριφορά ενός ταμιά, ο οποίος καλείται να επιλέξει τον ελάχιστο αριθμό από τα νομίσματα που διαθέτει, προκειμένου να καταβάλλει το επιθυμητό αντίτιμο στον εκάστοτε πελάτη. Η μαθηματική μοντελοποίησή του είναι η ακόλουθη:

$$\begin{aligned} &\text{ελαχιστοποίηση} \quad \sum_{i=1}^N x_i \\ &\text{ώστε} \quad \sum_{i=1}^N w_i * x_i = c \end{aligned}$$

$$x_i \geq 0 \quad i = 1,2,\dots, N$$

ενώ επειδή αναφερόμαστε σε πλήθος αντικειμένων, η μεταβλητή x_i παίρνει μόνο ακέραιες τιμές.

Ένα άλλο πολύ ενδιαφέρον πρόβλημα της οικογένειας του σακιδίου είναι το *Multiple Knapsack Problem*. Στο πρόβλημα αυτό ζητείται η επιλογή υποσυνόλου από μια ομάδα N αντικειμένων – όπου όπως και πριν έχουν διαφορετική αξία και βάρος – με στόχο να μεγιστοποιηθεί το συνολικό κέρδος – αξία, ενώ παράλληλα θα πρέπει να εισαχθούν τα αντικείμενα αυτά σε σακίδια διαφορετικής χωρητικότητας c_i . Η μαθηματική μοντελοποίηση του προβλήματος αυτού είναι:

$$\begin{aligned} &\text{μεγιστοποίηση} \quad \sum_{i=1}^M \sum_{j=1}^N p_j * x_{ij} \\ &\text{ώστε} \quad \sum_{j=1}^N w_j * x_{ij} \leq c_i \quad i = 1,2,\dots, M \\ &\quad \quad \quad \sum_{i=1}^M x_{ij} \leq 1 \quad j = 1,2,\dots, N \\ &\quad \quad \quad x_{ij} \in \{0,1\} \quad i = 1,2,\dots, M, \quad j = 1,2,\dots, N \end{aligned}$$

Όπως και πριν, η μεταβλητή x_{ij} υποδεικνύει αν το αντικείμενο j επιλέγεται να εισαχθεί στο σακίδιο i ή όχι. Σε περίπτωση που επιλέγεται, παίρνει την τιμή 1, αλλιώς την τιμή 0.

Επισημαίνεται ότι ο περιορισμός $\sum_{i=1}^M x_{ij} \leq 1$ πιστοποιεί ότι το κάθε αντικείμενο δε μπορεί να επιλεγθεί περισσότερο από μια φορές.

Το *Multi-constrained Multi-dimensional Knapsack Problem (MMKP)* είναι άλλο ένα πρόβλημα της οικογένειας του σακιδίου που έχει μελετηθεί εκτενώς. Διαφοροποιείται σε σχέση με το *MCKP* ως προς το σύνολο των περιορισμών. Η μαθηματική διατύπωση του προβλήματος αυτού είναι:

$$\begin{aligned} \text{μεγιστοποίηση} \quad & \sum_{i=1}^N \sum_{j=1}^{M_i} p_{ij} * x_{ij} \\ \text{ώστε} \quad & \sum_{i=1}^t \sum_{j=1}^{M_i} w_{ij} * x_{ij} \leq c_t \quad t = 1, 2, \dots, N \\ & \sum_{j=1}^{M_i} x_{ij} = 1 \quad i = 1, 2, \dots, N \\ & x_{ij} \in \{0, 1\} \quad i = 1, 2, \dots, N, \quad j \in M_i \end{aligned}$$

Τέλος, η γενικότερη μορφή των προβλημάτων της οικογένειας του σακιδίου είναι το *Multi-constrained Knapsack Problem*, το οποίο είναι ένα πρόβλημα *ακέραιου προγραμματισμού (Integer Programming)* [102], όπου όλες οι παράμετροι (p_i, w_i, x_i) είναι θετικοί, μη μηδενικοί ακέραιοι. Όπως εύκολα διαπιστώνεται η διαφορά του με το *Unbounded Knapsack Problem*, είναι ο αριθμός των περιορισμών (M) που πρέπει να ικανοποιηθούν. Η μαθηματική μοντελοποίησή του είναι:

$$\begin{aligned} \text{μεγιστοποίηση} \quad & \sum_{i=1}^N p_i * x_i \\ \text{ώστε} \quad & \sum_{i=1}^N w_i * x_i \leq c_j \quad j = 1, 2, \dots, M \\ & x_i \geq 0 \quad i = 1, 2, \dots, N \end{aligned}$$

4.3. 0-1 Knapsack Problem (KP)

Αντικείμενο της παραγράφου αυτής αποτελεί η παρουσίαση των τεχνικών - μεθόδων που εφαρμόζονται κατά την επίλυση του *0-1 Knapsack Problem (KP)*. Επισημαίνεται ότι οι μέθοδοι αυτές στοχεύουν σε προσεγγιστική και όχι ακριβή λύση του προβλήματος αυτού.

Επομένως, οι τεχνικές αυτές μπορούν να υιοθετηθούν από ευρυστικούς (heuristic) αλγορίθμους, αφού όπως έχει ήδη αναφερθεί και στην προηγούμενη παράγραφο, το πρόβλημα αυτό ανήκει στην κατηγορία προβλημάτων NP – Δύσκολα (NP – Hard) [83], επομένως δεν επιδέχεται ακριβή λύση σε κάθε περίπτωση. Ακριβείς λύσεις που εφαρμόζονται μόνο στην περίπτωση που υπάρχει περιορισμένος αριθμός αντικειμένων και στηρίζονται στις τεχνικές του *δυναμικού προγραμματισμού (Dynamic Programming)* και στην *επέκταση και οριοθέτηση (Branch and Bound)* παρουσιάζονται μεταξύ άλλων στις εργασίες [99], [103] και [104].

Όπως αναφέρθηκε και στην προηγούμενη παράγραφο, το πρόβλημα *0-1 KP* στοχεύει στην επιλογή αντικειμένων από μια ομάδα N αντικειμένων – διαφορετικής αξίας και βάρους - ώστε να μεγιστοποιηθεί το άθροισμα της αξίας των επιλεγμένων αντικειμένων, ενώ δε θα πρέπει να υπερβαίνει το συνολικό βάρος τους μια προκαθορισμένη τιμή c . Υπενθυμίζεται η μαθηματική διατύπωση του προβλήματος αυτού, η οποία είναι:

$$\begin{aligned} \text{μεγιστοποίηση} \quad & \sum_{i=1}^N p_i * x_i \\ \text{ώστε} \quad & \sum_{i=1}^N w_i * x_i \leq c \\ & x_i \in \{0,1\} \quad i = 1,2,\dots,N \end{aligned}$$

όπου ο δείκτης i παριστάνει το εκάστοτε αντικείμενο. Η μεταβλητή p_i παριστάνει την αξία του κάθε αντικειμένου, ενώ η μεταβλητή w_i παριστάνει το βάρος του. Η μεταβλητή x_i υποδεικνύει αν το αντικείμενο i επιλέγεται τελικά ή όχι. Σε περίπτωση που επιλέγεται, παίρνει την τιμή 1, αλλιώς την τιμή 0.

Είναι εύλογο ότι το πρόβλημα *0-1 KP* θα είχε προφανή λύση, αν ήταν εφικτό να επιλεγθεί «μερικώς» ένα αντικείμενο. Στην περίπτωση αυτή θα είχαμε το «*συνεχές KP*» (*continuous KP*) όπως συνηθίζεται να λέγεται, το οποίο έχει την εξής πολύ απλή λύση [105]:

- Διατάσσονται όλα τα αντικείμενα σε φθίνουσα σειρά σύμφωνα με το λόγο

$\frac{p_i}{w_i}$, σχηματίζοντας την ακόλουθη διάταξη:

$$\frac{p_1}{w_1} \geq \frac{p_2}{w_2} \geq \dots \frac{p_i}{w_i} \geq \dots \geq \frac{p_N}{w_N} \quad \text{όπου } i = 1,2,\dots,N$$

Ο λόγος αυτός όπως είναι φανερό ταξινομεί τα αντικείμενα βάσει της ωφελιμότητας που έχει η επιλογή του συγκεκριμένου αντικειμένου, αφού όσο μεγαλύτερος ο λόγος $\frac{p_i}{w_i}$, τόσο πιο πολύτιμο είναι το αντικείμενο αυτό.

- Από την παραπάνω διάταξη επιλέγονται τα αντικείμενα εκείνα (με προτεραιότητα από τα αριστερά προς τα δεξιά) τα οποία δεν παραβιάζουν τον περιορισμό του συνολικού βάρους c . Όταν κατά την επιλογή αυτή, το βάρος κάποιου αντικειμένου – έστω b – ξεπεράσει τον περιορισμό αυτό, η προσθήκη αντικειμένων σταματά, ενώ το αντικείμενο αυτό επιλέγεται «μερικώς», σύμφωνα με τη σχέση:

$$\sum_{i=1}^{b-1} w_i + a * w_b = c \Rightarrow a = \frac{1}{w_b} (c - \sum_{i=1}^{b-1} w_i)$$

Το μέγιστο – βέλτιστο κέρδος σε αυτή την περίπτωση είναι:

$$\max = \sum_{i=1}^{b-1} p_i + \frac{p_b}{w_b} (c - \sum_{i=1}^{b-1} w_i)$$

Ο όρος b ονομάζεται *break item* και χρησιμοποιείται για τον ορισμό των άνω και κάτω ορίων. Επισημαίνεται ότι υπάρχουν διάφοροι ορισμοί των ορίων αυτών. Ο ορισμός του άνω ορίου που προτάθηκε από τον Dantzig [105] ταυτίζεται με το συνολικό κέρδος που υπολογίζεται κατά την επίλυση του «συνεχούς KP» (*continuous KP*). Επομένως, σύμφωνα με τον ορισμό αυτόν ένα άνω όριο είναι το:

$$UpperBound = \sum_{i=1}^{b-1} p_i + \frac{p_b}{w_b} (c - \sum_{i=1}^{b-1} w_i)$$

Ο ορισμός του κάτω ορίου μπορεί να επιτευχθεί κάνοντας χρήση «άπληστων» (*greedy*) μεθόδων, σύμφωνα με τις οποίες ως κάτω όριο υπολογίζεται το συνολικό άθροισμα της αξίας των όρων - αντικειμένων μέχρι τον *break item*, ενώ στο άθροισμα αυτό προστίθεται ο πρώτος όρος μετά τον *break item*, ο οποίος δεν υπερβαίνει τον περιορισμό του συνολικού βάρους c .

Η μαθηματική διατύπωση του ορισμού αυτού είναι:

$$LowerBound = \sum_{i=1}^{b-1} p_i + p_m$$

$$\text{με } \sum_{i=1}^{b-1} w_i + w_m \leq c, b < m \leq N$$

Έχοντας λύσει το «συνεχές KP» (*continuous KP*) και έχοντας υπολογίσει τα άνω και κάτω όρια, ο Dantzig [105] καθώς και άλλοι ([90], [95] και [106]) όρισαν έναν «πυρήνα» (*core*), ως την περιοχή εκατέρωθεν (συμμετρική) του *break item*. Ο ορισμός αυτός οφείλεται στην

ακόλουθη πειραματική παρατήρηση: η βέλτιστη λύση μπορεί να υπολογιστεί, κάνοντας μια απλή αντιμετάθεση όρων - αντικειμένων στην περιοχή γύρω από τον «πυρήνα». (Η παρατήρηση αυτή βέβαια δεν είναι απόλυτη ακριβής, όπως θα αποδειχθεί και στο αριθμητικό παράδειγμα που ακολουθεί.) Πλήθος διαφορετικών ορισμών πυρήνων έχει προταθεί στη διεθνή βιβλιογραφία, με άλλους ορισμούς να περιλαμβάνουν σταθερό μέγεθος πυρήνα ([90]) και άλλους μεταβλητού μεγέθους ([95]), στους οποίους το μέγεθος αυξάνεται σύμφωνα με το επιθυμητό επίπεδο απόκλισης από τη βέλτιστη λύση. Κάνοντας χρήση «ευφύων» τεχνικών απαριθμήσεων (*δυναμικού προγραμματισμού* και / ή *επέκταση και οριοθέτηση*) στα αντικείμενα - όρους που βρίσκονται εντός του πυρήνα υπολογίζεται τελικά η επιθυμητή βέλτιστη συνολική αξία. Ορισμένες φορές βέβαια, η βέλτιστη λύση επιτυγχάνεται μόνο με πλήρη απαρίθμηση όλων των εφικτών λύσεων. Για να καταστεί αυτό σαφές, ακολουθεί ένα αριθμητικό παράδειγμα υπολογισμού της βέλτιστης αυτής συνολικής αξίας. Έστω ότι έχουμε

6 αντικείμενα των οποίων η αξία, το βάρος καθώς και ο λόγος $\frac{p_i}{w_i}$ περιέχονται στον

ακόλουθο πίνακα (ταξινομημένα κατά φθίνουσα σειρά, για λόγους που έχουν ήδη αναφερθεί):

i	p_i	w_i	p_i / w_i
1	10	10	1.0
2	35	50	0.7
3	2	5	0.4
4	18	45	0.4
5	40	100	0.4
6	4	20	0.2

Πίνακας 4-2 Αριθμητικό παράδειγμα υπολογισμού βέλτιστης συνολικής αξίας για το 0-1 KP

Επιπρόσθετα, έστω ότι το συνολικό βάρος δε μπορεί να υπερβεί το 100. Ακολουθώντας τις τεχνικές που αναφέρθηκαν παραπάνω, υπολογίζεται αρχικά το άνω και κάτω όριο με *break item* όπως εύκολα μπορεί να υπολογιστεί τον τέταρτο όρο. Αναλυτικά έχουμε:

$$UpperBound = \sum_{i=1}^3 p_i + \frac{p_4}{w_4} (100 - \sum_{i=1}^3 w_i) = 61$$

$$LowerBound = \sum_{i=1}^3 p_i + p_6 = 51$$

Ο υπολογισμός του κάτω ορίου είναι και η βέλτιστη λύση που θα δώσει οποιαδήποτε επιλογή πυρήνα, η οποία δεν περιλαμβάνει τον πρώτο όρο. Πράγματι με απαρίθμηση όλων των δυνατών περιπτώσεων υπολογίζεται ότι στη βέλτιστη λύση – που αποτελείται από το 2^ο, 3^ο

και 4^ο όρο με βέλτιστη συνολική αξία 55 - δε συμμετέχει ο όρος που διαθέτει τον καλύτερο λόγο $\frac{p_i}{w_i}$.

4.4. Multiple Choice Knapsack Problem (MCKP)

Αντικείμενο της παραγράφου αυτής αποτελεί η παρουσίαση των τεχνικών - μεθόδων που εφαρμόζονται κατά την επίλυση του *Multiple Choice Knapsack Problem* (MCKP). Επισημαίνεται – όπως και στην προηγούμενη παράγραφο - ότι οι μέθοδοι αυτές στοχεύουν σε προσεγγιστική και όχι ακριβή λύση του προβλήματος αυτού. Επομένως, οι τεχνικές αυτές μπορούν να υιοθετηθούν από ευρυστικούς (heuristic) αλγορίθμους, αφού το πρόβλημα αυτό ανήκει στην κατηγορία προβλημάτων NP – Δύσκολα (NP – Hard) [83]. Ακριβείς λύσεις που στηρίζονται στις τεχνικές του *δυναμικού προγραμματισμού* (*Dynamic Programming*) και στην *επέκταση και οριοθέτηση* (*Branch and Bound*) παρουσιάζονται μεταξύ άλλων στις εργασίες [91], [107] και [108].

Όπως αναφέρθηκε και σε προηγούμενη παράγραφο (§4.2), στο MCKP υπάρχουν πολλαπλοί σάκοι (N) που διαθέτουν αντικείμενα διαφορετικής αξίας και βάρους. Αυτό που ζητείται είναι η επιλογή ενός ακριβώς αντικειμένου από κάθε έναν από τους σάκους αυτούς, προκειμένου να μεγιστοποιηθεί η συνολική αξία, ενώ το συνολικό βάρος δε θα πρέπει να υπερβαίνει μια προκαθορισμένη τιμή c . Η μαθηματική διατύπωση του προβλήματος αυτού είναι:

$$\begin{aligned} \text{μεγιστοποίηση} \quad & \sum_{i=1}^N \sum_{j=1}^{M_i} p_{ij} * x_{ij} \\ \text{ώστε} \quad & \sum_{i=1}^N \sum_{j=1}^{M_i} w_{ij} * x_{ij} \leq c \\ & \sum_{j=1}^{M_i} x_{ij} = 1 \quad i = 1, 2, \dots, N \\ & x_{ij} \in \{0, 1\} \quad i = 1, 2, \dots, N, \quad j \in M_i \end{aligned}$$

όπου ο δείκτης i παριστάνει τον εκάστοτε σάκο, ενώ ο δείκτης j παριστάνει τα αντικείμενα που βρίσκονται στο σάκο αυτό. Επισημαίνεται ότι ο αριθμός των αντικειμένων που βρίσκονται στον κάθε σάκο είναι διαφορετικός (δηλαδή η τιμή της παραμέτρου M_i είναι διαφορετική για κάθε i). Η μεταβλητή p_{ij} παριστάνει την αξία του αντικειμένου j που βρίσκεται στο σακίδιο i , ενώ η μεταβλητή w_{ij} παριστάνει το βάρος του. Τέλος, η μεταβλητή

x_{ij} υποδεικνύει αν το αντικείμενο j που βρίσκεται στο σακίδιο i επιλέγεται ή όχι. Σε περίπτωση που επιλέγεται, παίρνει την τιμή 1, αλλιώς την τιμή 0.

Η επίλυση του $MCKP$ στηρίζεται στην επίλυση του $Linear\ MCKP\ \{LMCKP\}$. Το πρόβλημα αυτό διαφοροποιείται σε σχέση με το $MCKP$ στο ότι η μεταβλητή x_{ij} μπορεί να επιλεγεί και «μερικώς», επομένως το πεδίο τιμών της είναι το διάστημα $0 \leq x_{ij} \leq 1$ όπως και στην περίπτωση του $0-1\ KP$. Η επίλυση του $LMCKP$ στηρίζεται στα ακόλουθα βήματα: Αρχικά, εφαρμόζονται σε κάθε σάκο 2 κριτήρια επικράτησης (*dominance criteria*) [109]:

1. Όταν σε δύο αντικείμενα r, s του σακιδίου i ισχύει η σχέση:

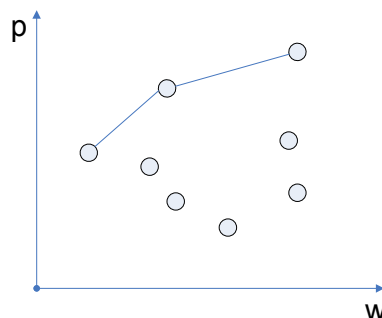
$$w_{ir} \leq w_{is} \text{ και } p_{ir} \geq p_{is}$$

τότε όπως είναι αναμενόμενο, μπορεί να αγνοηθεί ο όρος s του σακιδίου i , δηλαδή $x_{is} = 0$

2. Όταν σε τρία αντικείμενα r, s, t του σακιδίου i ισχύει η σχέση:

$$\frac{p_{is} - p_{ir}}{w_{is} - w_{ir}} \leq \frac{p_{it} - p_{is}}{w_{it} - w_{is}}$$

τότε μπορεί να αγνοηθεί ο όρος του σακιδίου i , δηλαδή $x_{is} = 0$. Το κριτήριο αυτό ουσιαστικά απομακρύνει τους όρους οι οποίοι βρίσκονται εντός της καμπύλης, όπως φαίνεται και στο παρακάτω σχήμα:



Σχήμα 4-1 Όροι που απομακρύνονται στο $LMCKP$ σύμφωνα με το 2^ο κριτήριο επικράτησης

Στη συνέχεια, η επίλυση του $LMCKP$ συνεχίζει ως εξής:

Επιλέγεται από το κάθε σακίδιο i , το αντικείμενο με το μικρότερο βάρος w_{ij} και σχηματίζεται μια αρχική λύση έστω $W - P$, όπου W το συνολικό βάρος των επιλεγμένων αντικειμένων, ενώ P το συνολικό κέρδος των αντικειμένων αυτών. Είναι εύλογο ότι για να έχει λύση το πρόβλημα αυτό – όπως και το $MCKP$ – δε θα πρέπει το συνολικό βάρος των αντικειμένων αυτών να ξεπερνά τη συνολική χωρητικότητα c .

Κατόπιν, υπολογίζεται ο λόγος $\lambda_{ij} = \frac{p_{ij} - p_{ij-1}}{w_{ij} - w_{ij-1}}$ (slope - κλίση) για όλα τα αντικείμενα

όλων των σακιδίων (εκτός από αυτά με το μικρότερο βάρος) και ταξινομούνται όλα τα λ_{ij} σε φθίνουσα σειρά. Επισημαίνεται ότι σε περίπτωση που βρεθούν αντικείμενα με ίδια κλίση λ_{ij} προτεραιότητα έχει αυτό με τη μεγαλύτερη αξία p_{ij} .

Στη συνέχεια βελτιώνεται διαρκώς το συνολικό κέρδος P με τη διαδοχική επιλογή των λ_{ij} .

Στο βήμα αυτό γίνονται αμοιβαίες αντικαταστάσεις όρων με αυτούς που βρίσκονται στο εκάστοτε σακίδιο i , λαμβάνοντας πάντα υπόψη τον περιορισμό c . Τα νέα W' - P' ορίζονται ως:

$$W' = W + w_{ij} - w_{ij-1} \leq c$$

$$P' = P + p_{ij} - p_{ij-1}$$

Η επαναληπτική αυτή διαδικασία τερματίζει όταν βρεθεί ο πρώτος όρος που ξεπερνά τη συνολική χωρητικότητα c . Επισημαίνεται ότι σε περίπτωση όπου:

$$W' = c$$

τότε η λύση που βρέθηκε είναι η βέλτιστη όχι μόνο για το $LMCKP$ αλλά και για το $MCKP$. Σε διαφορετική περίπτωση και λαμβάνοντας υπόψη τη συνθήκη χαλάρωσης για τη μεταβλητή x_{ij} ($0 \leq x_{ij} \leq 1$), βρίσκεται η βέλτιστη λύση για το $LMCKP$, η οποία θα αποτελέσει το άνω όριο της $MCKP$. Όπως είναι αναμενόμενο στη λύση αυτή θα συμμετέχουν «μερικώς» δύο όροι $j - j-1$ ενός σακιδίου έστω i , με συντελεστές $x_{ij} - x_{ij-1}$. Οι συντελεστές αυτοί προφανώς θα πρέπει να δίνουν άθροισμα 1 (αφού θα πρέπει υποχρεωτικά ο αριθμός των αντικειμένων τελικά να είναι ακέραιος αριθμός) με τιμές που θα πρέπει να πληρούν τις σχέσεις:

$$W' = W - w_{ij-1} + x_{ij} * w_{ij} + x_{ij-1} * w_{ij-1} = c$$

$$x_{ij} + x_{ij-1} = 1$$

$$P' = P - p_{ij-1} + x_{ij} * p_{ij} + x_{ij-1} * p_{ij-1}$$

Επομένως, κάνοντας τις απαραίτητες πράξεις, καταλήγουμε στο ότι:

$$x_{ij} = \frac{c - W}{w_{ij} - w_{ij-1}}$$

$$x_{ij-1} = 1 - x_{ij}$$

Λαμβάνοντας υπόψη τους παραπάνω συντελεστές, η τελική λύση του $LMCKP$ είναι η ακόλουθη:

$$W' = c$$

$$P' = P + (c - W)\lambda_{ij}$$

Το κέρδος αυτό, όπως ήδη αναφέρθηκε, αποτελεί το άνω όριο του κέρδους που μπορεί να υπολογιστεί από το *MCKP*. Χρησιμοποιώντας τους όρους που συμμετέχουν στο όριο αυτό, ορίζοντας διάφορα μεγέθη πυρήνων, και χρησιμοποιώντας διάφορες τεχνικές (*άπληστες (greedy)*, *δυναμικού προγραμματισμού (dynamic programming)*, *επέκταση και οριοθέτηση (branch and bound)*), επιτυγχάνεται η επίλυση του *MCKP* όπως και στην περίπτωση του *0-1 KP*. Ενδεικτικά αναφέρονται οι εργασίες [109], [110] και [111].

Ακολουθούν αριθμητικά παραδείγματα υπολογισμού του συνολικού κέρδους στο *LMCKP* για διάφορες τιμές της συνολικής χωρητικότητας c .

Έστω ότι υπάρχουν 2 σάκοι με 3 αντικείμενα ο κάθε ένας από αυτούς (μετά την εφαρμογή των 2 κριτηρίων επικράτησης). Η αξία και το βάρος του κάθε αντικειμένου περιλαμβάνονται στον πίνακα που ακολουθεί:

i	j	p _i	w _i	λ _{ij}
1	1	3	1	-
1	2	5	2	2
1	3	8	4	1.5
2	1	6	2	2
2	2	4	1	-
2	3	7	3	1

Πίνακας 4-3 Αριθμητικό παράδειγμα υπολογισμού βέλτιστης συνολικής αξίας για το *MCKP*

Ακολουθώντας τα βήματα που αναφέρθηκαν παραπάνω κατά την επίλυση του *LMCKP*, επιλέγονται τα ελαφρύτερα αντικείμενα, οπότε από τον πρώτο σάκο επιλέγεται το πρώτο αντικείμενο (με βάρος 1) και από το δεύτερο σάκο το δεύτερο αντικείμενο (με βάρος επίσης 1). Είναι εύλογο ότι για να επιδέχεται το πρόβλημα αυτό λύση θα πρέπει η ελάχιστη μέγιστη χωρητικότητα να υπερβαίνει την τιμή 2. Κατόπιν διατάσσονται οι όροι που αφορούν την κλίση σε φθίνουσα σειρά, ως εξής:

$$\lambda_{2,1-2} \geq \lambda_{1,2-1} \geq \lambda_{1,3-2} \geq \lambda_{2,3-1}$$

Διακρίνονται οι ακόλουθες δύο περιπτώσεις όσον αφορά τη συνολική χωρητικότητα:

$$(i) c = 6$$

Κάνοντας αμοιβαίες αντικαταστάσεις υπολογίζεται το μέγιστο κέρδος του *LMCKP*, το οποίο στην προκειμένη περίπτωση είναι ταυτόσημο με αυτό του *MCKP* και ισούται με 14.

i	W	P
1	2	7
2	3	9
3	4	11
4	6	14

Πίνακας 4-4 Αριθμητικό παράδειγμα υπολογισμού βέλτιστης συνολικής αξίας για το *MCKP* με χωρητικότητα $c = 6$

(ii) $c = 5$

Σε αυτή την περίπτωση, εκτός από τις αμοιβαίες αντικαταστάσεις είναι απαραίτητο να επιλεγθούν «μερικώς» το δεύτερο και τρίτο αντικείμενο του πρώτου σακιδίου. Κάνοντας τις απαραίτητες πράξεις προκύπτει ότι οι συντελεστές είναι:

$$x_{13} = x_{23} = 0.5$$

Το βέλτιστο κέρδος που υπολογίζεται για το *LMCKP* ισούται όπως φαίνεται και στον πίνακα που ακολουθεί με 12.5. Η τιμή αυτή αποτελεί το άνω όριο του *MCKP*, όπως έχει ήδη αναφερθεί.

i	W	P
1	2	7
2	3	9
3	4	11
4	5	12.5

Πίνακας 4-5 Αριθμητικό παράδειγμα υπολογισμού βέλτιστης συνολικής αξίας για το *MCKP* με χωρητικότητα $c = 5$

=====

Κεφάλαιο 5ο

5. Σχεδιασμός αλγορίθμων επιλογής υπηρεσιών με στόχο τη βελτιστοποίηση του κέρδους

=====

Στο κεφάλαιο αυτό θα προταθούν κατάλληλοι αλγόριθμοι, οι οποίοι μπορούν να χρησιμοποιηθούν από έναν πάροχο υπηρεσιών για την επιλογή των καταλληλότερων υπηρεσιών που εκείνος προσφέρει, προκειμένου να εξυπηρετήσει τους πελάτες του. Όπως αναφέρθηκε και στο κεφάλαιο 1, αυτό είναι το τελευταίο βήμα που πρέπει να ακολουθηθεί προκειμένου να λυθεί το γενικό πρόβλημα της επιλογής διαδικτυακών υπηρεσιών στο σημασιολογικό δίκτυο. Πιο συγκεκριμένα, το πρόβλημα που θα αναλυθεί, σχετίζεται με την επιλογή των καταλληλότερων υπηρεσιών που προσφέρει ένας πάροχος με στόχο τη μεγιστοποίηση του κέρδους του. Η επιλογή αυτή θα στηρίζεται σε κριτήρια που αφορούν τις παραμέτρους ποιότητας των υπηρεσιών (κεφάλαιο 3) διακρίνοντας διαφορετικά μοντέλα ανάλογα με την παράμετρο - περιορισμό επιλογής του παρόχου. Έχοντας επιλέξει την παράμετρο αυτή, ο πάροχος καλείται στη συνέχεια να επιλέξει τις υπηρεσίες που θα προσφέρει στους πελάτες του. Επισημαίνεται ότι επειδή εξετάζεται το πρόβλημα από τη μεριά του παρόχου, οι αλγόριθμοι επιλογής υπηρεσιών στοχεύουν στη μεγιστοποίηση του κέρδους από τη μεριά αυτή, χωρίς να είναι απαραίτητο να εξυπηρετηθούν όλοι οι πελάτες του. Εξάλλου η μοντελοποίηση αυτή δεν εστιάζει στην εύρεση των υπηρεσιών που ικανοποιούν τον κάθε πελάτη, αφού όπως αναφέρθηκε και στο κεφάλαιο 1, αυτή είναι μια διαδικασία διαδοχικού φιλτραρίσματος, η οποία δεν αποτελεί αντικείμενο ενασχόλησης της παρούσας διατριβής. Αντίθετα, οι αλγόριθμοι επιλογής παραμέτρων ποιότητας και υπηρεσιών που παρουσιάζονται στο κεφάλαιο αυτό επικεντρώνονται στο βήμα που ακολουθεί τη διαδικασία αυτή. Δηλαδή, διαθέτοντας ένα σύνολο από υπηρεσίες που ικανοποιούν τον κάθε πελάτη χωριστά και έχοντας επιλέξει την παράμετρο ποιότητας που

επιθυμεί να εξετάσει ο πάροχος, να επιλεγθεί η πιο κατάλληλη από τις προσφερόμενες υπηρεσίες για κάθε έναν πελάτη, προκειμένου να μεγιστοποιηθεί το συνολικό κέρδος του παρόχου. Βέβαια σε περίπτωση που ο πάροχος διαθέτει πολλαπλούς περιορισμούς όσον αφορά τις παραμέτρους αυτές, θα πρέπει να ακολουθηθεί κάποια διαφορετική διαδικασία. Μια τέτοια διαδικασία προτείνεται και στο κεφάλαιο αυτό.

Το κεφάλαιο οργανώνεται ως εξής: Στην πρώτη παράγραφο γίνεται η μοντελοποίηση του προβλήματος που θα μας απασχολήσει, ενώ στην παράγραφο 5.2 παρουσιάζονται διάφορα πρακτικά προβλήματα της καθημερινότητας που μπορούν να αντιμετωπιστούν με τις τεχνικές που προτείνονται στο κεφάλαιο αυτό. Στην επόμενη παράγραφο (§5.3) γίνεται αναφορά και ανάλυση σε ανάλογες εργασίες της παγκόσμιας βιβλιογραφίας. Στην παράγραφο 5.4 εξετάζονται οι παράμετροι ποιότητας υπηρεσίας που περιλαμβάνονται στην οντολογία Λεξιλόγιο (§3.3.3) και μπορεί να εφαρμοστεί ο προτεινόμενος αλγόριθμος. Αντικείμενο της παραγράφου §5.5 είναι η ανάλυση του προτεινόμενου αλγορίθμου, μελετώντας 3 διαφορετικές παραμέτρους ποιότητας υπηρεσίας. Η περίπτωση που κάποιος πάροχος έχει πολλαπλούς περιορισμούς όσον αφορά τους διαθέσιμους πόρους του εξετάζεται στην παράγραφο 5.6, ενώ το κεφάλαιο ολοκληρώνεται στην παράγραφο 5.7 με την αποτίμηση της επίδοσης του προτεινόμενου αλγορίθμου επιλογής υπηρεσιών, η οποία συνοδεύεται με πειραματικές μετρήσεις και σχετικά διαγράμματα. Τέλος, στο Παράρτημα Β περιέχεται όλος ο πηγαίος κώδικας του αλγορίθμου επιλογής υπηρεσιών. Σημειώνεται, ότι ως εργαλείο ανάπτυξης και εκτέλεσης του αλγορίθμου αυτού χρησιμοποιήθηκε το MATLAB v7.0 [112].

5.1. Μοντελοποίηση προβλήματος

Η ανάπτυξη και εξάπλωση του διαδικτύου, όπως έχει ήδη αναφερθεί σε προηγούμενο κεφάλαιο, οδήγησε στη δημιουργία διαδικτυακών υπηρεσιών (web services) με πλήθος εφαρμογών σε όλους τους τομείς. Αυτή η εξάπλωση γρήγορα συνοδεύτηκε και από τη δημιουργία αντιστοίχου πλήθους παρόχων υπηρεσιών που ικανοποιούν ακόμα και τις πιο ιδιαίτερες προτιμήσεις και απαιτήσεις προσφέροντας ανταγωνιστικές και ελκυστικές υπηρεσίες. Απώτερος στόχος και σκοπός της προσφοράς αυτής βέβαια είναι από τη μεριά των παρόχων, να βελτιώσουν την ανταγωνιστικότητά τους, ενισχύοντας κατ' αυτόν τον τρόπο τα κέρδη τους. Προκειμένου να επιτευχθεί αυτό και λαμβάνοντας υπόψη τις παραμέτρους ποιότητας των υπηρεσιών που προσφέρουν (κεφάλαιο 3), είναι απαραίτητο να σχεδιάσουν τα συστήματά τους κατά τέτοιο τρόπο, ώστε όχι μόνο να αντιμετωπίσουν τα προβλήματα που αναλύθηκαν εκτενώς σε προηγούμενο κεφάλαιο, αλλά επιπρόσθετα να αντιμετωπίσουν προβλήματα που σχετίζονται με τους πόρους των συστημάτων που διαθέτουν. Τέτοια προβλήματα είναι απόρροια του γεγονότος ότι οι υπολογιστικοί πόροι όλων των συστημάτων, όσο μεγάλοι και να είναι, ωστόσο δεν είναι ανεξάντλητοι.

Καθίσταται αναγκαίο και απαραίτητο επομένως να λαμβάνονται μέτρα, προκειμένου να αντιμετωπιστούν πιθανά προβλήματα που προκαλούνται λόγω εξάντλησης των πόρων αυτών. Χαρακτηριστικό παράδειγμα που αποτελεί και το αντικείμενο του κεφαλαίου αυτού, είναι η περίπτωση επιλογής υπηρεσιών / πελατών από τη μεριά του παρόχου. Αυτό συμβαίνει σε περιπτώσεις όπου ένας πάροχος διαθέτει υπηρεσίες που το συνολικό άθροισμα κάποιας παραμέτρου ποιότητας (σε περίπτωση που αναφερόμαστε σε γραμμική παράμετρο όπως για παράδειγμα η *Χωρητικότητα*) των υπηρεσιών που μπορεί να διαθέσει ταυτόχρονα στους πελάτες του δε μπορεί να υπερβαίνει κάποια προκαθορισμένη τιμή. Σε αυτή την περίπτωση είναι εύλογο ότι ο πάροχος καλείται να επιλέξει τις κατάλληλες υπηρεσίες που θα προσφέρει / πελάτες που θα εξυπηρετήσει, προκειμένου να μεγιστοποιήσει το κέρδος του. Το πρόβλημα γίνεται ακόμα πιο σύνθετο, σε περίπτωση όπου ο πάροχος καλείται να επιλέξει από μια ομάδα από υπηρεσίες που ικανοποιούν τον κάθε πελάτη χωριστά, ενώ είναι πιθανό ο κάθε πελάτης να έχει πολλαπλές / διαφορετικές απαιτήσεις οι οποίες ικανοποιούνται προφανώς από διαφορετικές τέτοιες ομάδες υπηρεσιών. Υπενθυμίζεται, ότι το πρόβλημα διερευνάται από τη σκοπιά της επιλογής των καταλληλότερων υπηρεσιών που θα ενισχύσουν τα κέρδη των παρόχων και όχι της εύρεσης των υπηρεσιών αυτών. Η διαδικασία αυτή αποτελεί και το τελευταίο βήμα που πρέπει να ακολουθηθεί, προκειμένου να επιλυθεί το γενικό πρόβλημα της επιλογής υπηρεσιών στο σημασιολογικό δίκτυο (κεφάλαιο 1).

Στην παρούσα μοντελοποίηση έχουν χρησιμοποιηθεί 2 παράμετροι ποιότητας υπηρεσίας (κεφάλαιο 3): το *Κόστος* της υπηρεσίας (που ταυτίζεται με το κέρδος του παρόχου) καθώς και μια γενική παράμετρος ποιότητας (*Χωρητικότητα*, *Ρυθμός Δεδομένων* κτλ), η οποία θα συμπεριληφθεί στη συνθήκη περιορισμού. Επισημαίνεται ότι αυτή η παράμετρος περιλαμβάνεται στην οντολογία Λεξιλόγιο (§3.3.3), διακρίνοντας διαφορετικές μοντελοποιήσεις στη συνθήκη περιορισμού ανάλογα με το αν είναι γραμμική ή μη γραμμική παράμετρος, όπως άλλωστε θα φανεί και στη συνέχεια.

Το πρόβλημα το οποίο καλείται να λύσει ο προτεινόμενος αλγόριθμος είναι το εξής: Δοθέντος N πελάτες, K αιτήσεις από τον κάθε πελάτη που ουσιαστικά μοντελοποιεί την περίπτωση που κάποιος πελάτης επιθυμεί κάποια υπηρεσία με διαφορετικές τιμές στις παραμέτρους ποιότητας (και σε διαφορετικό *Κόστος* προφανώς), και $N \times K$ ομάδες από υπηρεσίες οι οποίες διατίθενται από έναν πάροχο και ικανοποιούν την αίτηση του κάθε πελάτη, να βρεθεί η υπηρεσία που πρέπει να επιλεγεί για κάθε έναν από αυτούς (δεν είναι απαραίτητο να εξυπηρετηθούν όλοι οι πελάτες) ώστε να μεγιστοποιηθεί το κέρδος για τον πάροχο. Επισημαίνεται, ότι οι ομάδες των υπηρεσιών που ικανοποιούν τις αιτήσεις είναι διαφορετικές μεταξύ τους. Υπάρχει μια συνθήκη περιορισμού (π.χ. *Χωρητικότητα*, *Ρυθμός Δεδομένων*) για την οποία ισχύει το εξής: το άθροισμα της αντίστοιχης παραμέτρου των υπηρεσιών που θα επιλεγθούν τελικά, να μην υπερβαίνει ποτέ την τιμή της (σε περίπτωση

που αναφερόμαστε σε γραμμική παράμετρο). Η μαθηματική διατύπωση και μοντελοποίηση του παραπάνω προβλήματος είναι η ακόλουθη:

$$\text{μεγιστοποίηση} \quad \sum_{i=1}^N \sum_{k=1}^{L_i} \sum_{j=1}^{M_{ik}} p_{ikj} * S_{ikj} \quad (5-1)$$

$$\text{ώστε} \quad \sum_{i=1}^N \sum_{k=1}^{L_i} \sum_{j=1}^{M_{ik}} w_{ikj} * S_{ikj} \leq c \quad (5-2)$$

$$\sum_{k=1}^{L_i} \sum_{j=1}^{M_{ik}} S_{ikj} \leq 1 \quad i = 1, 2, \dots, N \quad (5-3)$$

$$S_{ikj} \in \{0, 1\} \quad \begin{array}{l} i = 1, 2, \dots, N, \\ k = 1, 2, \dots, L_i, \\ j = 1, 2, \dots, M_{ik} \end{array} \quad (5-4)$$

όπου ο δείκτης i παριστάνει τον εκάστοτε πελάτη, ο δείκτης k παριστάνει την εκάστοτε αίτηση του πελάτη, ενώ ο δείκτης j παριστάνει την υπηρεσία που ικανοποιεί την εκάστοτε αίτηση. Επιπρόσθετα, ο αριθμός των διαφορετικών «συμβολαίων» που αιτείται ο κάθε πελάτης δεν είναι ο ίδιος (δηλαδή η τιμή της παραμέτρου L_i στις σχέσεις (5-1), (5-2), (5-3) και (5-4) είναι διαφορετική για διαφορετικό i), ενώ το ίδιο ισχύει και για το πλήθος των υπηρεσιών που ικανοποιούν την κάθε αίτηση (δηλαδή η τιμή της παραμέτρου M_{ik} στις σχέσεις (5-1), (5-2), (5-3) και (5-4) είναι διαφορετική για διαφορετικά i, k). Ακόμη, η μεταβλητή p_{ikj} παριστάνει το κέρδος του παρόχου από την εκάστοτε υπηρεσία, ενώ η μεταβλητή w_{ikj} παριστάνει την «επιβάρυνση» από την επιλογή της συγκεκριμένης υπηρεσίας. Η μεταβλητή S_{ikj} υποδεικνύει αν η υπηρεσία j που ικανοποιεί την αίτηση k του πελάτη i επιλέγεται ή όχι. Σε περίπτωση που επιλέγεται, παίρνει την τιμή 1, αλλιώς την τιμή 0 (σχέση (5-4)). Είναι εύλογο ότι ο κάθε πελάτης μπορεί να εξυπηρετηθεί για το σύνολο των «συμβολαίων» του από μια μόνο υπηρεσία, παρατήρηση η οποία είναι πολύ σημαντική, όπως θα φανεί και παρακάτω. Εξάλλου όπως έχει ήδη αναφερθεί, ο προτεινόμενος αλγόριθμος δε στοχεύει στην ικανοποίηση όλων των πελατών (σχέση (5-3)), αλλά στη μεγιστοποίηση του κέρδους από τη μεριά του παρόχου (σχέση (5-1)). Τέλος, όπως φαίνεται και στη σχέση (5-2) το άθροισμα της παραμέτρου περιορισμού των επιλεγμένων υπηρεσιών δεν υπερβαίνει μια προκαθορισμένη τιμή c . Σε αυτή τη μοντελοποίηση βέβαια έχει θεωρηθεί «σιωπηρά» ότι αυτή η παράμετρος είναι γραμμική. Όπως θα φανεί και στη συνέχεια όμως, αυτό δεν ισχύει πάντα (όπως για παράδειγμα στην περίπτωση της *Καθυστέρησης*) και διαφορετικές μοντελοποιήσεις θα πρέπει να προταθούν, όσον αφορά τη συνθήκη περιορισμού.

Εξετάζοντας προσεκτικά την παρούσα μοντελοποίηση μπορεί να διαπιστώσει κανείς ότι είναι εύκολη η αναγωγή της σε κάποια απλούστερη με τη δημιουργία νέων ομάδων υπηρεσιών για τον κάθε πελάτη. Πράγματι, εφόσον ο κάθε πελάτης μπορεί να εξυπηρετηθεί για το σύνολο των «συμβολαίων» του από μια μόνο υπηρεσία, τότε αν δημιουργηθούν νέες ομάδες από υπηρεσίες, όπου στην κάθε ομάδα θα περιλαμβάνονται οι υπηρεσίες όλων των «συμβολαίων» του κάθε πελάτη, απομακρύνοντας τις πολλαπλές εμφανίσεις της ίδιας υπηρεσίας, τότε εύκολα μετασχηματίζεται η αρχική μοντελοποίηση στην ακόλουθη ισοδύναμή της:

$$\text{μεγιστοποίηση} \quad \sum_{i=1}^N \sum_{j=1}^{M_i} p_{ij} * S_{ij} \quad (5-5)$$

$$\text{ώστε} \quad \sum_{i=1}^N \sum_{j=1}^{M_i} w_{ij} * S_{ij} \leq c \quad (5-6)$$

$$\sum_{j=1}^{M_i} S_{ij} \leq 1 \quad i = 1, 2, \dots, N \quad (5-7)$$

$$S_{ij} \in \{0, 1\} \quad i = 1, 2, \dots, N, \quad j \in M_i \quad (5-8)$$

Η μόνη διαφοροποίηση όσον αφορά τις μεταβλητές σε σχέση με την προηγούμενη μοντελοποίηση είναι η μεταβλητή M_i , η οποία χρησιμοποιείται για να αναπαραστήσει το πλήθος των διαφορετικών υπηρεσιών που ικανοποιούν τον κάθε πελάτη, μετά τη συγχώνευση και την απομάκρυνση – ενδεχομένως – των ίδιων υπηρεσιών που ικανοποιούν τα διαφορετικά «συμβολαία» του. Εξάλλου όπως είναι αναμενόμενο έχει αφαιρεθεί από όλες τις μεταβλητές ο δείκτης k , αφού στη νέα μοντελοποίηση δεν υπάρχουν διαφορετικά «συμβολαία» για τον κάθε πελάτη. Αξίζει να αναφερθεί ότι σε περίπτωση που η σχέση (5-7) είχε ισότητα αντί για ανίσωση, δηλαδή αν έπρεπε να ικανοποιηθούν όλες οι αιτήσεις-πελάτες, τότε το πρόβλημα είναι το ΜСKР [95] (σε περίπτωση βέβαια που η τιμή της επιλεγμένης παραμέτρου ποιότητας υπηρεσίας μεταβάλλεται γραμμικά καθώς αυξάνεται το πλήθος των αιτήσεων). Σε αυτή την παρατήρηση, καθώς και στους τρόπους επίλυσης των προβλημάτων που ανήκουν γενικότερα στην οικογένεια προβλημάτων του σακιδίου (knapsack family) και μελετήθηκαν εκτενώς στο κεφάλαιο 4, θα στηριχθεί ο αλγόριθμος του κεφαλαίου αυτού. Βέβαια, σε περίπτωση που η παράμετρος ποιότητας υπηρεσίας που συμμετέχει στη συνθήκη περιορισμού δε μεταβάλλεται γραμμικά, τότε νέα τεχνάσματα πρέπει να «επινοηθούν» τα οποία θα οδηγήσουν στον υπολογισμό του επιθυμητού κέρδους.

5.2. Εφαρμογή προτεινόμενων τεχνικών

Το εύρος των προβλημάτων που μπορούν να αντιμετωπιστούν με τις τεχνικές του κεφαλαίου αυτού είναι αρκετά μεγάλο, βρίσκοντας εφαρμογή σε διάφορους τομείς. Η

εξαντλητική απαρίθμηση όλων αυτών των περιπτώσεων ξεφεύγει από τα πλαίσια της παρούσας διατριβής. Ωστόσο, ορισμένα παραδείγματα που αντικατοπτρίζουν συχνά αντιμετωπίσιμα προβλήματα της καθημερινότητας και μπορούν να εφαρμοστούν οι τεχνικές αυτές είναι τα ακόλουθα:

Ένα πρώτο παράδειγμα είναι η περίπτωση που εξετάζεται στο κεφάλαιο αυτό με έναν πάροχο τηλεπικοινωνιακών υπηρεσιών, ο οποίος καλείται να επιλέξει τους πελάτες που θα εξυπηρετήσει και τις υπηρεσίες που θα τους προσφέρει, προκειμένου να μεγιστοποιήσει το κέρδος του. Πράγματι, έστω N διαφορετικοί πελάτες, οι οποίοι ικανοποιούνται από μια ομάδα υπηρεσιών. Η προσφορά της κάθε υπηρεσίας έχει ένα συγκεκριμένο κέρδος για τον πάροχο και επιβαρύνει τους διαθέσιμους πόρους του κατά ένα ποσό. Η εφαρμογή των προτεινόμενων τεχνικών θα τον βοηθήσει να επιλέξει ποιους πελάτες πρέπει να εξυπηρετήσει και ποιες υπηρεσίες να τους διαθέσει, ώστε να έχει το μέγιστο κέρδος και να μην ξεπερνά τον περιορισμό των διαθέσιμων πόρων του.

Ένα άλλο παράδειγμα είναι η περίπτωση επιλογής επενδυτικών σχεδίων που πρέπει να ακολουθήσει μια επιχείρηση. Πράγματι έστω ότι μια επιχείρηση στοχεύει να δραστηριοποιηθεί σε N διαφορετικούς τομείς. Έστω ακόμη ότι για τον κάθε τομέα υπάρχουν διαφορετικό πλήθος επενδυτικών σχεδίων που μπορούν να εφαρμοστούν, όπου το κάθε «σχέδιο» χαρακτηρίζεται από ένα υποσχόμενο «κέρδος» και η εφαρμογή του απαιτεί ένα ποσό επένδυσης. Λαμβάνοντας τότε υπόψη το γεγονός ότι το σύνολο των διαθέσιμων χρημάτων για τη συνολική επένδυση δε μπορεί να ξεπερνά ένα συγκεκριμένο πόσο, οι προτεινόμενες τεχνικές μπορούν να τη βοηθήσουν στο να αποφασίσει σε ποιους τομείς πρέπει να δραστηριοποιηθεί και με ποια επενδυτικά σχέδια, ώστε να μεγιστοποιήσει τα μελλοντικά «υποσχόμενα» κέρδη της.

Αλλά και στη ναυτιλία μπορούν να εφαρμοστούν οι προτεινόμενες τεχνικές εξασφαλίζοντας μεγαλύτερο κέρδος για τις διάφορες ναυτιλιακές εταιρείες. Πράγματι, έστω ότι κάποια ναυτιλιακή εταιρεία διαθέτει στο στόλο της N διαφορετικά πλοία. Ακόμη, έστω ότι σε κάθε πλοίο μπορεί να φορτώσει διαφορετικά υλικά / πρώτες ύλες, όπου ανάλογα με την επιλογή υλικού θα έχει και διαφορετικό κέρδος καθώς και διαφορετικά συνολικά έξοδα (ναύλωσης και υλικού). Εφαρμόζοντας τότε τις τεχνικές του κεφαλαίου αυτού, μπορεί να επιλέξει ποια πλοία πρέπει να χρησιμοποιήσει και με ποιο υλικό να τα φορτώσει, ώστε να έχει το μέγιστο δυνατό κέρδος και να μην ξεπερνά το σύνολο των εξόδων ένα συγκεκριμένο ποσό.

Ένα πρόβλημα της καθημερινότητας που μπορεί να αντιμετωπιστεί με τις τεχνικές του κεφαλαίου αυτού είναι το πρόβλημα της επιλογής τροφών και γευμάτων που μπορεί να καταναλώσει ένας άνθρωπος σε μια μέρα. Έστω N διαφορετικά γεύματα που μπορεί να καταναλώσει και έστω ότι στο κάθε γεύμα περιλαμβάνονται τροφές διαφορετικής θρεπτικής

αξίας και θερμίδων. Η εφαρμογή των προτεινόμενων τεχνικών θα τον βοηθήσει να αποφασίσει ποια γεύματα πρέπει να καταναλώσει και με ποια τροφή, ώστε να τραφεί όσο το δυνατόν πιο υγιεινά, χωρίς να ξεπεράσει ένα συγκεκριμένο αριθμό θερμίδων που επιθυμεί.

5.3. Σχετική βιβλιογραφία

Σε αυτήν την παράγραφο θα αναλυθούν εργασίες, που έχουν γίνει στην παγκόσμια βιβλιογραφία, σχετικές με το πρόβλημα που παρουσιάστηκε στην προηγούμενη παράγραφο, καθώς και τρόποι αντιμετώπισης που προτάθηκαν από τις αντίστοιχες έρευνες. Επισημαίνεται, ότι η παράγραφος αυτή λειτουργεί συμπληρωματικά με το προηγούμενο κεφάλαιο που παρουσιάζονται τα προβλήματα και οι τρόποι αντιμετώπισής προβλημάτων που ανήκουν στην οικογένεια προβλημάτων του σακιδίου (knapsack family) και κρίνεται σκόπιμο επομένως να χρησιμοποιείται σαν αναφορά αυτό, όπου είναι απαραίτητο. Η βιβλιογραφική ανάλυση που ακολουθεί περιλαμβάνει αρχιτεκτονικές και αλγορίθμους επιλογής υπηρεσιών που επιλύουν συγγενικά προβλήματα, ενώ τα προβλήματα αυτά διαφοροποιούνται από το πρόβλημα του κεφαλαίου αυτού ως προς το γεγονός ότι οι αλγόριθμοι που προτείνουν (κάνοντας την απαραίτητη αναγωγή, όπως θα φανεί και παρακάτω) εφαρμόζονται μόνο σε περιπτώσεις που εξυπηρετούνται όλοι οι πελάτες, χωρίς να δίνεται η δυνατότητα απόρριψης ορισμένων από αυτών. Πράγματι, τα προβλήματα που επιλύουν οι αλγόριθμοι της παγκόσμιας βιβλιογραφίας που ακολουθούν, σχετίζονται με τη διαδικασία σύνθεσης πολύπλοκων διαδικτυακών υπηρεσιών. Η διαδικασία αυτή περιλαμβάνει σε κάθε στάδιο την επιλογή μιας υπηρεσίας από ένα σύνολο υπηρεσιών, με την τελική σύνθετη υπηρεσία να μεγιστοποιεί το κέρδος ή μια συνάρτηση με μια ή περισσότερες συνθήκες περιορισμού, οι οποίες περιλαμβάνουν μια ή περισσότερες παραμέτρους ποιότητας. Θεωρώντας το κάθε στάδιο της σύνθεσης αυτής σαν ένα πελάτη που πρέπει να εξυπηρετήσει το πρόβλημα του κεφαλαίου αυτού, εύκολα μπορεί να γίνει η αναγωγή.

Οι Tao Yu και Kwei-Jay Lin έχουν προτείνει αλγορίθμους για την επιλογή διαδικτυακών υπηρεσιών (web services) που θα οδηγήσουν στη σύνθεση νέων και πιο πολύπλοκων υπηρεσιών με μία ή περισσότερες συνθήκες ελέγχου αντίστοιχα ([113], [114]). Η προτεινόμενη αρχιτεκτονική και στις δύο αυτές εργασίες στηρίζεται στη χρήση ενός μεσίτη (broker) ποιότητας υπηρεσίας, ο οποίος αφενός είναι υπεύθυνος για την ορθή λειτουργία του αλγορίθμου και την κατάλληλη επιλογή της υπηρεσίας σε κάθε βήμα και αφετέρου συντελεί στη δόμηση της σύνθετης υπηρεσίας βάσει των υπηρεσιών που έχουν ήδη επιλεγεί. Στόχος των προτεινόμενων αλγορίθμων είναι η μεγιστοποίηση μιας συνάρτησης, η οποία λαμβάνει υπόψη τις προτιμήσεις του χρήστη, χωρίς να παραβιάζει ωστόσο τους περιορισμούς που αφορούν τις παραμέτρους ποιότητας από άκρη σε άκρη. Όσον αφορά τους τρόπους επιλογής των υπηρεσιών, 2 διαφορετικές προσεγγίσεις ακολουθήθηκαν. Η πρώτη μοντελοποιεί το

πρόβλημα σαν πρόβλημα γραμμικού προγραμματισμού [82], ενώ η δεύτερη το μοντελοποιεί σαν πρόβλημα θεωρίας γράφων [115].

Πιο αναλυτικά, ο μεσίτης ποιότητας υπηρεσίας στον οποίο στηρίζεται η προτεινόμενη αρχιτεκτονική, λειτουργεί ως μεσολαβητής μεταξύ του παρόχου των υπηρεσιών και των πελατών – χρηστών της υπηρεσίας. Ο ρόλος του είναι πολλαπλός. Δέχεται τις αιτήσεις από τους χρήστες – πελάτες για εύρεση κατάλληλων υπηρεσιών. Οι αιτήσεις αυτές περιέχουν εκτός από το είδος της υπηρεσίας που ζητείται και μία ή περισσότερες παραμέτρους ποιότητας που πρέπει να ικανοποιεί η υπηρεσία που θα επιλεγεί τελικά και οι οποίες παράμετροι αφορούν τις προτιμήσεις του χρήστη. Στη συνέχεια, συλλέγει πληροφορίες που αφορούν την ποιότητα της υπηρεσίας που αιτήθηκε ο χρήστης και έχοντας γνώση των προτιμήσεών του, επιλέγει την καταλληλότερη υπηρεσία γι' αυτόν και την συνθέτει. Η επιλογή της καταλληλότερης υπηρεσίας, όπως αυτή γίνεται από το μεσίτη, στηρίζεται σε διάφορα κριτήρια, όπως για παράδειγμα στις απαιτήσεις ποιότητας του χρήστη, αλλά και στις πληροφορίες που αφορούν τον πάροχο των υπηρεσιών (είτε στατικού είτε δυναμικού περιεχομένου) ή ακόμα και στην καθυστέρηση που υπάρχει στο δίκτυο. Λαμβάνοντας όλα αυτά υπόψη, ο μεσίτης αποτιμά μια αντικειμενική συνάρτηση U την οποία επιχειρεί να μεγιστοποιήσει, χρησιμοποιώντας σαν αλγόριθμο επιλογής υπηρεσιών μία από τις δύο προαναφερθείσες μοντελοποιήσεις. Οι διαδικτυακές υπηρεσίες που μελετώνται στις εργασίες αυτές, διαθέτουν 4 παραμέτρους ποιότητας. Αυτές είναι: (i) χρόνος απόκρισης, (ii) αξιοπιστία, (iii) κόστος υπηρεσίας και (iv) διαθεσιμότητα υπηρεσίας. Εκτενείς πληροφορίες για όλες αυτές τις παραμέτρους υπάρχουν στην παράγραφο 3.3.3. Η προτεινόμενη αντικειμενική συνάρτηση U εισάγει έναν παράγοντα οφέλους b που ουσιαστικά μοντελοποιεί το όφελος που έχει ένας πελάτης αν επιλέξει ένα συγκεκριμένο πάροχο υπηρεσιών που στην τρέχουσα περίοδο εξυπηρετεί μικρό αριθμό πελατών. Το όφελος που προκύπτει είναι εξυπηρέτηση του πελάτη με μικρότερο χρόνο απόκρισης, ενώ επιτυγχάνεται παράλληλα με αυτόν τον τρόπο εξισορρόπηση φορτίου (load balancing) [66] στους παρόχους παρόμοιων διαδικτυακών υπηρεσιών.

Όσον αφορά τους προτεινόμενους αλγορίθμους επιλογής υπηρεσιών, αυτοί ανήκουν σε δύο κατηγορίες όπως έχει ήδη αναφερθεί. Η πρώτη κατηγορία, και η οποία προσεγγίζει περισσότερο τους προτεινόμενους αλγορίθμους του κεφαλαίου αυτού, μοντελοποιεί το πρόβλημα της επιλογής υπηρεσιών σαν $MCKP$ [95] σε περίπτωση που ζητείται να ικανοποιηθεί μια παράμετρος ποιότητας του χρήστη. Σε αυτή την περίπτωση, η μοντελοποίηση επιτυγχάνεται θεωρώντας κάθε σακίδιο του $MCKP$ σαν βήμα επιλογής υπηρεσίας που το σύνολό τους θα συγκροτήσει τη σύνθετη υπηρεσία, ενώ το βάρος της παραμέτρου που χρειάζεται στο $MCKP$, προέρχεται από την παράμετρο ποιότητας που έχει θέσει ο χρήστης. Ωστόσο, σε περίπτωση που το πρόβλημα περιλαμβάνει περισσότερες από μια παραμέτρους ποιότητας, τότε ο αλγόριθμος που θα χρησιμοποιηθεί ανήκει στην

κατηγορία αλγορίθμων κατάλληλων για επίλυση του *MMKP* (περισσότερες πληροφορίες για την κατηγορία αυτή υπάρχουν στο κεφάλαιο 4). Σε αυτή την περίπτωση τα βάρη των παραμέτρων προέρχονται από τις προτιμήσεις του χρήστη, ενώ και στις δύο προσεγγίσεις, η συνάρτηση κέρδους που θα επιχειρήσει να μεγιστοποιήσει ο αλγόριθμος είναι η προτεινόμενη αντικειμενική συνάρτηση U , όπως βέβαια αυτή ορίζεται σε κάθε μία από τις περιπτώσεις, ανάλογα με το αν υπάρχουν μία ή περισσότερες παράμετροι – απαιτήσεις από τη μεριά του χρήστη.

Όσον αφορά την περίπτωση με μία παράμετρο, στην εργασία αυτή δεν προτείνεται κάποιος αλγόριθμος επιλογής υπηρεσιών αλλά αντίθετα γίνεται μια αποτίμηση των υπαρχουσών λύσεων: της εξαντλητικής προσέγγισης (που ωστόσο δε χρησιμοποιήθηκε κάποια τεχνική επέκτασης και οριοθέτησης (κεφάλαιο 4)), του *δυναμικού προγραμματισμού* (κεφάλαιο 4), καθώς και προσεγγιστικών λύσεων (βασισμένων στον προτεινόμενο αλγόριθμο του Pisinger [95]) κατάλληλων για επίλυση προβλημάτων που ανήκουν στην οικογένεια προβλημάτων του σακιδίου (knapsack family). Όσον αφορά τη μοντελοποίηση με πολλαπλές παραμέτρους ποιότητας, ο αλγόριθμος που προτείνεται, στηρίζεται σε έναν ευρυστικό αλγόριθμο (*HEU*) [116]. Ο προτεινόμενος αλγόριθμος *WS_HEU* όπως και ο αλγόριθμος *HEU* είναι προσεγγιστικοί μια και το *MMKP* ανήκει στην κατηγορία NP-Hard, προβλήματα δηλαδή που δεν επιδέχονται ακριβή λύση σε ορισμένες περιπτώσεις [83], όπως έχει ήδη αναφερθεί και στο προηγούμενο κεφάλαιο. Ο *WS_HEU* διαφοροποιείται από τον *HEU* στο εξής: Ενώ ο *HEU* αποτιμά αρχικά μια λύση (εξετάζοντας όλες τις λύσεις ακόμα και τις μη εφικτές) και τη βελτιώνει διαρκώς εφαρμόζοντας ένα μηχανισμό αντικαταστάσεων κλάσεων, ο *WS_HEU* απορρίπτει εξαρχής τις μη εφικτές λύσεις και ακολουθεί κατόπιν τον ίδιο μηχανισμό, με αποτέλεσμα όπως ισχυρίζονται να βελτιώνεται ο χρόνος απόκρισης του αλγορίθμου.

Όσον αφορά τη μοντελοποίηση του προβλήματος σαν πρόβλημα θεωρίας γράφων (εύρεσης συντομότερου μονοπατιού), αυτή επιτυγχάνεται ως εξής: Αρχικά, η κάθε υπηρεσία αποτελεί έναν κόμβο στο γράφο. Ωστόσο, στα προβλήματα θεωρίας γράφων είναι γνωστό ότι οι κόμβοι που συνδέονται μεταξύ τους έχουν βάρη στις ακμές τους και όχι οι ίδιοι οι κόμβοι. Επομένως, για να ξεπεράσουν το πρόβλημα αυτό, μεταφέρουν τα βάρη – απαιτήσεις της κάθε υπηρεσίας από τον κάθε κόμβο – υπηρεσία, στην ακμή που συνδέει την κάθε υπηρεσία με την επόμενη της (που ουσιαστικά παριστάνει την επόμενη υπηρεσία που θα συγκροτεί την σύνθετη υπηρεσία). Και στις δύο αυτές εργασίες [113], [114], οι προτεινόμενοι αλγόριθμοι κατασκευάζουν κατευθυνόμενους ακυκλικούς γράφους (Directed Acyclic Graph DAG) [115].

Στην περίπτωση που υπάρχει μια παράμετρος απαίτησης από το χρήστη, ο προτεινόμενος αλγόριθμος Constrained Shortest Path (*CSP*) στηρίζεται στην τεχνική του Breadth-First [82], διατηρώντας μια λίστα με την αντικειμενική συνάρτηση U για το κάθε μονοπάτι, ενώ στη λίστα προστίθενται υπηρεσίες που βελτιώνουν διαρκώς την τελική σύνθετη υπηρεσία.

Γνωστός αλγόριθμος από τη θεωρία γράφων με τον οποίο επιχειρείται σύγκριση είναι ο αλγόριθμος Constraint Bellman-Ford (*CBF*) [117], ενώ στη σύγκριση συμμετέχει και ο αλγόριθμος All Execution Path (*AEP*), που ουσιαστικά εξετάζει εξαντλητικά όλα τα δυνατά μονοπάτια κάνοντας χρήση τεχνικών επίλυσης, παρόμοιες με αυτές που εφαρμόζονται στο *MCKP* [95]. Ο αλγόριθμος *CSP* εμφανίζεται να έχει πολύ καλύτερα αποτελέσματα από τον *CBF* που μεγαλώνει εκθετικά με το πλήθος των κόμβων – υπηρεσιών. Σε σύγκριση με τον αλγόριθμο *AEP*, ο *CSP* παρουσιάζει χειρότερα αποτελέσματα, ωστόσο όπως οι Tao Yu και Kwei-Jay Lin ισχυρίζονται, δεν υπάρχει τρόπος στον εξαντλητικό αλγόριθμο να μοντελοποιήσει την επιβάρυνση στο χρόνο απόκρισης που οφείλεται στο χρόνο μετάδοσης από τον πάροχο.

Στην περίπτωση που υπάρχουν πολλές απαιτήσεις ποιότητας από τη μεριά του χρήστη, το πρόβλημα μοντελοποιείται ως Multi-Constraint Optimal Path Problem (*MCOP*). Ο προτεινόμενος αλγόριθμος για Multi Constrained Selection Path-K (*MCSP-K*) στηρίζεται στον *CSP* για μια παράμετρο και επομένως και σε αυτή την υλοποίηση διατηρείται μια λίστα με το κάθε μονοπάτι. Ωστόσο, επειδή αυτή η λίστα μπορεί να μεγαλώνει εκθετικά σε κάθε βήμα, καθιστώντας τη ως μη εφικτή λύση με γνώμονα το χρόνο απόκρισης και τη μνήμη, ο *MCSP-K* αντιμετωπίζει το πρόβλημα αυτό διατηρώντας ένα συγκεκριμένο (*K*) αριθμό από μονοπάτια σε κάθε λύση, μέθοδος που αναφέρεται στην εργασία [118].

Οι εργασίες των Tao Yu και Kwei-Jay Lin αν και προσφέρουν ολοκληρωμένη αρχιτεκτονική επιλογής και σύνθεσης υπηρεσιών, ωστόσο παρουσιάζουν κάποια μειονεκτήματα. Αρχικά, σε όλες τις περιπτώσεις που εξετάζουν, θεωρούν ότι κάθε παράμετρος στην τελική υπηρεσία προκύπτει σαν άθροισμα των αντίστοιχων παραμέτρων ποιότητας των υπηρεσιών που συνθέτουν την υπηρεσία αυτή, κάτι που προφανώς δεν ισχύει εν γένει. Επιπρόσθετα, δεν υπάρχει αλγόριθμος επιλογής υπηρεσιών όσον αφορά τη μοντελοποίηση της πρώτης κατηγορίας με μια παράμετρο, επομένως σε αυτή την κατηγορία η προτεινόμενη αρχιτεκτονική λειτουργεί συμπληρωματικά με τους γνωστούς αλγορίθμους επιλογής υπηρεσιών. Εξάλλου, ο αλγόριθμος *WS_HEU* που προτείνεται για πολλαπλές παραμέτρους αν και θεωρητικά βελτιώνει το χρόνο απόκρισης σε σχέση με τον *HEU*, όταν γίνεται αποτίμηση της επίδοσης του αλγορίθμου *WS_HEU*, η σύγκριση δε γίνεται με τον *HEU* ώστε να γίνει αντιληπτός ο ισχυρισμός τους, αλλά αντίθετα συγκρίνεται με την εξαντλητική λύση. Τέλος, ο αλγόριθμος *MCSP-K* είναι εκθετικής πολυπλοκότητας, κάτι που τον καθιστά προφανώς μη κατάλληλο σε ορισμένες περιπτώσεις.

Μια άλλη εξίσου ενδιαφέρουσα εργασία, είναι αυτή των Wand, Vitvar, Kerrigan και Toma [20]. Η αρχιτεκτονική που προτείνουν μαζί με τον αλγόριθμο επιλογής διαδικτυακών υπηρεσιών που παρουσιάζεται, στοχεύει στην αυτοματοποίηση της διαδικασίας σύνθεσης και επιλογής αυτών των υπηρεσιών. Και σε αυτή την εργασία, η επιλογή των υπηρεσιών στηρίζεται στις παραμέτρους ποιότητας των υπηρεσιών αυτών καθώς και στις προτιμήσεις

ποιότητας του χρήστη. Η αυτοματοποίηση της διαδικασίας επιτυγχάνεται με χρήση οντολογιών και πιο συγκεκριμένα στηριζόμενοι στο μοντέλο οντολογιών για διαδικτυακές υπηρεσίες (Web Service Modeling Ontology – WSMO) [119]. Αναλυτικότερα, επεκτείνεται το μοντέλο που προτείνεται από το WSMO με τη χρήση μιας νέας κλάσης QoS, η οποία θα χρησιμοποιηθεί για να περιγράψει τις παραμέτρους των υπηρεσιών καθώς και τις προτιμήσεις του χρήστη. Η κλάση αυτή αποτελεί πλήρες αντίγραφο της οντολογίας Γλώσσας, όπως αυτή έχει αναλυθεί στην παράγραφο 3.3.2 της παρούσας διατριβής. Το μόνο πεδίο με το οποίο επεκτείνεται η οντολογία αυτή, είναι το πεδίο που παριστάνει τη «ροπή» της παραμέτρου, που ουσιαστικά δηλώνει αν ο χρήστης επιθυμεί να έχει την εκάστοτε παράμετρο με τιμή μεγαλύτερη ή μικρότερη από την αντίστοιχή της, όπως αυτή προσφέρεται από την κάθε υπηρεσία. Η εργασία αυτή, εκτός από την υιοθέτηση της οντολογίας Γλώσσας, κάνει χρήση και της οντολογίας Λεξιλόγιο (§3.3.3). Επομένως, μια και οι δύο αυτές οντολογίες, αποτελούν κομμάτι της διατριβής αυτής, δεν κρίνεται σκόπιμο να αναλυθούν περαιτέρω.

Στηριζόμενοι σε αυτές τις οντολογίες, προτείνεται μια αρχιτεκτονική και ένας αλγόριθμος επιλογής υπηρεσιών. Ο κάθε χρήστης περιγράφει την υπηρεσία με το επιθυμητό επίπεδο ποιότητας, ενώ ο κάθε πάροχος υπηρεσιών δημοσιοποιεί τις υπηρεσίες που προσφέρει, που εκτός από τις λειτουργίες τους, περιλαμβάνουν και τις μη-λειτουργικές παραμέτρους τους (§3.1). Το μοντέλο που προτείνεται, εισάγει μια ταξινόμηση όσον αφορά τις παραμέτρους αυτές. Αυτή περιλαμβάνει μεταξύ άλλων, τη δημιουργία προκαθορισμένων επιλογών, που αφορούν την ποιότητα της προσφερόμενης υπηρεσίας από τη μεριά του χρήστη για πολλές παραμέτρους, διευκολύνοντας κατ' αυτόν τον τρόπο τους χρήστες που δε γνωρίζουν τη σημασία των παραμέτρων αυτών. Αποτελεί βέβαια επιλογή του χρήστη το αν θα χρησιμοποιήσει τελικά τις προεπιλογές αυτές, αν θα τις απορρίψει ή αν θα χρησιμοποιήσει ένα μέρος μόνο εξ αυτών. Στη συνέχεια, και με χρήση του προτεινόμενου μοντέλου, η διαδικασία συνεχίζει ως ακολούθως: Εφόσον εν γένει ο αριθμός των παραμέτρων από τη μεριά του χρήστη θα είναι διαφορετικός από αυτόν από τη μεριά των υπηρεσιών, ο προτεινόμενος αλγόριθμος χρησιμοποιεί πίνακα διαστάσεων $m \times k$ όπου m το πλήθος των υπηρεσιών μεταξύ των οποίων πρέπει να επιλεγθεί η καταλληλότερη υπηρεσία και k το πλήθος των παραμέτρων όπως αυτές ορίζονται από τη μεριά του χρήστη. Αφού κατασκευαστεί ο πίνακας αυτός, κανονικοποιείται ώστε όλες οι τιμές να έχουν εύρος στο διάστημα $[0, 1]$. Η κανονικοποίηση αυτή στηρίζεται στο πεδίο που περιγράφει τη «ροπή» της παραμέτρου. Στη συνέχεια, και αφού ο χρήστης επιλέξει το βάρος που έχει γι' αυτόν η κάθε παράμετρος, αθροίζονται όλες οι επιμέρους υπηρεσίες σε μια αντικειμενική συνάρτηση και η υπηρεσία με τη μεγαλύτερη τιμή στη συνάρτηση αυτή, είναι ουσιαστικά αυτή που επιλέγεται. Ο προτεινόμενος αλγόριθμος δεν προσφέρει καμία κλιμάκωση μια και εξετάζονται εξαντλητικά όλες οι περιπτώσεις - υπηρεσίες, ενώ παράλληλα βρίσκει εφαρμογή στην

επιλογή των υπηρεσιών για ένα μόνο χρήστη. Δηλαδή, δεν είναι εφικτό να χρησιμοποιηθεί ο αλγόριθμος αυτός στο πρόβλημα του κεφαλαίου αυτού για έναν επιπρόσθετο λόγο, αφού η επιλογή υπηρεσιών για ένα μόνο χρήστη συνοδεύεται από την απουσία συνθηκών περιορισμού. Επομένως, ο προτεινόμενος αλγόριθμος περιορίζεται σε πολύ μικρό αριθμό αιτήσεων – χρηστών.

Η αρχιτεκτονική που προτείνεται από τους Taher, Khatib και Basha [63] βρίσκει εφαρμογή στην επιλογή διαδικτυακών υπηρεσιών, στηριζόμενη στις παραμέτρους ποιότητας που διαθέτουν οι υπηρεσίες αυτές. Η αρχιτεκτονική εισάγει 2 μοντέλα, το ένα είναι το Μοντέλο Δεδομένων (Data Model) και είναι το μοντέλο στο οποίο στηρίζεται η αρχιτεκτονική για την περιγραφή του καθώς και για την περιγραφή των υπηρεσιών με τις παραμέτρους ποιότητάς τους με χρήση οντολογιών και το δεύτερο μοντέλο είναι το Υπολογιστικό Μοντέλο (Computation Model) που περικλείει ουσιαστικά την υλοποίηση του αλγόριθμου που χρησιμοποιείται για την επιλογή των υπηρεσιών αυτών. Ιδιαίτερο ενδιαφέρον της προτεινόμενης αρχιτεκτονικής αποτελεί το γεγονός ότι δεν προσφέρει ένα απλό μοντέλο επιλογής υπηρεσιών βάσει των παραμέτρων ποιότητάς τους, αλλά αντίθετα διαθέτει ένα δυναμικό περιβάλλον το οποίο προσαρμόζεται στις αλλαγές των παραμέτρων των υπηρεσιών αυτών, ενημερώνοντας ταυτόχρονα το χρήστη για την αλλαγή αυτή. Ο τρόπος που διαχειρίζεται δυναμικά τις αλλαγές αυτές στηρίζεται σε μια κλιμάκωση που εισάγεται όσον αφορά το χρόνο, που ουσιαστικά περιγράφει την κλιμάκωση που παρατηρείται σε έναν πάροχο όσον αφορά τη ζήτηση των υπηρεσιών του, ενώ υπεύθυνος για τη διαχείριση αυτή είναι το Υπολογιστικό Μοντέλο.

Πιο αναλυτικά, η προτεινόμενη αρχιτεκτονική QoS_IC (QoS Information and Communication) εισάγει 4 νέα στοιχεία συμπληρωματικά για το UDDI [37] : (i) το στοιχείο Διαχωρισμού αιτήσεων SOAP (Request Filterer) που διαχωρίζει αν η εισερχόμενη αίτηση για υπηρεσία δημιουργήθηκε από πάροχο ή από πελάτη, (ii) το στοιχείο Διαχειριστής Ενημέρωσης (Update Manager) που ενημερώνει το χρήστη των υπηρεσιών για τυχόν αλλαγές στις παραμέτρους ποιότητας των υπηρεσιών που χρησιμοποιεί, (iii) το στοιχείο Διαχειριστής QoS που είναι υπεύθυνο για την ενημέρωση του UDDI για τις αλλαγές αυτές και (iv) το στοιχείο Διαχειριστής Πιστοποίησης (Validation Manager) που ουσιαστικά πιστοποιεί ότι οι αιτήσεις των παροχών των υπηρεσιών είναι σύμφωνες με το προτεινόμενο μοντέλο οντολογιών.

Η επιλογή των υπηρεσιών στηρίζεται σε ένα μοντέλο υπολογισμού αποστάσεων στηριζόμενοι στην Ευκλείδεια γεωμετρία. Κατασκευάζεται ένας πίνακας με N γραμμές και K στήλες, όπου κάθε γραμμή αποτελεί υπηρεσία που ικανοποιεί τη λειτουργία που απαιτεί ο χρήστης από την υπηρεσία, ενώ η κάθε στήλη αποτελεί την κάθε παράμετρο ποιότητας της υπηρεσίας αυτής. Προτού υπολογίσουν την απόσταση αυτή, κανονικοποιούν τα στοιχεία του

πίνακα αυτού με εύρος τιμών $[0,1]$, σύμφωνα με το αν ο χρήστης επιθυμεί να έχει για την κάθε παράμετρο χαμηλότερη τιμή (για παράδειγμα χρόνο απόκρισης) ή υψηλότερη τιμή (για παράδειγμα διαθεσιμότητα) σε σχέση με αυτές που διαθέτουν οι υπάρχουσες υπηρεσίες. Χρησιμοποιώντας λοιπόν την απόσταση από την Ευκλείδεια γεωμετρία μεταξύ της κάθε υπηρεσίας και των προτιμήσεων που αφορούν την ποιότητα του χρήστη, υπολογίζεται η υπηρεσία που ικανοποιεί περισσότερο το χρήστη (η υπηρεσία που έχει τη μικρότερη απόσταση σε σχέση με τις προτιμήσεις του χρήστη). Πολυπλοκότητα του προτεινόμενου αλγορίθμου είναι $O(n)$, όπου n είναι ο αριθμός των υπηρεσιών που ικανοποιούν τις λειτουργικές απαιτήσεις του χρήστη. Αδυναμία του προτεινόμενου αλγορίθμου αποτελεί ότι η υπηρεσία που τελικά επιλέγεται μπορεί να μην ικανοποιεί τις προτιμήσεις του χρήστη ή τουλάχιστον να τις ικανοποιεί μόνο μερικώς, αφού ουσιαστικά ο προτεινόμενος αλγόριθμος επιλέγει την υπηρεσία που βρίσκεται πιο κοντά στις προτιμήσεις του χρήστη. Επιπρόσθετα, ο αλγόριθμος εξετάζει το πρόβλημα από τη μεριά του πελάτη, σε αντίθεση με τους αλγορίθμους του κεφαλαίου αυτού που εξετάζουν πολλαπλές αιτήσεις ταυτόχρονα, διαθέτοντας πολλές υπηρεσίες που ικανοποιούν την κάθε αίτηση ξεχωριστά.

Η εργασία των Liu, Liu, Jing, Tang, και Tang [120] μελετά το πρόβλημα επιλογής υπηρεσιών με παραμέτρους ποιότητας υπηρεσίας επισημαίνοντας αρχικά τη δυσκολία εύρεσης ενός αλγορίθμου επιλογής υπηρεσιών με παραμέτρους, αφού είναι πρόβλημα NP Complete (κεφάλαιο 4). Επομένως, για την επίλυση του προβλήματος επιχειρούν να βρουν μια προσεγγιστική λύση, ανάγοντας το πρόβλημα επιλογής υπηρεσιών από NP Complete σε ένα πρόβλημα μεγιστοποίησης με περιορισμούς. Ο προτεινόμενος αλγόριθμος στηρίζεται στη θεωρία βελτιστοποίησης γενετικών αλγορίθμων [121]. Χρησιμοποιώντας τη θεωρία γράφων όπως και οι Tao Yu και Kwei-Jay Lin ([113], [114]), δημιουργούν κόμβους στο γράφο που ουσιαστικά παριστάνουν τις διαθέσιμες υπηρεσίες, διαχωρίζοντας «λογικά» τους κόμβους αυτούς σε ομάδες. Η κάθε ομάδα από υπηρεσίες ενώνεται με την προηγούμενη και την επόμενη της, ενώ η τελική υπηρεσία απαρτίζεται από μια υπηρεσία από την κάθε ομάδα. Με αυτόν τον τρόπο, ανάγουν το πρόβλημα επιλογής υπηρεσιών σε πρόβλημα εύρεσης βέλτιστου μονοπατιού με πολλαπλούς περιορισμούς. Στον αλγόριθμο που προτείνουν, 4 παράμετροι ποιότητας χρησιμοποιούνται: κόστος, χρόνος απόκρισης, φήμη και αξιοπιστία (§3.3.3). Οι δύο πρώτες χρησιμοποιούνται στην αντικειμενική συνάρτηση (συνάρτηση εύρεσης ελαχίστου), ενώ οι άλλες δύο χρησιμοποιούνται σαν περιορισμοί. Ο αλγόριθμος στο πρώτο βήμα του, υπολογίζει κάποια αρχική λύση χρησιμοποιώντας τον αλγόριθμο *DFS* (Depth First Search) [82]. Στη συνέχεια, ο αλγόριθμος συνεχίζει χρησιμοποιώντας την ταξινόμηση Pareto [83] και τεχνικές διαχείρισης ποικιλίας πληθυσμού [121]. Κατόπιν, επιλέγονται πιθανές υπηρεσίες από κάθε ομάδα με τη μέθοδο του «τροχού της ρουλέτας», υιοθετώντας τεχνικές «κυριαρχίας» υπηρεσιών σε κάθε ομάδα και χρησιμοποιώντας διάφορες

γενετικές διασταυρώσεις, επιλέγεται τελικά μια υπηρεσία από κάθε ομάδα. Μειονέκτημα της εργασίας αυτής αποτελεί το γεγονός ότι δε γίνεται αποτίμηση του αλγορίθμου ούτε σε επίπεδο πολυπλοκότητας αλλά ούτε και σε επίπεδο μετρήσεων. Επιπρόσθετα, δεν καθίσταται σαφής ο ορισμός της αντικειμενικής συνάρτησης. Ωστόσο, η προσέγγιση που γίνεται για την επιλογή υπηρεσιών με χρήση γενετικών αλγορίθμων είναι ιδιαίτερα ελκυστική. Αξίζει να αναφερθεί τέλος ότι διαφοροποιούνται σε σχέση με τις περισσότερες άλλες εργασίες που μελετήθηκαν, στο γεγονός ότι η προτεινόμενη αντικειμενική συνάρτηση βελτιστοποιεί τη συνολική παρεχόμενη ποιότητα της τελικής υπηρεσίας και όχι αυτό της κάθε ομάδας των υπηρεσιών που συνθέτουν την τελική υπηρεσία, επομένως το πρόβλημα προσεγγίζει περισσότερο αυτό που παρουσιάζεται και στο παρόν κεφάλαιο.

Μια άλλη επίσης ενδιαφέρουσα εργασία επιλογής διαδικτυακών υπηρεσιών με παραμέτρους ποιότητας που στοχεύει στη βελτιστοποίηση της παρεχόμενης ποιότητας συνολικά είναι η [122]. Η βελτιστοποίηση αυτή στηρίζεται σε μοντέλα γραμμικού προγραμματισμού [82] και σε διαγράμματα καταστάσεων – Statecharts [123]. Χρησιμοποιώντας τα διαγράμματα αυτά και στηριζόμενοι στο γεγονός ότι κάθε σύνθετη υπηρεσία απαρτίζεται από «απλές» υπηρεσίες, αναπαριστάται το κάθε πιθανό μονοπάτι σύνθεσης της τελικής υπηρεσίας σε έναν προσανατολισμένο ακυκλικό γράφο (Direct Acyclic Graph) [115]. Κάθε κόμβος στο γράφο αυτό παριστάνει και μια διαδικτυακή υπηρεσία, η οποία στο προτεινόμενο μοντέλο δύναται να διαθέτει τις ακόλουθες 5 παραμέτρους (§3.3.3): κόστος, διάρκεια εκτέλεσης, φήμη, αξιοπιστία και διαθεσιμότητα, ενώ το μοντέλο μπορεί να εμπλουτιστεί και με άλλες παραμέτρους αν αυτό κρίνεται απαραίτητο. Όσον αφορά το μοντέλο επιλογής διαδικτυακών υπηρεσιών, η υλοποίησή του στηρίζεται στην τεχνική επιλογής πολλαπλών χαρακτηριστικών (Multiple Attribute Decision Making) [124]. Σύμφωνα με την τεχνική αυτή, κατασκευάζεται ένας πίνακας που κάθε γραμμή του παριστάνει και ένα μονοπάτι σύνθεσης της υπηρεσίας, ενώ η επιλογή του μονοπατιού γίνεται με την τεχνική του Simple Additive Weighting (SAW) [125]. Η προσέγγιση αυτή ωστόσο σε περίπτωση που έχουμε M κόμβους που συνθέτουν την υπηρεσία και N διαθέσιμες υπηρεσίες σε κάθε κόμβο, υποδηλώνει ότι θα πρέπει να ελεγχθούν M^N διαφορετικά δυνατά μονοπάτια σύνθεσης. Κάτι που άλλωστε επισημαίνεται και από τους συγγραφείς, αναγνωρίζοντάς την ως μη πρακτική για μεγάλες τιμές M και N . Επομένως, η επιλογή μονοπατιού σύνθεσης υπηρεσιών ακολουθεί ένα άλλο πιο πρακτικό μοντέλο, στηριζόμενο σε μια αντικειμενική συνάρτηση που εμπλέκονται και οι 5 παράμετροι ποιότητας. Ωστόσο, οι συνθήκες περιορισμού που εισάγονται στο μοντέλο αυτό σχετίζονται περισσότερο με λειτουργικές διεργασίες σύνθεσης της υπηρεσίας, αντί να προσφέρουν στο χρήστη τη δυνατότητα να προσαρμόσει την τελική υπηρεσία στις δικές του ανάγκες. Για παράδειγμα, δεν εισάγεται κάποια συνθήκη περιορισμού στο κόστος, αφού όπως ισχυρίζονται οι

συγγραφείς σε αυτή την περίπτωση το πρόβλημα ανάγεται σε NP Hard [83]. Αυτό βέβαια αποτελεί μεγάλο μειονέκτημα του προτεινόμενου αλγορίθμου, αφού δε δίνεται η δυνατότητα στο χρήστη να προσαρμόσει την τελική σύνθετη υπηρεσία στις ιδιαίτερες προτιμήσεις και απαιτήσεις του. Επιπρόσθετα, δε γίνεται αποτίμηση της πολυπλοκότητας του αλγορίθμου, ούτε δίνονται πρόσθετες πληροφορίες για τον τρόπο που βελτιστοποιείται η αντικειμενική συνάρτηση.

Η εργασία των Pathak, Koul, Caragea, και Honavar [126] προσφέρει μια αρχιτεκτονική εύρεσης και επιλογής διαδικτυακών υπηρεσιών στηριζόμενοι σε οντολογίες (OWL, OWL-S). Προσφέρουν μια διαφορετική ταξινόμηση στις παραμέτρους ποιότητας των υπηρεσιών σε σχέση με αυτή που παρουσιάζεται στην παράγραφο 3.3.3 μια και ταξινομούν την κάθε παράμετρο, βάσει δύο κριτηρίων: (i) Αν αφορά συγκεκριμένη περιοχή (domain) (π.χ. εστιατόριο) ή όχι και (ii) αναγνωρίζουν την ανάγκη να προσαρμοστούν στις επιθυμίες του χρήστη, ο οποίος μπορεί να θέλει να ορίσει ορισμένες μόνο παραμέτρους ποιότητας και όχι όλες όσες προσφέρονται από τον κάθε πάροχο υπηρεσιών. Γι' αυτό το λόγο, δημιουργούν ένα πίνακα $m \times n$, όπου m το πλήθος των παραμέτρων που ενδιαφέρουν το χρήστη και n το πλήθος των υπηρεσιών. Ακόμη, όπως και στην εργασία των Roman, Lausen, και Keller [119], διαχωρίζουν τις παραμέτρους αυτές ανάλογα με το αν ο χρήστης θέλει να έχει μεγαλύτερη ή μικρότερη τιμή από αυτές που προσφέρονται από τις υπηρεσίες, ενώ παράλληλα ορίζουν και ένα βάρος σε κάθε παράμετρο, που παριστάνει τη σπουδαιότητα που έχει η παράμετρος αυτή της εκάστοτε υπηρεσίας για το χρήστη. Κατόπιν, ορίζεται μια αντικειμενική συνάρτηση η οποία υπολογίζει, για την κάθε υπηρεσία που ικανοποιεί το χρήστη, το γινόμενο των παραμέτρων με το βάρος της κάθε παραμέτρου. Σε περίπτωση που τελικά επιλέγονται περισσότερες από μία υπηρεσίες, τότε το μοντέλο επιλογής υπηρεσιών επεκτείνεται με τη χρήση μιας τεχνικής βαθμολόγησης της κάθε υπηρεσίας από τη μεριά του χρήστη, που ουσιαστικά εκφράζει ποια παράμετρο επιθυμεί ο χρήστης να προσεγγίζει περισσότερο την αντίστοιχη παράμετρο όπως την έχει θέσει αυτός. Ο αλγόριθμος αυτός δεν προσφέρει καμία κλιμάκωση, ενώ είναι προφανές ότι λειτουργεί εξαντλητικά εξετάζοντας και υπολογίζοντας την κάθε υπηρεσία ξεχωριστά, κάτι που προφανώς βρίσκει εφαρμογή σε μικρό αριθμό υπηρεσιών. Ωστόσο, η εναλλακτική προσέγγιση που προτείνεται για επιλογή υπηρεσιών με γνώμονα τις επιθυμίες του χρήστη τόσο όσον αφορά τον αριθμό των παραμέτρων όσο και της τεχνικής βαθμολόγησης καθιστούν την προσέγγιση αυτή ελκυστική, τουλάχιστον για περιορισμένο αριθμό διαδικτυακών υπηρεσιών.

Ο πίνακας που ακολουθεί περιέχει συγκεντρωτικά κάποια στοιχεία των εργασιών που μελετήθηκαν, ενώ η τελευταία γραμμή του πίνακα παρουσιάζει τα αντίστοιχα στοιχεία για τον αλγόριθμο που προτείνεται στο κεφάλαιο αυτό. Στην πρώτη στήλη αναφέρεται το όνομα του πρώτου συγγραφέα μόνο - για εργονομικούς λόγους - από την κάθε εργασία. Η δεύτερη στήλη παριστάνει αν η αντίστοιχη εργασία έχει στηριχθεί σε κάποιο μοντέλο, στο οποίο οι

παράμετροι ποιότητας των διαδικτυακών υπηρεσιών περιγράφονται με οντολογίες. Η τρίτη στήλη αν η επιλογή των διαδικτυακών υπηρεσιών γίνεται με κριτήρια που αφορούν τον κάθε πελάτη ξεχωριστά ή αφορά συνολικά όλους τους πελάτες, όποτε το πεδίο συμπληρώνεται αρνητικά ή καταφατικά αντίστοιχα. Η τέταρτη στήλη παρουσιάζει την πολυπλοκότητα του αλγορίθμου της κάθε εργασίας, ενώ η τελευταία στήλη δηλώνει αν ο προτεινόμενος αλγόριθμος διαθέτει συνθήκες περιορισμού που αφορούν τις παραμέτρους ποιότητας των υπηρεσιών και αν ναι, τον αριθμό των συνθηκών αυτών.

<i>Αλγόριθμος Επιλογής</i>	<i>Οντολογία QoS</i>	<i>Συνολική Επιλογή</i>	<i>Πολυπλοκότητα</i>	<i>Συνθήκες Περιορισμού</i>
Yu	Όχι	Ναι	Πολυωνυμικοί ~ Εκθετικοί	>2
Wand	Ναι	Όχι	Εξαντλητικά	Όχι
Taher	Ναι	Όχι	Εξαντλητικά	Όχι
Liu	Όχι	Ναι	Δεν αναφέρεται	2
Zeng	Όχι	Ναι	Δεν αναφέρεται	Όχι
Pathak	Ναι	Όχι	Εξαντλητικά	Όχι
SMCK	Ναι	Ναι	$O(n \log n) \sim$ $O(n^2)$	1

Πίνακας 5-1 Σύγκριση αλγορίθμων επιλογής διαδικτυακών υπηρεσιών παγκόσμιας βιβλιογραφίας

5.4. Γραμμικές και μη γραμμικές παράμετροι

Εξετάζοντας τις παραμέτρους ποιότητας που περιλαμβάνονται στην οντολογία Λεξιλόγιο (§3.3.3), αντιλαμβάνεται κανείς εύκολα ότι δεν έχει λογική να συμπεριληφθεί η κάθε μια από αυτές στη συνθήκη περιορισμού της μοντελοποίησης του προβλήματος που εξετάζεται στο κεφάλαιο αυτό (σχέση (5-6)). Αντικείμενο της παραγράφου αυτής είναι η εύρεση των παραμέτρων που μπορούν να συμπεριληφθούν και περιλαμβάνονται στην εν λόγω οντολογία, τροποποιώντας για κάθε μια από αυτές τη συνθήκη περιορισμού, αν αυτό κρίνεται απαραίτητο. Αυτό οφείλεται στο γεγονός ότι ορισμένες από τις παραμέτρους ποιότητας υπηρεσίας δεν είναι γραμμικές. Κατά συνέπεια η απαίτηση από τη μεριά των πελατών για συγκεκριμένη τιμή στην παράμετρο αυτή δε λειτουργεί αθροιστικά για τους διαθέσιμους πόρους του παρόχου για την εν λόγω παράμετρο, θεώρηση που έγινε για τη δημιουργία της σχέσεως (5-6). Ένα παράδειγμα για να καταστεί αυτό περισσότερο σαφές είναι η περίπτωση της παραμέτρου της *Καθυστέρησης*. Αν ένας πάροχος εξυπηρετεί 2 πελάτες του προσφέροντάς τους υπηρεσίες με *Καθυστέρηση* 0.1msec η κάθε μια, τότε αυτό δε σημαίνει ότι για να μπορέσει να τους εξυπηρετήσει θα πρέπει η *Καθυστέρηση* του να είναι μεγαλύτερη ή ίση από 0.2 msec.

Ένας γενικός κανόνας που πρέπει να ακολουθηθεί κατά την εύρεση των παραμέτρων που μπορούν να συμμετέχουν στη συνθήκη περιορισμού είναι ότι θα πρέπει η αύξηση του πλήθους των πελατών να έχει επίδραση στο άνω όριο της τιμής που μπορεί να διαθέσει ο πάροχος για την παράμετρο αυτή. Αυτό έχει σαν συνέπεια από τη διαδικασία επιλογής να εξαιρούνται οι παράμετροι εκείνες που είναι (i) μη ποσοτικές και (ii) η τιμή τους δεν επηρεάζει τους διαθέσιμους πόρους του παρόχου. Για παράδειγμα η παράμετρος *Ρυθμιστικότητα* αν και είναι ποσοτική παράμετρος, ωστόσο επειδή η τιμή της δεν επηρεάζει κάποιο πόρο (resource) του παρόχου, δε συμμετέχει κατά τη διαδικασία της επιλογής. Επιπρόσθετα, παράμετροι που δηλώνουν την ικανότητα του παρόχου να προσαρμόζει το σύστημά του καθώς αυξάνεται το πλήθος των πελατών και να είναι διαθέσιμος σε αυτούς, όπως η *Κλιμάκωση* και η *Προσπελασιμότητα* αντίστοιχα, επειδή αποτελούν «χαρακτηριστικά» του παρόχου και δε μεταβάλλεται η τιμή τους όσο αυξάνεται ο αριθμός των αιτήσεων αλλά αντίθετα δηλώνουν ότι ο πάροχος μπορεί να ανταποκριθεί σε πιο απαιτητικές καταστάσεις, επίσης δε μπορούν να συμμετάσχουν στη συνθήκη περιορισμού. Επομένως παράμετροι που εξαιρούνται από τη διαδικασία συμμετοχής είναι για παράδειγμα ο *Ρυθμός Λαθών*, το *MTTR*, το *MTBF* κ.α. αφού και αυτές αποτελούν «χαρακτηριστικά» του παρόχου.

Συνοψίζοντας και λαμβάνοντας υπόψη τα όσα αναφέρθηκαν παραπάνω από τη διαδικασία επιλογής εξαιρούνται οι ακόλουθες παράμετροι:

- Μη ποσοτικές: *Υποστηριζόμενα πρότυπα, Ακεραιότητα, Εμπιστευτικότητα, Ελεγκσιμότητα, Πιστοποίηση, Εξουσιοδότηση, Κρυπτογράφηση δεδομένων και Μη αποκήρυξη.*
- Ποσοτικές: *Διαθεσιμότητα, Προσπελασιμότητα, Κλιμάκωση, MTTR, MTBF, Ρυθμός λαθών, Σταθερότητα και Ρυθμιστικότητα.*

Αυτό βέβαια δε σημαίνει ότι η συνεισφορά των παραμέτρων αυτών είναι αμελητέα, αφού όλες οι παράμετροι συμμετέχουν κατά τη διαδικασία ταιριάσματος (matchmaking), διαδικασία που επιλέγει από το σύνολο των υπηρεσιών που προσφέρει ένας πάροχος τις υπηρεσίες εκείνες που ικανοποιούν τον κάθε πελάτη / αίτηση ξεχωριστά. Υπενθυμίζεται ότι η διαδικασία αυτή αν και δεν αποτελεί αντικείμενο ενασχόλησης της διατριβής αυτής, ωστόσο κάποια στοιχεία της διατίθενται στην παράγραφο 2.3.3.

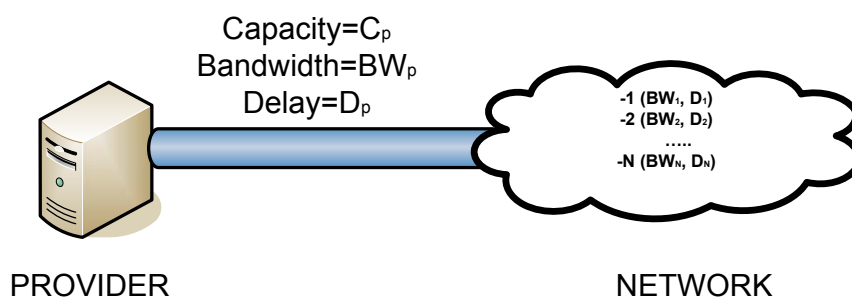
Επομένως, οι παράμετροι ποιότητας υπηρεσίας που περιλαμβάνονται στην οντολογία Λεξιλόγιο και είναι εφικτό να συμμετέχουν στη συνθήκη περιορισμού είναι οι ακόλουθες:

- *Γραμμικές: Κόστος, Χωρητικότητα και Ρυθμός Δεδομένων.*
- *Μη γραμμικές: Καθυστέρηση, Χρόνος Απόκρισης και Μεταβολή Καθυστέρησης.*

Στο σημείο αυτό πρέπει να επισημανθεί ότι η μοντελοποίηση των μη γραμμικών παραμέτρων ποιότητας με δεδομένες τιμές τόσο στην κάθε υπηρεσία που προσφέρεται όσο και στο άνω όριο των τιμών που διαθέτει ο πάροχος για τις παραμέτρους αυτές είναι σχεδόν αδύνατο να επιτευχθεί. Αυτό οφείλεται στο γεγονός ότι οι παράμετροι αυτές εξαρτώνται από πάρα πολλούς παράγοντες, με αποτέλεσμα ακόμα και σε ιδανικές συνθήκες να μην είναι εφικτή η μοντελοποίησή τους.

Ωστόσο, μια μη γραμμική παράμετρος που θα επιχειρηθεί η μοντελοποίησή της στην παρούσα διατριβή και θα εξεταστεί πως επιδρά η αύξηση του πλήθους των πελατών στη συνολική τιμή που διαθέτει ο πάροχος για την παράμετρο αυτή είναι η *Καθυστέρηση*.

Ένα «αυστηρό» μοντέλο επίδρασης της παραμέτρου της *Καθυστέρησης* της κάθε υπηρεσίας στη συνολική *Καθυστέρηση* που διαθέτει ο πάροχος επιτυγχάνεται με την υιοθέτηση ενός μοντέλου σημείο-προς-σημείο (point-to-point) για τον πάροχο και το δίκτυο των πελατών του, όπως στο σχήμα που ακολουθεί:



Σχήμα 5-1 Μοντέλο παρόχου σημείο-προς-σημείο για τον υπολογισμό της *Καθυστέρησης*

Η υιοθέτηση του μοντέλου αυτού στηρίζεται σε ορισμένες παραδοχές. Οι σημαντικότερες από αυτές είναι ότι θεωρείται ότι ο πάροχος όπως και οι υπηρεσίες που προσφέρει έχουν σταθερή (και όχι μέση) *Καθυστέρηση*, ενώ δεν εξετάζεται η τοπολογία του δικτύου με αποτέλεσμα να μη μελετάται η επίδραση της «πρόσθετης» *Καθυστέρησης* του κάθε πελάτη που οφείλεται στη θέση του στο δίκτυο. Η θεώρηση αυτή βέβαια δεν είναι τόσο αυστηρή, αφού κατά την επιλογή των υπηρεσιών που ικανοποιούν τον κάθε πελάτη – στάδιο το οποίο προηγείται της εφαρμογής των τεχνικών που θα παρουσιαστούν στο κεφάλαιο αυτό – μπορεί να ληφθεί υπόψη η «πρόσθετη» αυτή *Καθυστέρηση*. Ακόμη, θεωρείται ότι ο πάροχος διαθέτει πολύ μεγάλου μήκους ουρά αναμονής για τα πακέτα που πρόκειται να αποστείλει στους πελάτες του, με αποτέλεσμα να μην επιβαρύνεται με πρόσθετες καθυστερήσεις η διαδικασία εξυπηρέτησης των πελατών του, ενώ η εξυπηρέτηση των πελατών του επιτυγχάνεται ακολουθώντας το εξής μοντέλο: Σε κάθε κύκλο αποστολής πακέτων, ο κάθε πελάτης έχει λάβει μόνο ένα πακέτο. Στηριζόμενοι λοιπόν σε αυτές τις παραδοχές και θεωρώντας ότι ο πάροχος αποστέλλει πακέτα με σταθερή *Καθυστέρηση* D_p ,

τότε αν σε μια δεδομένη χρονική στιγμή ο πάροχος διαθέτει l διαφορετικές υπηρεσίες σε N πελάτες με:

m_1 πλήθος πελατών να χρησιμοποιούν υπηρεσία με *Καθυστέρηση* D_1 ,

m_2 πλήθος πελατών να χρησιμοποιούν υπηρεσία με *Καθυστέρηση* D_2 ,

.....

m_l πλήθος πελατών να χρησιμοποιούν υπηρεσία με *Καθυστέρηση* D_l ,

$$\text{όπου} \quad \sum_{i=1}^l m_i = N$$

τότε ο κάθε κύκλος αποστολής πακέτων ολοκληρώνεται σε χρόνο:

$$\sum_{i=1}^l m_i \times D_p = N \times D_p$$

Επομένως, όπως είναι εύλογο για να είναι βέβαιο ότι ο πάροχος θα εξυπηρετήσει τους πελάτες του με τους «υποσχόμενους» χρόνους *Καθυστέρησης* θα πρέπει να ισχύει:

$$N \times D_p \leq D_1$$

$$N \times D_p \leq D_2$$

....

$$N \times D_p \leq D_l$$

Για να ικανοποιούνται ταυτόχρονα όλες αυτές οι συνθήκες, αρκεί ο πάροχος να ικανοποιεί την υπηρεσία που έχει τη μικρότερη *Καθυστέρηση*, έστω $D_{\min}$. Επομένως, καταλήγουμε στην επόμενη σχέση:

$$\frac{N}{D_{\min}} \leq \frac{1}{D_p} \quad (5-9)$$

Η σχέση αυτή αποτελεί την τροποποιημένη συνθήκη περιορισμού για την περίπτωση της παραμέτρου της *Καθυστέρησης*, αφού όπως έχει ήδη αναφερθεί πρόκειται για μια μη γραμμική παράμετρο.

Η υιοθέτηση του μοντέλου αυτού από τη μεριά του παρόχου θα έχει σαν αποτέλεσμα να επωφελούνται οι πελάτες στους οποίους προσφέρονται υπηρεσίες με *Καθυστέρηση* μεγαλύτερη από $D_{\min}$, αφού θα εξυπηρετούνται με *Καθυστέρηση* μικρότερη από αυτή που έχουν συμφωνήσει με τον πάροχο, περίπτωση που προφανώς δε λειτουργεί επιβαρυντικά ως προς τον πελάτη.

Στον πίνακα που ακολουθεί περιλαμβάνονται οι 3 παράμετροι ποιότητας υπηρεσίας της οντολογίας Λεξιλόγιο (§3.3.4) για τις οποίες θα προταθούν αλγόριθμοι μεγιστοποίησης του κέρδους ενός παρόχου. Σύμφωνα και με το παραπάνω σχήμα (Σχήμα 5-1), αν υποθεθεί

ότι ο πάροχος διαθέτει μέγιστη *Χωρητικότητα* C_p , μέγιστο *Ρυθμό Δεδομένων* BW_p και σταθερή *Καθυστέρηση* D_p , τότε η συνθήκη περιορισμού για την κάθε μια παράμετρο είναι αυτή που περιλαμβάνονται στη δεύτερη στήλη του Πίνακα 5-2.

Παράμετρος Ποιότητας	Συνθήκη Περιορισμού
Χωρητικότητα	$\sum_{i=1}^N \sum_{j=1}^{M_i} S_{ij} \leq C_p$
Ρυθμός δεδομένων	$\sum_{i=1}^N \sum_{j=1}^{M_i} BW_{ij} * S_{ij} \leq BW_p$
Καθυστέρηση	$\frac{N}{D_{\min}} \leq \frac{1}{D_p}$

Πίνακας 5-2 Παράμετροι ποιότητας υπηρεσίας που συμμετέχουν στη συνθήκη περιορισμού

Στον πίνακα αυτό η συνθήκη περιορισμού για την *Καθυστέρηση* είναι η σχέση (5-10), ενώ για το *Ρυθμό Δεδομένων* προκύπτει από τη σχέση (5-6), αν αντικατασταθεί η μεταβλητή w με τη μεταβλητή BW . Από την ίδια σχέση ((5-6)) προκύπτει και η συνθήκη περιορισμού για την παράμετρο της *Χωρητικότητας* με τη θεώρηση ότι $w_{ij} = 1$, αφού η παράμετρος αυτή παριστάνει ουσιαστικά το πλήθος των πελατών που μπορεί να εξυπηρετήσει ο πάροχος σε μια δεδομένη στιγμή (§3.3.3).

Κλείνοντας την παράγραφο αυτή, επισημαίνεται ότι η μοντελοποίηση της παραμέτρου του *Χρόνου Απόκρισης* θα μπορούσε να επιτευχθεί κάνοντας τις ίδιες θεωρήσεις και ακολουθώντας ακριβώς τα ίδια βήματα που έγιναν κατά τη μοντελοποίηση της παραμέτρου της *Καθυστέρησης*, αφού ουσιαστικά μπορεί να θεωρηθεί ότι ο *Χρόνος Απόκρισης* είναι ο διπλάσιος χρόνος της *Καθυστέρησης* θεωρώντας σταθερό – και αμελητέο – το χρόνο επεξεργασίας της κάθε αίτησης από τη μεριά του παρόχου. Όσον αφορά την παράμετρο της *Μεταβολής Καθυστέρησης*, η μοντελοποίησή της είναι ένα πολύ δύσκολο επίτευγμα, αρκεί να αναλογιστεί κανείς ότι στηρίζεται σε ένα μέγεθος (*Καθυστέρηση*) του οποίου η μοντελοποίηση γίνεται προσεγγιστικά μόνο και κάνοντας αρκετές παραδοχές – θεωρήσεις. Άλλωστε και οι περισσότεροι πάροχοι για να προσφέρουν υπηρεσίες, των οποίων η τιμή στην παράμετρο αυτή κυμαίνεται μεταξύ επιτρεπτών τιμών, εφαρμόζουν διάφορους μηχανισμούς στηριζόμενοι αποκλειστικά σε μετρήσεις σε πραγματικό περιβάλλον. Από τους πιο δημοφιλείς μηχανισμούς είναι η δημιουργία ουρών μεγαλύτερου μήκους εις βάρος ωστόσο της προσφερόμενης *Καθυστέρησης* [127].

5.5. Αλγόριθμος επιλογής υπηρεσιών – *Selective Multiple Choice Knapsack (SMCK)*

Σε αυτήν την παράγραφο παρουσιάζεται ο προτεινόμενος αλγόριθμος *Selective Multiple Choice Knapsack (SMCK)*, ο οποίος στοχεύει στην εύρεση κατάλληλων υπηρεσιών με στόχο τη βελτιστοποίηση του κέρδους από τη μεριά του παρόχου για κάθε μια από τις παραμέτρους ποιότητας υπηρεσίας της προηγούμενης παραγράφου. Όπως έχει ήδη αναφερθεί, ο αλγόριθμος αυτός υιοθετεί τεχνικές παρόμοιες με αυτές που ακολουθούνται κατά την επίλυση προβλημάτων που ανήκουν στην οικογένεια προβλημάτων του σακιδίου (knapsack family) (κεφάλαιο 4). Κατά συνέπεια προτείνεται η χρήση του προηγούμενου κεφαλαίου ως αναφορά, όπου αυτό κρίνεται απαραίτητο.

Εξετάζοντας ξεχωριστά την κάθε παράμετρο ποιότητας, ο αλγόριθμος που θα μεγιστοποιήσει το κέρδος του παρόχου είναι:

Περίπτωση μέγιστης Χωρητικότητας C_p :

Η περίπτωση αυτή είναι η ευκολότερη για να αντιμετωπιστεί. Πράγματι αν ο πάροχος διαθέτει μέγιστη Χωρητικότητα C_p , τότε ο υπολογισμός του μέγιστου κέρδους του επιτυγχάνεται ακολουθώντας την ακόλουθη μέθοδο:

- Επιλέγω από τον κάθε πελάτη i την υπηρεσία που έχει το μεγαλύτερο κέρδος για τον πάροχο, έστω S_{ik} :

$$p_{ik} = \max \sum_{j=1}^{M_i} p_{ij} \quad \text{για } i = 1, 2, \dots, N$$

- Διατάσσω τις υπηρεσίες αυτές όλων των πελατών σε φθίνουσα σειρά ως προς το κέρδος:

$$p_{1k} \geq p_{2k} \geq \dots \geq p_{Nk}$$

- Επιλέγω τις πρώτες C_p πλήθος υπηρεσίες από την παραπάνω διάταξη, με αποτέλεσμα το κέρδος του παρόχου να δίνεται τελικά από τη σχέση:

$$\text{Max_Profit} = \sum_{i=1}^{C_p} p_{ik}$$

Περίπτωση μέγιστου Εύρους Ζώνης BW_p :

Η περίπτωση αυτή μπορεί να αντιμετωπιστεί επιτυχώς με τεχνικές που αναλύθηκαν στο προηγούμενο κεφάλαιο και ακολουθήθηκαν για την επίλυση του *MCKP* (§4.4). Για να είναι αυτό βέβαια ορθό, θα πρέπει να γίνει κατάλληλη αναγωγή του υπάρχοντος προβλήματος σε μορφή επιλύσιμη από τις μεθόδους που εφαρμόζονται για το *MCKP*. Αυτό επιτυγχάνεται αν προστεθεί σε κάθε πελάτη η «μηδενική» υπηρεσία, δηλαδή η υπηρεσία με μηδενικό κέρδος και μηδενικό εύρος ζώνης. Πράγματι, με τη θεώρηση αυτή είναι εύλογο ότι σε περίπτωση

που ο πελάτης συμμετέχει στην τελική λύση με τη «μηδενική» υπηρεσία, τότε είναι προφανές ότι ο πελάτης αυτός δε θα εξυπηρετηθεί. Σε διαφορετική περίπτωση, θα εξυπηρετηθεί με την υπηρεσία που επιλέχθηκε για αυτόν από την προτεινόμενη λύση. Επομένως, εισάγοντας τη «μηδενική» υπηρεσία στη συνθήκη περιορισμού, αυτή μετασχηματίζεται από ανισότητα σε ισότητα ως εξής:

$$\sum_{i=1}^N \sum_{j=1}^{M_i} BW_{ij} * S_{ij} = BW_p$$

με αποτέλεσμα το πρόβλημα να μετασχηματίζεται απόλυτα στο *MCKP*. Κατά συνέπεια, η επίλυσή του μπορεί να στηριχθεί σε μεθόδους όπως αυτές που αναλύονται στις εργασίες [91], [109], [128] και [129].

Περίπτωση σταθερής Καθυστέρησης D_p :

Αν και η περίπτωση αυτή εμφανίζει αρκετές ομοιότητες με αυτή του μέγιστου *Εύρους Ζώνης* και κατά συνέπεια και του *MCKP*, ωστόσο δε μπορούν να ακολουθηθούν οι προτεινόμενες μέθοδοι απόλυτα εξαιτίας της ακόλουθης αδυναμίας: Δεν είναι εφικτό να προστεθεί η «μηδενική» υπηρεσία (υπηρεσία με μηδενικό κέρδος και άπειρη *Καθυστέρηση*), αφού σε αυτή την περίπτωση η υπηρεσία που εξυπηρετεί τον κάθε πελάτη και έχει τη μεγαλύτερη

Καθυστέρηση θα αγνοείται. Αυτό οφείλεται στο γεγονός ότι λόγος $\lambda_{ij} = \frac{P_{ij} - P_{ij-1}}{w_{ij} - w_{ij-1}}$ (slope -

κλίση) που εφαρμόζεται κατά την επίλυση του *LMCKP* (§4.4) με όρο $(j-1)$ τη μηδενική υπηρεσία να είναι πάντα 0 και κατά συνέπεια να μη λαμβάνονται ποτέ υπόψη οι υπηρεσίες αυτές. Βέβαια η «εξαιρέση» των υπηρεσιών με τη μεγαλύτερη *Καθυστέρηση* δεν είναι το μόνο πρόβλημα που προκύπτει, ωστόσο είναι αρκετό για να καταστεί σαφές για ποιο λόγο δεν υιοθετήθηκαν απόλυτα οι μέθοδοι αυτές. Συμπερασματικά λοιπόν θα πρέπει η αρχική λύση που θα προταθεί από τον αλγόριθμο να είναι αναγκαστικά μεταξύ των υπάρχουσών προσφερόμενων υπηρεσιών από τον πάροχο. Αναλογιζόμενοι ακόμα την «αδυναμία» του προτεινόμενου μοντέλου στη συνθήκη περιορισμού (σχέση (5-9)), όπου ο πάροχος «συμβιβάζεται» να προσφέρει υπηρεσίες με *Καθυστέρηση* μικρότερη ή ίση από αυτή που ικανοποιεί τον κάθε πελάτη, προκειμένου να εξασφαλίσει ότι σε κάθε περίπτωση όλοι οι πελάτες θα εξυπηρετηθούν στη χειρότερη περίπτωση με την «υποσχόμενη» *Καθυστέρηση*, οδηγούμαστε στο συμπέρασμα ότι θα πρέπει η αρχική λύση να απαρτίζεται από υπηρεσίες με τις μεγαλύτερες *Καθυστερήσεις*. Κάτι τέτοιο βέβαια δε θα πρέπει να θεωρηθεί ότι λειτουργεί εις βάρος του συνολικού κέρδους του παρόχου, αφού ο αλγόριθμος συνεχίζει λαμβάνοντας υπόψη όλες τις υπηρεσίες, δίνοντας τη δυνατότητα αντικατάστασης των αρχικών επιλογών, εφόσον αυτό οδηγεί στη βελτίωση του κέρδους του παρόχου, όπως άλλωστε θα φανεί και παρακάτω. Κατά συνέπεια εφόσον στην τελική προτεινόμενη λύση δε θα συμμετέχει κάποια υπηρεσία με μικρή *Καθυστέρηση*, αυτό θα σημαίνει ότι το προτεινόμενο μοντέλο είναι αρκετά

ακριβές. Τα πειραματικά αποτελέσματα επιβεβαιώνουν την άποψη αυτή, όπως θα φανεί και στην επόμενη παράγραφο.

Ο υπολογισμός του μέγιστου κέρδους του παρόχου στηρίζεται στα ακόλουθα βήματα:

Βήμα 1 (Εφαρμογή κριτηρίων επικράτησης (*dominance criteria*), παρόμοια με αυτά που εφαρμόζονται κατά την επίλυση του *LMCKP* (§4.4)):

- Τα κριτήρια αυτά προτάθηκαν για πρώτη φορά από τους Sinha και Zoltners [109], ενώ ο ρόλος τους είναι η απομάκρυνση εκείνων των υπηρεσιών από τον κάθε πελάτη, οι οποίες δε μπορούν να συμβάλλουν στον υπολογισμό του βέλτιστου κέρδους του παρόχου. Το πρώτο κριτήριο είναι το ακόλουθο:

$$\text{Αν } p_{ij} \geq p_{ik} \text{ και } D_{ij} \geq D_{ik} \text{ τότε } S_{ik} = 0 \text{ όπου } i = 1, 2, \dots, N$$

Υπενθυμίζεται ότι δείκτης i παριστάνει τον εκάστοτε πελάτη, ενώ οι δείκτες j, k παριστάνουν υπηρεσίες που ικανοποιούν αυτόν τον πελάτη. Ακόμη, η μεταβλητή p παριστάνει το κέρδος του παρόχου από τις υπηρεσίες, ενώ η μεταβλητή D παριστάνει τη «συμφωνημένη» *Καθυστέρηση* που θα προσφέρεται η υπηρεσία στον πελάτη. Τέλος, η μεταβλητή S_{ik} υποδεικνύει αν η υπηρεσία k που προσφέρεται στον πελάτη i επιλέγεται ή όχι. Σε περίπτωση που επιλέγεται, παίρνει την τιμή 1, αλλιώς την τιμή 0.

Πρακτικά το κριτήριο αυτό απομακρύνει από τον κάθε πελάτη τις υπηρεσίες εκείνες οι οποίες ενώ έχουν μικρότερο κέρδος για τον πάροχο, έχουν ταυτόχρονα και μικρότερη *Καθυστέρηση* και κατά συνέπεια δε μπορούν να οδηγήσουν στη βέλτιστη λύση, λαμβάνοντας υπόψη τη συνθήκη περιορισμού (σχέση (5-9)).

Η εφαρμογή του δεύτερου κριτηρίου επικράτησης στον κάθε πελάτη θα απομακρύνει τις υπηρεσίες εκείνες που βρίσκονται «εκτός» της καμπύλης, όπως αναλύθηκε και στο προηγούμενο κεφάλαιο (Σχήμα 4-1). Αυτό έχει σαν αποτέλεσμα να παραμένουν οι υπηρεσίες εκείνες που ενώ το κέρδος τους αυξάνεται με μεγάλο «ρυθμό», η *Καθυστέρησή* τους μειώνεται με μικρό «ρυθμό».

Το κριτήριο αυτό είναι:

$$\text{Αν } p_{it} \geq p_{is} \geq p_{ir} \text{ και } \left| \frac{p_{is} - p_{ir}}{D_{is} - D_{ir}} \right| \leq \left| \frac{p_{it} - p_{is}}{D_{it} - D_{is}} \right|$$

$$\text{τότε } S_{is} = 0 \text{ με } i = 1, 2, \dots, N$$

Βήμα 2 (Υπολογισμός αρχικής λύσης και ενός πρώτου κάτω ορίου $Lower_Bound_1$):

- Επιλέγω από τον κάθε πελάτη i την υπηρεσία που έχει τη μεγαλύτερη *Καθυστέρηση*, έστω S_{ik} :

$$D_{ik} = \max \sum_{j=1}^{M_i} D_{ij} \quad \text{για } i = 1, 2, \dots, N$$

- Διατάσσω αυτές τις υπηρεσίες και για όλους τους πελάτες σε φθίνουσα σειρά ως προς την *Καθυστέρηση*:

$$D_{1k} \geq D_{2k} \geq \dots \geq D_{Nk}$$

- Επιλέγω διαρκώς υπηρεσίες μέχρι να ξεπεραστεί η συνθήκη περιορισμού (σχέση (5-9)). Αν $m' < N$ ο «πρώτος» πελάτης του οποίου η *Καθυστέρηση* υπερβαίνει τη συνθήκη περιορισμού, τότε θα πρέπει να ισχύει:

$$\frac{m'}{D_{m'k}} > \frac{1}{D_p}$$

ενώ αν $m = m' - 1$ το μέγιστο πλήθος πελατών που δεν υπερβαίνει τη συνθήκη αυτή, θα ισχύει:

$$\frac{m}{D_{mk}} \leq \frac{1}{D_p}$$

όπου όπως έχει ήδη αναφερθεί D_p είναι η σταθερή *Καθυστέρηση* του παρόχου.

Το κέρδος του παρόχου δίνεται τότε από τη σχέση:

$$Lower_Bound_1 = \sum_{i=1}^m p_{ik}$$

- Επισημαίνεται ότι σε περίπτωση που οι υπηρεσίες m και m' έχουν την ίδια *Καθυστέρηση*, τότε επιλέγεται η υπηρεσία με το μεγαλύτερο κέρδος p .

Βήμα 3 (Υπολογισμός αυστηρότερου κάτω ορίου $Lower_Bound_2$):

- Υπολογίζω για όλες τις υπηρεσίες που έχουν παραμείνει σε κάθε πελάτη μετά την εφαρμογή των κριτηρίων επικράτησης το λόγο:

$$\lambda_{ij} = \frac{p_{ij} - p_{ij-1}}{|D_{ij} - D_{ij-1}|}$$

- Ταξινομώ όλους τους λόγους όλων των πελατών σε φθίνουσα σειρά
- Επιλέγω διαδοχικά λ_{ij} βελτιώνοντας την αρχική λύση μέχρι να ξεπεραστεί η συνθήκη περιορισμού (σχέση (5-9)). Όπως και στο προηγούμενο βήμα, αν $l' < N$ (όπου $l' \geq m'$ για λόγους που θα εξηγηθούν στη συνέχεια) ο «πρώτος»

πελάτης του οποίου η *Καθυστέρηση* υπερβαίνει τη συνθήκη περιορισμού, τότε θα πρέπει να ισχύει:

$$\frac{l'}{D'_{i\min}} \succ \frac{1}{D_p}$$

όπου $D'_{i\min}$ η υπηρεσία που προσφέρεται στον πελάτη i , έχει τη μικρότερη *Καθυστέρηση* και συμμετέχει στη λύση αυτή. Ομοίως, αν $l = l' - 1$ το μέγιστο πλήθος πελατών που δεν υπερβαίνει τη συνθήκη αυτή, θα ισχύει:

$$\frac{l}{D_{i\min}} \leq \frac{1}{D_p}$$

όπου $D_{i\min}$ η υπηρεσία που προσφέρεται στον πελάτη i , έχει τη μικρότερη *Καθυστέρηση* και συμμετέχει στη λύση αυτή, ενώ προφανώς η *Καθυστέρηση* αυτή μπορεί να είναι ίση με την προηγούμενη. Το κέρδος του παρόχου που θα υπολογιστεί μέσω της διαδικασίας αυτής θα είναι το $Lower_Bound_2$.

- Επισημαίνεται σε αυτό το σημείο, ότι αντίθετα με τις μεθόδους που εφαρμόζονται κατά την επίλυση του *MCKP*, όπου όλοι οι «όροι» συμμετέχουν από την αρχή στην αρχική λύση, για λόγους που έχουν γίνει προφανείς στην προηγούμενη παράγραφο, κάτι τέτοιο δεν ήταν εφικτό να ακολουθηθεί και στην περίπτωση αυτή. Πράγματι σε αυτό το βήμα, υπάρχει περίπτωση ο λόγος λ_{ij} να αναφέρεται σε υπηρεσίες κάποιου πελάτη που δε συμμετείχε στην αρχική λύση του βήματος 2. Σε αυτή την περίπτωση, επιλέγεται να συμμετέχει μεταξύ των 2 υπηρεσιών που απαρτίζουν το λόγο αυτό, η υπηρεσία με τη μικρότερη *Καθυστέρηση* εκ των δύο, κάτι που θα επιτρέψει την ομαλή συνέχιση της διαδικασίας υπολογισμού του επιθυμητού κέρδους. Εξάλλου, αποτέλεσμα αυτού είναι η αύξηση του πλήθους των πελατών που θα εξυπηρετηθούν από τον πάροχο (οπότε θα ισχύει και $l > m$).

Βήμα 4 (Υπολογισμός του αυστηρότερου κάτω ορίου $Lower_Bound_3$):

- Σύμφωνα με το προηγούμενο βήμα, η υπηρεσία με τη μικρότερη *Καθυστέρηση* που συμμετέχει στον υπολογισμό του κέρδους του παρόχου, είναι η υπηρεσία με *Καθυστέρηση* έστω $D_{i\min}$. Αν τότε από τον κάθε πελάτη που συμμετέχει στη λύση αυτή, και εφόσον η υπηρεσία που έχει επιλεγεί για αυτόν έχει *Καθυστέρηση* μεγαλύτερη από $D_{i\min}$ (κάτι που θα ισχύει για τους περισσότερους πελάτες προφανώς), αντικατασταθεί η υπηρεσία που του προσφέρεται με την υπηρεσία που έχει *Καθυστέρηση* ακριβώς ίδια ή λίγο μεγαλύτερη *Καθυστέρηση* από την $D_{i\min}$, δηλαδή με υπηρεσία μικρότερης

Καθυστέρησης από την ήδη επιλεγείσα, τότε οι υπηρεσίες που θα συμμετέχουν στη λύση αυτή θα βελτιώνουν το κέρδος του παρόχου, δίνοντας ένα αυστηρότερο κάτω όριο $Lower_Bound_3$. Προφανώς η επιλογή των αντικαταστάσεων για τον κάθε πελάτη θα γίνει μεταξύ των υπηρεσιών που ανταποκρίνονται στις απαιτήσεις του, ενώ η ιδέα της αντικατάστασης είναι συνέπεια του πρώτου κριτηρίου επικράτησης και του τρόπου που έχει οριστεί η συνθήκη περιορισμού για την περίπτωση της σταθερής *Καθυστέρησης*. Πράγματι, η εφαρμογή του πρώτου κριτηρίου επικράτησης όπως ήδη αναφέρθηκε έχει σαν αποτέλεσμα να παραμείνουν σε κάθε πελάτη υπηρεσίες που ενώ αυξάνεται το κόστος τους μειώνεται ταυτόχρονα η *Καθυστέρηση* τους. Λαμβάνοντας ακόμα κανείς υπόψη του τρόπου με τον οποίο έχει οριστεί η συνθήκη περιορισμού, είναι εύλογο ότι αν αντικατασταθεί η υπηρεσία κάποιου πελάτη με κάποια άλλη με μεγαλύτερο κέρδος για τον πάροχο και η οποία δε θα είναι μικρότερη από το $D_{i\min}$, τότε είναι προφανές ότι η αντικατάσταση αυτή είναι προς όφελος του παρόχου, αυξάνοντας το κέρδος του. Δηλαδή εφόσον στη λύση του προηγούμενου βήματος ισχύει:

$$\frac{l}{D_{i\min}} \leq \frac{1}{D_p}$$

τότε κρατώντας σταθερό το πλήθος l και την *Καθυστέρηση* $D_{i\min}$ και βελτιώνοντας τις υπόλοιπες υπηρεσίες με τον τρόπο που ήδη αναφέρθηκε, είναι βέβαιο ότι το νέο όριο $Lower_Bound_3$ θα είναι μεγαλύτερο ή ίσο από το προηγούμενο. Δηλαδή θα ισχύει:

$$Lower_Bound_3 \geq Lower_Bound_2$$

Η περίπτωση ισότητας είναι η περίπτωση όπου οι πελάτες που συμμετέχουν στο $Lower_Bound_2$ δε διαθέτουν κάποια υπηρεσία μεταξύ $D_{i\min}$ και εκείνης που έχει ήδη επιλεγεί για αυτούς.

Βήμα 5 (Υπολογισμός μέγιστου κέρδους παρόχου Max_Profit):

- Ο αλγόριθμος ολοκληρώνεται εξετάζοντας όλους τους λόγους λ_{ij} που υπολογίστηκαν στο βήμα 3. Πιο αναλυτικά και με τη μέθοδο της εξαντλητικής προσέγγισης, ελέγχονται όλες οι υπηρεσίες όλων των πελατών που έχουν παραμείνει μετά την εφαρμογή των προηγούμενων βημάτων, προκειμένου να αυξηθεί το κέρδος του παρόχου. Αυτό το βήμα είναι απαραίτητο προκειμένου να αντισταθμιστεί η «αδυναμία» της αρχικής λύσης, όπου επιλέχθηκαν υπηρεσίες με τη μεγαλύτερη *Καθυστέρηση*. Πράγματι, σε περίπτωση όπου

κάποια υπηρεσία με πολύ μεγάλο κέρδος και πολύ μικρή *Καθυστέρηση* βελτιώνει το κέρδος του παρόχου, τότε υπάρχει περίπτωση να προκληθεί «μεγάλη» μείωση του πλήθους των πελατών που θα εξυπηρετηθούν τελικά. Βέβαια αυτή η περίπτωση είναι εξαιρετικά σπάνια, κάτι που άλλωστε επιβεβαιώνεται και στις μετρήσεις που ακολουθούν. Η διαδικασία που ακολουθείται στο βήμα αυτό είναι:

- Ταξινόμηση όλων των πελατών που συμμετέχουν στη λύση σε αύξουσα σειρά ως προς το κέρδος:

$$p_{1k} \geq p_{2k} \geq \dots \geq p_{lk}$$

- Έλεγχος κάθε υπηρεσίας που συμμετέχει στο λόγο λ_{ij} αν βελτιώνει το κέρδος του παρόχου – απορρίπτοντας ενδεχομένως κάποιους πελάτες – και δεν παραβιάζει τη συνθήκη περιορισμού. Έστω δηλαδή ότι:

$$Max_Profit = \sum_{i=1}^l p_{li}$$

τότε αν κάποια υπηρεσία που έχει *Καθυστέρηση* D_o και κέρδος p_o τέτοια ώστε:

$$Max_Profit_o = \sum_{i=1}^{k'} p_{li} + p_o \geq Max_Profit$$

$$\frac{k'+1}{D_o} \leq \frac{1}{D_p}$$

όπου $k' < l$, τότε προστίθεται αυτή η υπηρεσία στην τελική λύση, απομακρύνοντας όλες τις υπηρεσίες με δείκτη $i > k'$. Προσοχή πρέπει να δοθεί βέβαια σε περίπτωση που ο πελάτης αυτός συμμετέχει ήδη στη λύση, όπως αυτή έχει υπολογιστεί από τον αλγόριθμο *SMCK*. Σε αυτή την περίπτωση, οι παραπάνω σχέσεις έχουν τη μορφή:

$$Max_Profit_o = \sum_{i=1}^{k'} p_{li} + p_o - p_{ln} \geq Max_Profit$$

$$\frac{k'}{D_o} \leq \frac{1}{D_p}$$

όπου p_{ln} ($n < k'$) το κέρδος του παρόχου από την υπάρχουσα υπηρεσία για τον πελάτη που έχει επιλεγθεί να αντικατασταθεί. Επισημαίνεται ότι η υπηρεσία / πελάτης που θα προστεθούν στην τελική λύση μπορούν να διαθέτουν *Καθυστέρηση* μεγαλύτερη από D_{min} όπου D_{min} η μικρότερη *Καθυστέρηση* κάποιας υπηρεσίας που συμμετέχει στην προτεινόμενη λύση.

Σε αυτή την περίπτωση όπως είναι εύλογο μπορεί να μη χρειαστεί να απομακρυνθούν πελάτες από την τελική λύση.

Όσον αφορά την πολυπλοκότητα του αλγορίθμου, ο θεωρητικός υπολογισμός της είναι μεταξύ $O(n \log n)$ και $O(n^2)$. Αυτό οφείλεται κυρίως στο βήμα 1 όπου γίνεται η εφαρμογή των κριτηρίων επικράτησης αλλά και στο βήμα 3 κατά τον υπολογισμό του λόγου λ_{ij} , αφού τόσο ο αλγόριθμος ταξινόμησης που χρησιμοποιείται (QuickSort) όσο και ο υπολογισμός των λόγων λ_{ij} έχουν πολυπλοκότητα που κυμαίνεται μεταξύ των τιμών αυτών ([82], [110], [111]), ενώ η επιβάρυνση στην πολυπλοκότητα του αλγορίθμου από ταξινομήσεις ή άλλες ενέργειες των υπηρεσιών μπορεί να αγνοηθεί με ασφάλεια. Κατά συνέπεια, ο *SMCK* είναι πολυωνυμικής πολυπλοκότητας και μπορεί να χρησιμοποιηθεί με ασφάλεια ακόμη και στην περίπτωση που υπάρχει πολύ μεγάλο πλήθος πελατών. Όλα αυτά βέβαια επιβεβαιώνονται και πειραματικά στις μετρήσεις που ακολουθούν.

5.6. Πολλαπλοί περιορισμοί

Στην προηγούμενη παράγραφο παρουσιάστηκαν αλγόριθμοι οι οποίοι στοχεύουν στη μεγιστοποίηση του κέρδους ενός παρόχου, εξετάζοντας 3 διαφορετικές περιπτώσεις, ανάλογα με την παράμετρο ποιότητας που συμμετείχε στη συνθήκη περιορισμού. Υπάρχει όμως το ενδεχόμενο, ικανοποιώντας την παράμετρο αυτή να μην ικανοποιούνται ταυτόχρονα και οι περιορισμοί για τις άλλες δύο παραμέτρους. Αντικείμενο αυτής της παραγράφου είναι η παρουσίαση των βημάτων που μπορούν να ακολουθηθούν, προκειμένου να ικανοποιούνται ταυτόχρονα οι περιορισμοί και για τις 3 παραμέτρους. Ένας τρόπος για να επιτευχθεί αυτό είναι να ακολουθηθούν απόλυτα τα βήματα που αναλύθηκαν στην παράγραφο 5.5 για τη μια παράμετρο (αυτή που θεωρεί ο πάροχος πιο σημαντική για αυτόν) και κατόπιν να απορριφθούν κάποιες υπηρεσίες, προκειμένου να ικανοποιηθούν και οι υπόλοιποι περιορισμοί. Ο μηχανισμός με τους οποίους θα πρέπει να απομακρυνθούν οι υπηρεσίες αυτές είναι διαφορετικός κατά περίπτωση, όπως άλλωστε θα φανεί και στη συνέχεια. Προτού αναλυθούν τα βήματα αυτά, επισημαίνεται ότι η εξέταση του περιορισμού της παραμέτρου της *Χωρητικότητας* θα είναι πάντα το τελευταίο βήμα που θα πρέπει να γίνει, ενώ σημειώνεται ότι δε θα εξεταστεί η περίπτωση που επιλέγεται από τον πάροχο η παράμετρος αυτή ως η πρώτη παράμετρος της οποίας θα πρέπει να ικανοποιηθεί ο περιορισμός. Αυτό οφείλεται στο γεγονός ότι η εξάντληση των διαθέσιμων πόρων ενός παρόχου προκαλείται σχεδόν πάντα από τις άλλες δύο παραμέτρους.

Διακρίνονται επομένως οι ακόλουθες δύο περιπτώσεις:

Αρχικός περιορισμός - σταθερή Καθυστέρηση D_p :

Βήμα 1 (Υπολογισμός μέγιστου κέρδους παρόχου Max_Profit με συνθήκη περιορισμού που αφορά την παράμετρο της Καθυστέρησης):

- Σε αυτό το βήμα ακολουθείται απόλυτα η διαδικασία που αναλύεται εκτενώς στην προηγούμενη παράγραφο για την Περίπτωση της σταθερής Καθυστέρησης D_p . Όπως αναλύεται και στην παράγραφο αυτή, έστω Max_Profit το μέγιστο κέρδος του παρόχου που υπολογίζεται κατά την εφαρμογή του προτεινόμενου αλγορίθμου.

Βήμα 2 (Υπολογισμός νέου κέρδους παρόχου Max_Profit' με συνθήκη περιορισμού που αφορά την παράμετρο του Εύρους Ζώνης):

- Σε περίπτωση που ικανοποιείται ο περιορισμός του μέγιστου Εύρους Ζώνης BW_p για τις υπηρεσίες (έστω πλήθους N) που έχουν επιλεγθεί στο προηγούμενο βήμα, δηλαδή σε περίπτωση που ισχύει:

$$\sum_{i=1}^N BW_i \leq BW_p$$

τότε η διαδικασία συνεχίζει στο Βήμα 3, ενώ για το νέο κέρδος παρόχου θα ισχύει προφανώς :

$$Max_Profit' = Max_Profit$$

- Σε διαφορετική περίπτωση, θα πρέπει να απομακρυνθούν υπηρεσίες ώστε να ικανοποιείται ο περιορισμός αυτός (BW_p). Το πρόβλημα όμως που πρέπει να αντιμετωπιστεί σε αυτή την περίπτωση είναι το 0-1 KP, ενώ προτεινόμενες μέθοδοι που το λύνουν, έχουν παρουσιαστεί εκτενώς στο προηγούμενο κεφάλαιο (§4.3) ([95], [97], [130]). Έστω Max_Profit' το νέο κέρδος του παρόχου που υπολογίζεται με κάποια από τις μεθόδους αυτές.

Βήμα 3 (Υπολογισμός τελικού κέρδους παρόχου $Final_Max_Profit$ με συνθήκη περιορισμού που αφορά την παράμετρο της Χωρητικότητας):

- Σε περίπτωση που ικανοποιείται ο περιορισμός της μέγιστης Χωρητικότητας C_p για τις υπηρεσίες (έστω πλήθους N) που έχουν επιλεγθεί στα προηγούμενα βήματα, δηλαδή σε περίπτωση που ισχύει:

$$N \leq C_p$$

η διαδικασία τερματίζεται, ενώ το μέγιστο κέρδος του παρόχου θα ισούται προφανώς με το κέρδος που υπολογίστηκε στο προηγούμενο βήμα, δηλαδή θα ισχύει:

$$Final_Max_Profit = Max_Profit'$$

- Σε διαφορετική περίπτωση θα πρέπει να απομακρυνθούν υπηρεσίες ώστε να ικανοποιείται ο περιορισμός αυτός (C_p). Η μέθοδος που θα εφαρμοστεί για να επιλεγθούν ποιες υπηρεσίες πρέπει να απομακρυνθούν είναι παρόμοια με αυτή που εφαρμόζεται στην προηγούμενη παράγραφο.
- Διατάσσω δηλαδή αρχικά τις υπηρεσίες (έστω πλήθους N) που έχουν παραμείνει από τα προηγούμενα βήματα σε φθίνουσα σειρά ως προς το κέρδος:

$$p_1 \geq p_2 \geq \dots \geq p_N$$

- Επιλέγω κατόπιν τις πρώτες C_p πλήθος υπηρεσίες από την παραπάνω διάταξη, με αποτέλεσμα το κέρδος του παρόχου να δίνεται τελικά από τη σχέση:

$$Final_Max_Profit = \sum_{i=1}^{C_p} p_i$$

Αρχικός περιορισμός – μέγιστο Εύρος Ζώνης BW_p :

Βήμα 1 (Υπολογισμός μέγιστου κέρδους παρόχου Max_Profit με συνθήκη περιορισμού που αφορά την παράμετρο του *Εύρους Ζώνης*):

- Σε αυτό το βήμα ακολουθείται απόλυτα η διαδικασία που αναλύεται εκτενώς στην προηγούμενη παράγραφο για την *Περίπτωση του μέγιστου Εύρους Ζώνης BW_p* . Όπως αναλύεται και στην παράγραφο αυτή, έστω Max_Profit το μέγιστο κέρδος του παρόχου που υπολογίζεται κατά την εφαρμογή του προτεινόμενου αλγορίθμου.

Βήμα 2 (Υπολογισμός νέου κέρδους παρόχου Max_Profit' με συνθήκη περιορισμού που αφορά την παράμετρο της *Καθυστερήσης*):

- Σε περίπτωση που ικανοποιείται ο περιορισμός της σταθερής *Καθυστερήσης D_p* για τις υπηρεσίες (έστω πλήθους N) που έχουν επιλεγθεί στο προηγούμενο βήμα, δηλαδή σε περίπτωση που ισχύει:

$$\frac{N}{D_{\min}} \leq \frac{1}{D_p}$$

τότε η διαδικασία συνεχίζει στο Βήμα 3, ενώ για το νέο κέρδος παρόχου θα ισχύει προφανώς :

$$Max_Profit' = Max_Profit$$

- Σε διαφορετική περίπτωση θα πρέπει να απομακρυνθούν υπηρεσίες ώστε να ικανοποιείται ο περιορισμός αυτός (D_p). Η μέθοδος που θα εφαρμοστεί για

να επιλεγθούν ποιες υπηρεσίες πρέπει να απομακρυνθούν είναι παρόμοια με αυτή που εφαρμόζεται στην προηγούμενη παράγραφο. Αναλυτικά έχουμε:

- Εφαρμόζω το πρώτο κριτήριο επικράτησης [109] για τις υπηρεσίες που έχουν επιλεγθεί στο προηγούμενο βήμα. Υπάρχουν δύο περιπτώσεις:

- (i) Ο περιορισμός ικανοποιείται. Δηλαδή αν έχουν παραμείνει N' πλήθος υπηρεσίες με $D'_{\min}$ η μικρότερη *Καθυστέρηση* από όλες αυτές τις υπηρεσίες, να ισχύει:

$$\frac{N'}{D'_{\min}} \leq \frac{1}{D_p}$$

Σε αυτή την περίπτωση θα πρέπει να προστεθούν κάποιες υπηρεσίες από αυτές που απομακρύνθηκαν. Οι υπηρεσίες που θα επιλεγθούν να ενταχθούν ξανά θα βρεθούν με τη μέθοδο της εξαντλητικής προσέγγισης, αφού είναι βέβαιο ότι το πλήθος τους θα είναι πολύ μικρό, εξασφαλίζοντας με αυτόν τον τρόπο το μέγιστο κέρδος για τον πάροχο.

- (ii) Ο περιορισμός εξακολουθεί να μην ικανοποιείται με αποτέλεσμα να πρέπει να απομακρυνθούν και άλλες υπηρεσίες. Διατάσσονται οι υπηρεσίες που έχουν παραμείνει (έστω πλήθους N') ως προς την καθυστέρηση ως εξής:

$$D_1 \geq D_2 \geq \dots \geq D_{N'}$$

και απομακρύνονται διαρκώς υπηρεσίες μεγαλύτερης *Καθυστέρησης* μέχρις ότου να ικανοποιηθεί ο περιορισμός. Είναι εύλογο ότι για υπηρεσίες ίδιας *Καθυστέρησης* επιλέγεται πρώτα η απομάκρυνση υπηρεσιών μικρότερου κέρδους για τον πάροχο. Έστω *Max_Profit* το κέρδος του παρόχου που θα υπολογιστεί με τη μέθοδο αυτή. Η επιλογή αυτής της μεθόδου απομάκρυνσης στηρίζεται στην ακόλουθη παρατήρηση: για να ικανοποιηθεί ο περιορισμός της συνθήκης *Καθυστέρησης* (σχέση (5-9)) αρκεί να μειωθεί το πλήθος των πελατών (αριθμητής) ή να αυξηθεί η ελάχιστη *Καθυστέρηση* (παρονομαστής). Επειδή ωστόσο συνήθως θα υπάρχουν πολλές υπηρεσίες με *Καθυστέρηση* ίση με τη μικρότερη, είναι προφανές ότι η απομάκρυνση τέτοιων υπηρεσιών δε θα καταφέρουν να ικανοποιήσουν τον περιορισμό – εκτός και αν απομακρυνθούν όλες – αλλά να ελαττώσουν μόνο το κέρδος του παρόχου. Αν αναλογιστεί κανείς μάλιστα ότι οι υπηρεσίες αυτές είναι και οι υπηρεσίες με το μεγαλύτερο κέρδος, θα

διαπιστώσει εύκολα την ορθότητα αυτής της μεθόδου απομάκρυνσης. Εξάλλου με την τεχνική αυτή ικανοποιείται ταχύτερα και ο περιορισμός της *Χωρητικότητας* C_p .

Βήμα 3 (Υπολογισμός τελικού κέρδους παρόχου *Final_Max_Profit* με συνθήκη περιορισμού που αφορά την παράμετρο της *Χωρητικότητας*):

- Σε περίπτωση που ικανοποιείται ο περιορισμός της μέγιστης *Χωρητικότητας* C_p για τις υπηρεσίες (έστω πλήθους N) που έχουν επιλεχθεί στα προηγούμενα βήματα, δηλαδή σε περίπτωση που ισχύει:

$$N \leq C_p$$

η διαδικασία τερματίζεται, ενώ το μέγιστο κέρδος του παρόχου θα ισούται προφανώς με το κέρδος που υπολογίστηκε στο προηγούμενο βήμα, δηλαδή θα ισχύει:

$$Final_Max_Profit = Max_Profit'$$

- Σε διαφορετική περίπτωση θα πρέπει να απομακρυνθούν υπηρεσίες ώστε να ικανοποιείται ο περιορισμός αυτός (C_p). Η μέθοδος που θα εφαρμοστεί για να επιλεχθούν ποιες υπηρεσίες πρέπει να απομακρυνθούν είναι ίδια με αυτή που εφαρμόστηκε στην προηγούμενη περίπτωση.
- Διατάσσω δηλαδή αρχικά τις υπηρεσίες (έστω πλήθους N) που έχουν παραμείνει από τα προηγούμενα βήματα σε φθίνουσα σειρά ως προς το κέρδος:

$$p_1 \geq p_2 \geq \dots \geq p_N$$

- Επιλέγω κατόπιν τις πρώτες C_p πλήθος υπηρεσίες από την παραπάνω διάταξη, με αποτέλεσμα το κέρδος του παρόχου να δίνεται τελικά από τη σχέση:

$$Final_Max_Profit = \sum_{i=1}^{C_p} p_i$$

5.7. Αποτίμηση επίδοσης προτεινόμενου αλγόριθμου

Στόχος των μετρήσεων που ακολουθούν είναι να αποτιμήσουν την επίδοση του προτεινόμενου αλγορίθμου *SMCK*, μελετώντας παράλληλα (i) την επίδραση της μεταβολής της τιμής της σταθερής *Καθυστέρησης* του παρόχου D_p και (ii) την επίδραση της μεταβολής του πλήθους των υπηρεσιών που εξυπηρετούν τον κάθε πελάτη. Σε όλες τις μετρήσεις που έγιναν, εξετάστηκαν οι ακόλουθες περιπτώσεις όσον αφορά τη σχέση συσχέτισης μεταξύ της προσφερόμενης *Καθυστέρησης* της κάθε υπηρεσίας και του κέρδους του παρόχου από την

προσφορά της υπηρεσίας αυτής: (i) το κέρδος και η *Καθυστέρηση* ήταν ασυσχέτιστα μεταξύ τους, (ii) η τιμή του κέρδους σχετιζόταν με την τιμή της *Καθυστέρησης* με σχέση συσχέτισης περίπου 50% (ασθενής σχέση), και (iii) η τιμή του κέρδους σχετιζόταν με την τιμή του *Καθυστέρησης*, με σχέση συσχέτισης περίπου 70% (ισχυρή σχέση), ενώ το εύρος τιμών και για τα δύο αυτά μεγέθη ήταν μεταξύ [1-1000]. Επιπρόσθετα, σε όλες τις μετρήσεις που έγιναν, εξετάστηκαν οι περιπτώσεις για 10, 50, 100, 500 και 1.000 αιτήσεις. Επισημαίνεται ότι οι υπηρεσίες που ικανοποιούσαν την κάθε αίτηση επιλέχθηκαν από ένα σύνολο 10.000 υπηρεσιών, οι οποίες ήταν κοινές σε όλες τις μετρήσεις που αφορούσαν την ίδια σχέση συσχέτισης μεταξύ κέρδους και *Καθυστέρησης*. Ακόμη, ο χρόνος απόκρισης του αλγορίθμου, όπως αυτός καταγράφηκε στις μετρήσεις που ακολουθούν, δεν περιλαμβάνει το χρόνο αρχικοποίησής του, δηλαδή το χρόνο δημιουργίας αυτών των 10.000 υπηρεσιών καθώς και το χρόνο δημιουργίας των υπηρεσιών που εξυπηρετούσαν τον κάθε πελάτη. Τέλος, σε κάθε μέτρηση έγιναν 50 επαναλήψεις και οι τιμές που παρουσιάζονται στους παρακάτω πίνακες αντιπροσωπεύουν τη μέση τιμή των επαναλήψεων αυτών. Οι μετρήσεις έγιναν σε Laptop με επεξεργαστή Intel Pentium M 1.5GHz με 512 MB RAM, ενώ η ανάπτυξη και εκτέλεση του αλγορίθμου έγινε σε MATLAB v7.0 [112].

Παράμετροι πινάκων και διαγραμμάτων που ακολουθούν:

N : Πλήθος πελατών / αιτήσεων

k_i : Πλήθος υπηρεσιών που εξυπηρετούν τον κάθε πελάτη ($k_1=10$, $k_2=20$ και $k_3=50$)

D_p (Σταθερή Καθυστέρηση του παρόχου): Όταν επιλέγεται με τιμή $0.8 * D_p$, τότε κατά το βήμα 2 στην αρχική λύση θα περιλαμβάνονται [35-45]% πελάτες, Όταν επιλέγεται με τιμή D_p , τότε κατά το βήμα 2 στην αρχική λύση θα περιλαμβάνονται [45-55]% πελάτες, ενώ όταν επιλέγεται με τιμή $1.2 * D_p$, τότε κατά το βήμα 2 στην αρχική λύση θα περιλαμβάνονται [55-65]% πελάτες.

UC (Uncorrelated data): Το κέρδος και η προσφερόμενη Καθυστέρηση είναι ασυσχέτιστα μεγέθη

WC (Weakly uncorrelated data): Το κέρδος και η προσφερόμενη Καθυστέρηση έχουν ασθενή σχέση συσχέτισης

SC (Strongly uncorrelated data): Το κέρδος και η προσφερόμενη Καθυστέρηση έχουν ισχυρή σχέση συσχέτισης

$D \min_1$: Η «μικρότερη» ελάχιστη Καθυστέρηση που έχει υπολογιστεί για αυτή τη σχέση συσχέτισης και Καθυστέρηση παρόχου

$D \min_2$: Η «μεγαλύτερη» ελάχιστη Καθυστέρηση που έχει υπολογιστεί για αυτή τη σχέση συσχέτισης και Καθυστέρηση παρόχου

$D_{\max}$: Η «μεγαλύτερη» Καθυστέρηση που έχει υπολογιστεί για αυτή τη σχέση συσχέτισης και Καθυστέρηση παρόχου.

Το πρώτο μέγεθος που θα μελετηθεί είναι το κέρδος του παρόχου, όπως αυτό υπολογίζεται από τον προτεινόμενο αλγόριθμο *SMCK*. Στους πίνακες (Πίνακας 5-3 - Πίνακας 5-5) και στα σχήματα (Σχήμα 5-2 - Σχήμα 5-14) που ακολουθούν υπολογίζεται τόσο σε απόλυτα όσο και σε σχετικά μεγέθη το μέγιστο κέρδος του παρόχου.

Πιο αναλυτικά, όπως μπορεί να διαπιστώσει κανείς εύκολα από τους παρακάτω πίνακες, το κέρδος του παρόχου βελτιώνεται διαρκώς όσο αυξάνεται είτε το πλήθος των υπηρεσιών που εξυπηρετούν τον κάθε πελάτη, είτε η σταθερή Καθυστέρηση D_p του παρόχου, είτε η σχέση συσχέτισης μεταξύ κέρδους και Καθυστέρησης, όπως άλλωστε ήταν αναμενόμενο. Ακόμη, στα ακόλουθα σχήματα (Σχήμα 5-2 - Σχήμα 5-10) σκιαγραφείται το ποσοστό βελτίωσης του κέρδους του παρόχου σε σχέση με το κέρδος που υπολογίζεται από τον *SMCK* κατά το δεύτερο βήμα. Όπως ήταν αναμενόμενο, όσο αυξάνεται η σχέση συσχέτισης τόσο μειώνεται το ποσοστό αυτό, αφού για μεγαλύτερη σχέση συσχέτισης μειώνεται κατά πολύ το «εύρος» της ελάχιστης Καθυστέρησης (μέγεθος το οποίο μελετάται εκτενέστερα παρακάτω) και κατά συνέπεια το «εύρος» της απόκλισης του κέρδους των υπηρεσιών που συμμετέχουν στην τελική λύση. Τέλος, στα σχήματα που ακολουθούν συγκρίνονται τα κέρδη του παρόχου όπως αυτά υπολογίζονται από τον προτεινόμενο αλγόριθμο, είτε μεταβάλλοντας τη σταθερή Καθυστέρηση D_p του παρόχου για κάθε σχέση συσχέτισης (Σχήμα 5-11 - Σχήμα 5-13) είτε κρατώντας σταθερή την Καθυστέρηση αυτή και μεταβάλλοντας τη σχέση συσχέτισης (Σχήμα 5-14).

N	$0.8 * D_p (10^5)$			$D_p (10^5)$			$1.2 * D_p (10^5)$		
	k_1	k_2	k_3	k_1	k_2	k_3	k_1	k_2	k_3
10	0.027	0.030	0.030	0.036	0.042	0.043	0.039	0.046	0.048
50	0.127	0.131	0.154	0.149	0.152	0.175	0.191	0.195	0.205
100	0.255	0.263	0.269	0.288	0.316	0.337	0.364	0.379	0.383
500	1.159	1.168	1.201	1.437	1.475	1.499	1.805	1.829	1.891
1000	2.294	2.340	2.365	2.810	2.960	3.030	3.626	3.743	3.845

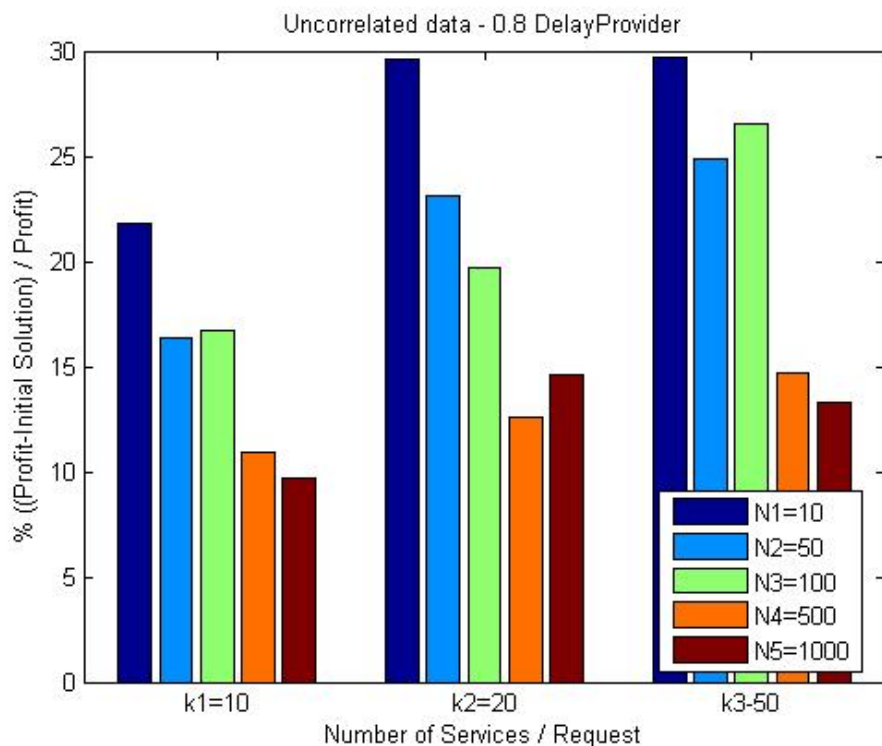
Πίνακας 5-3 Κέρδος παρόχου όταν η προσφερόμενη Καθυστέρηση και το κέρδος είναι ασυσχέτιστα μεταξύ τους

N	$0.8 * D_p(10^5)$			$D_p(10^5)$			$1.2 * D_p(10^5)$		
	k_1	k_2	k_3	k_1	k_2	k_3	k_1	k_2	k_3
10	0.033	0.034	0.035	0.041	0.043	0.044	0.049	0.050	0.051
50	0.154	0.156	0.158	0.194	0.194	0.198	0.229	0.235	0.238
100	0.304	0.305	0.306	0.386	0.386	0.388	0.451	0.461	0.469
500	1.488	1.512	1.559	1.881	1.907	1.945	2.280	2.308	2.411
1000	3.005	3.061	3.247	3.772	3.772	3.807	4.427	4.466	4.865

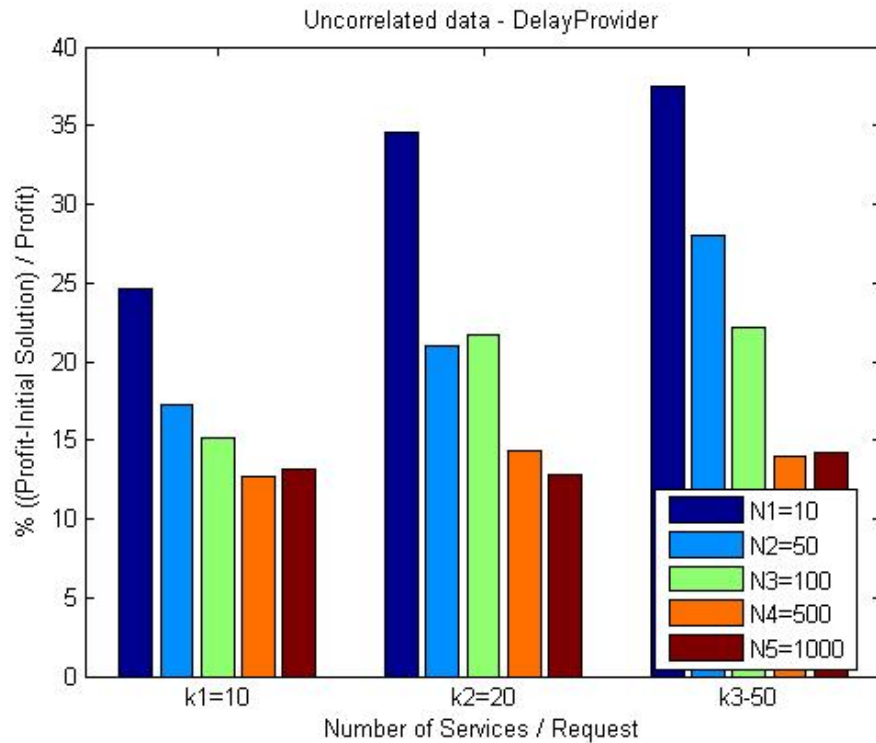
Πίνακας 5-4 Κέρδος παρόχου όταν η προσφερόμενη Καθυστέρηση και το κέρδος έχουν ασθενή σχέση συσχέτισης

N	$0.8 * D_p(10^5)$			$D_p(10^5)$			$1.2 * D_p(10^5)$		
	k_1	k_2	k_3	k_1	k_2	k_3	k_1	k_2	k_3
10	0.040	0.040	0.041	0.047	0.049	0.050	0.057	0.057	0.059
50	0.182	0.185	0.187	0.227	0.227	0.235	0.267	0.269	0.279
100	0.355	0.362	0.365	0.446	0.447	0.459	0.531	0.538	0.551
500	1.778	1.794	1.858	2.195	2.250	2.373	2.683	2.685	2.709
1000	3.499	3.533	3.747	4.458	4484	4.588	5.312	5.544	5.544

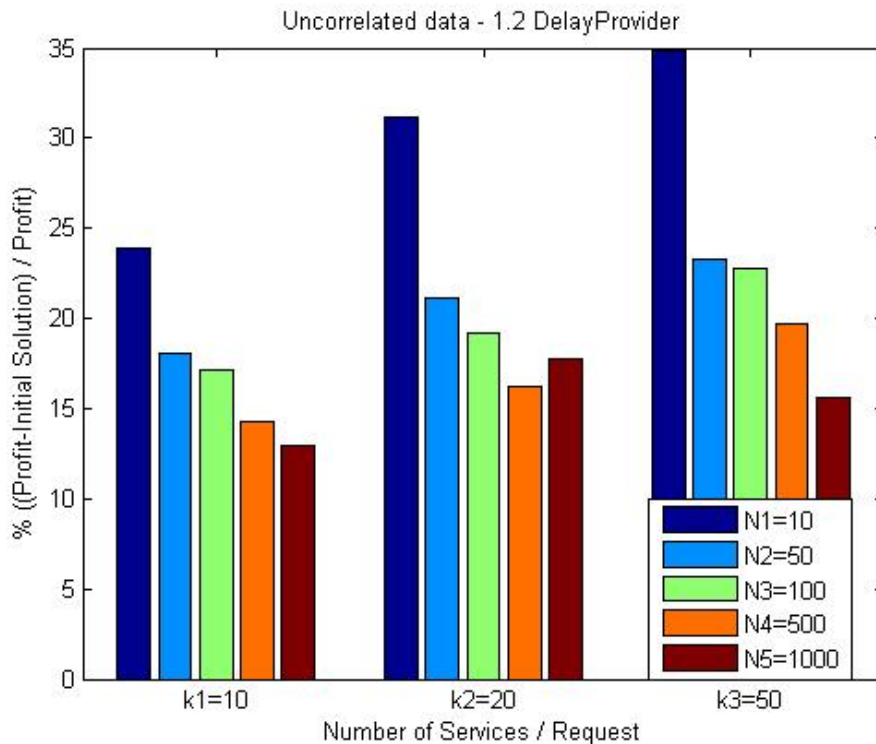
Πίνακας 5-5 Κέρδος παρόχου όταν η προσφερόμενη Καθυστέρηση και το κέρδος έχουν ισχυρή σχέση συσχέτισης



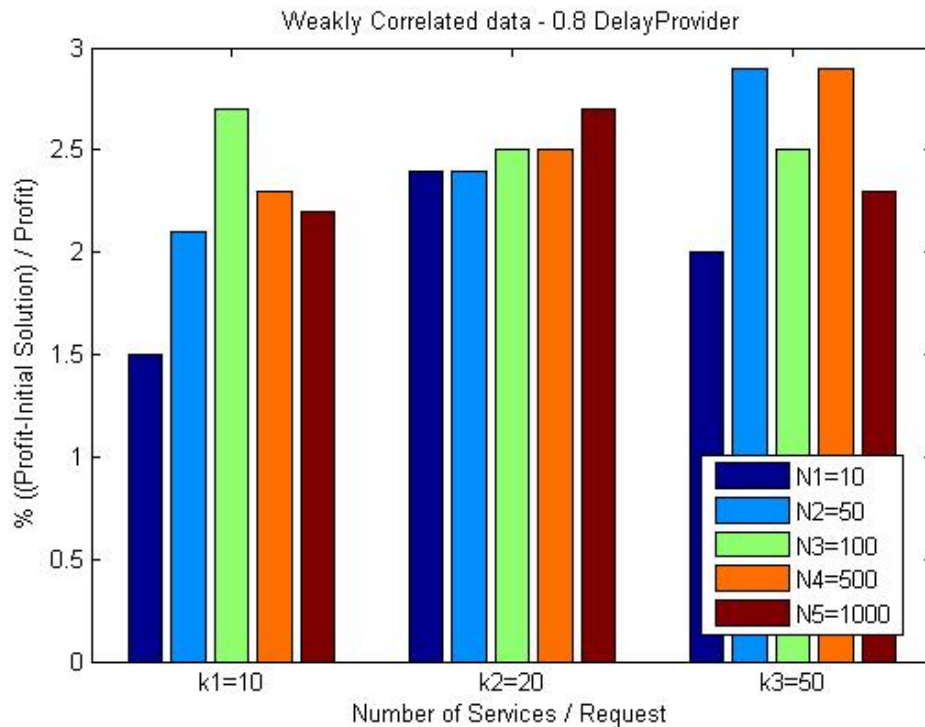
Σχήμα 5-2 Ποσοστό απόκλισης τελικού κέρδους σε σχέση με το αρχικό όταν η προσφερόμενη Καθυστέρηση και το κέρδος είναι ασυσχέτιστα μεταξύ τους και πλήθος πελατών=[35-45]%



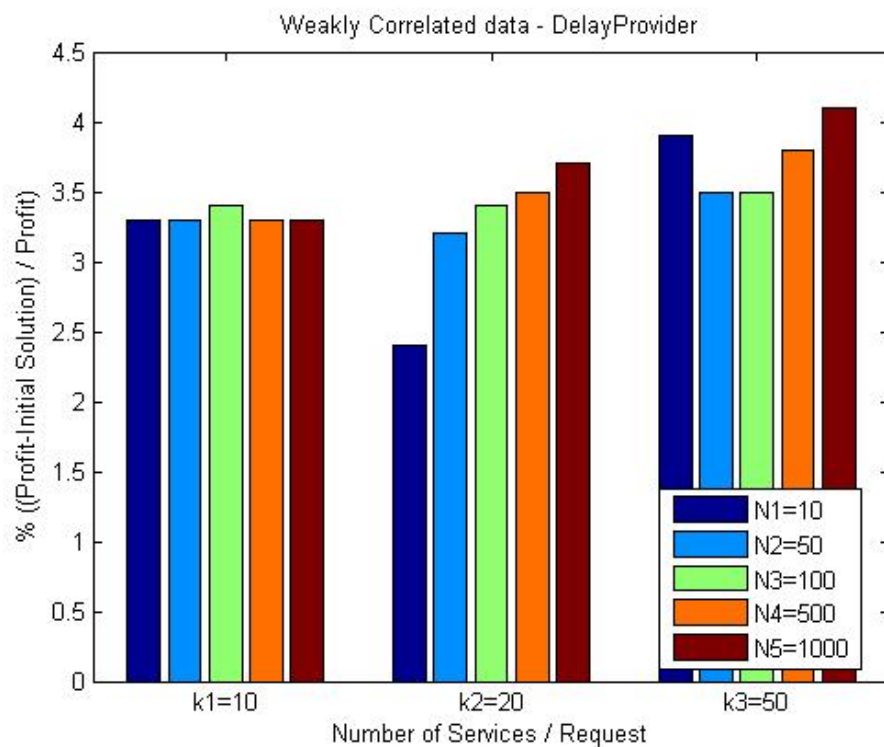
Σχήμα 5-3 Ποσοστό απόκλισης τελικού κέρδους σε σχέση με το αρχικό όταν η προσφερόμενη Καθυστερήση και το κέρδος είναι ασυσχέτιστα μεταξύ τους και πλήθος πελατών=[45-55]%



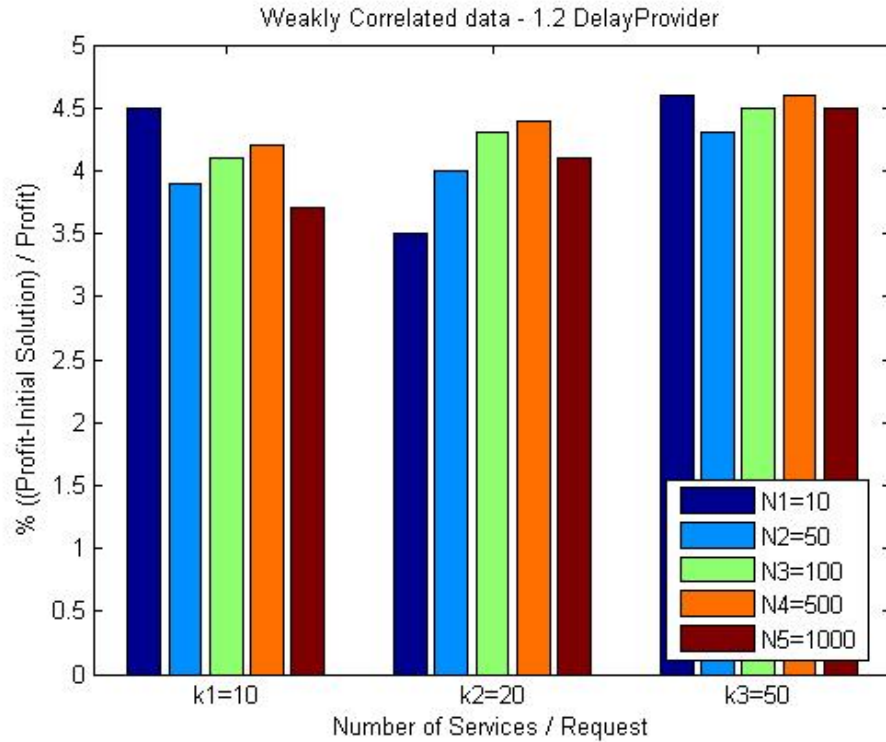
Σχήμα 5-4 Ποσοστό απόκλισης τελικού κέρδους σε σχέση με το αρχικό όταν η προσφερόμενη Καθυστερήση και το κέρδος είναι ασυσχέτιστα μεταξύ τους και πλήθος πελατών=[55-65]%



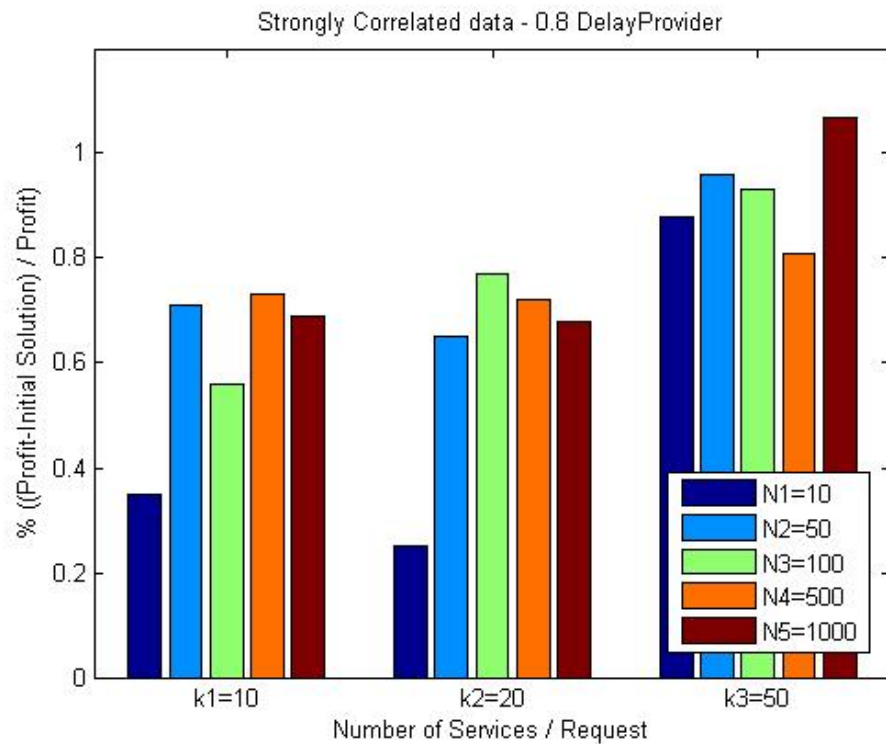
Σχήμα 5-5 Ποσοστό απόκλισης τελικού κέρδους σε σχέση με το αρχικό όταν η προσφερόμενη Καθυστέρηση και το κέρδος έχουν ασθενή σχέση συσχέτισης και πλήθος πελατών=[35-45]%



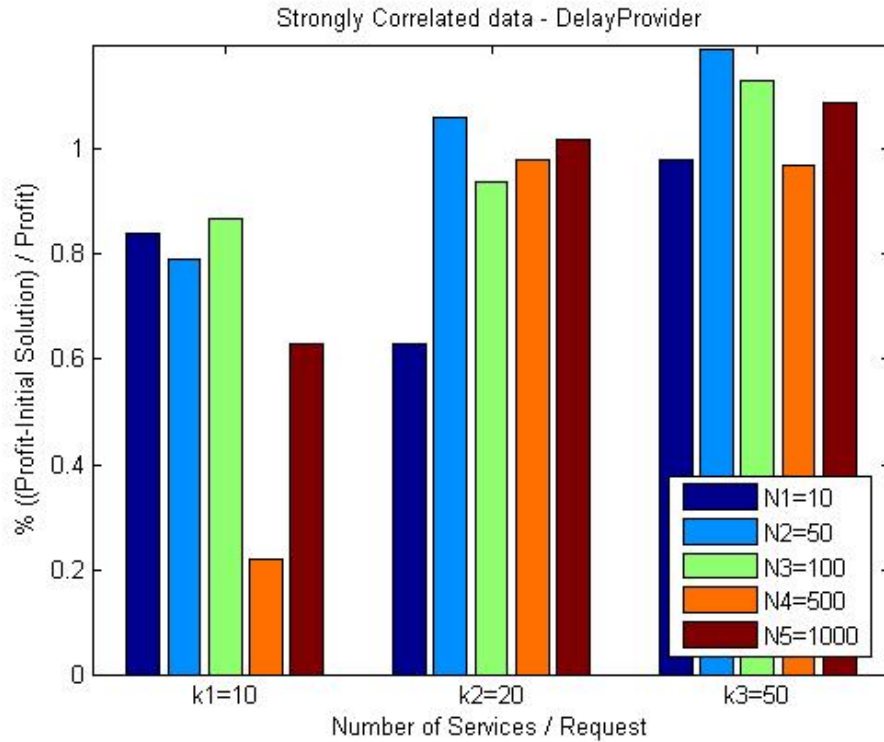
Σχήμα 5-6 Ποσοστό απόκλισης τελικού κέρδους σε σχέση με το αρχικό όταν η προσφερόμενη Καθυστέρηση και το κέρδος έχουν ασθενή σχέση συσχέτισης και πλήθος πελατών=[45-55]%



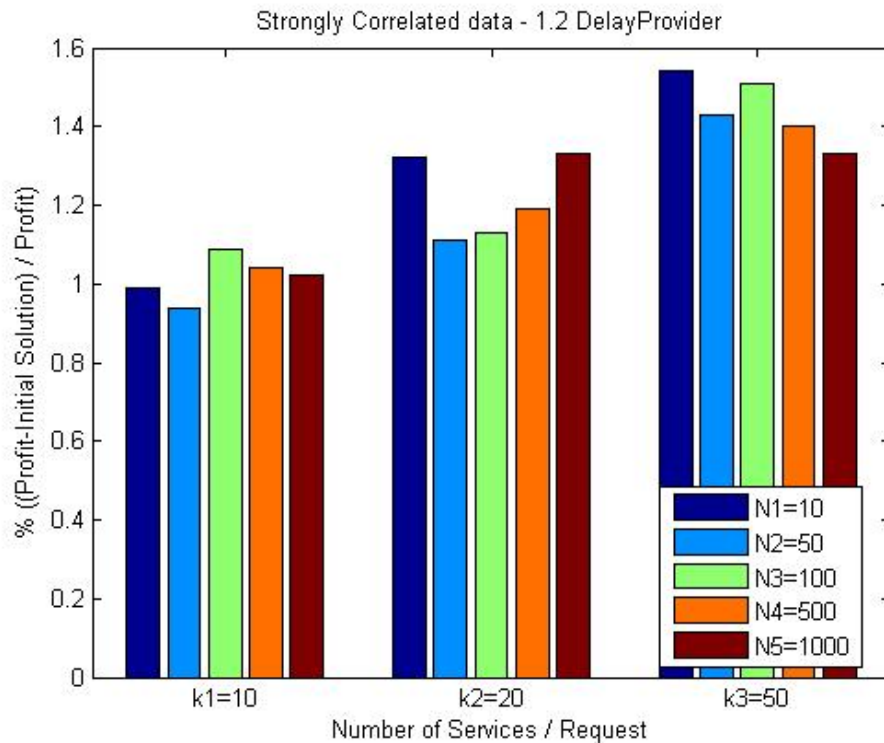
Σχήμα 5-7 Ποσοστό απόκλισης τελικού κέρδους σε σχέση με το αρχικό όταν η προσφερόμενη Καθυστερήση και το κέρδος έχουν ασθενή σχέση συσχέτισης και πλήθος πελατών=[55-65]%



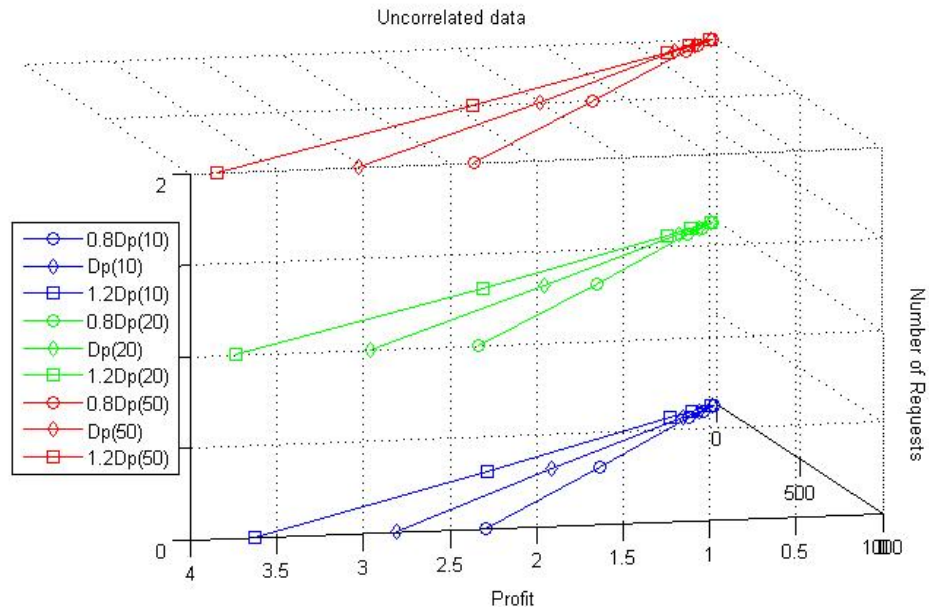
Σχήμα 5-8 Ποσοστό απόκλισης τελικού κέρδους σε σχέση με το αρχικό όταν η προσφερόμενη Καθυστερήση και το κέρδος έχουν ισχυρή σχέση συσχέτισης και πλήθος πελατών=[35-45]%



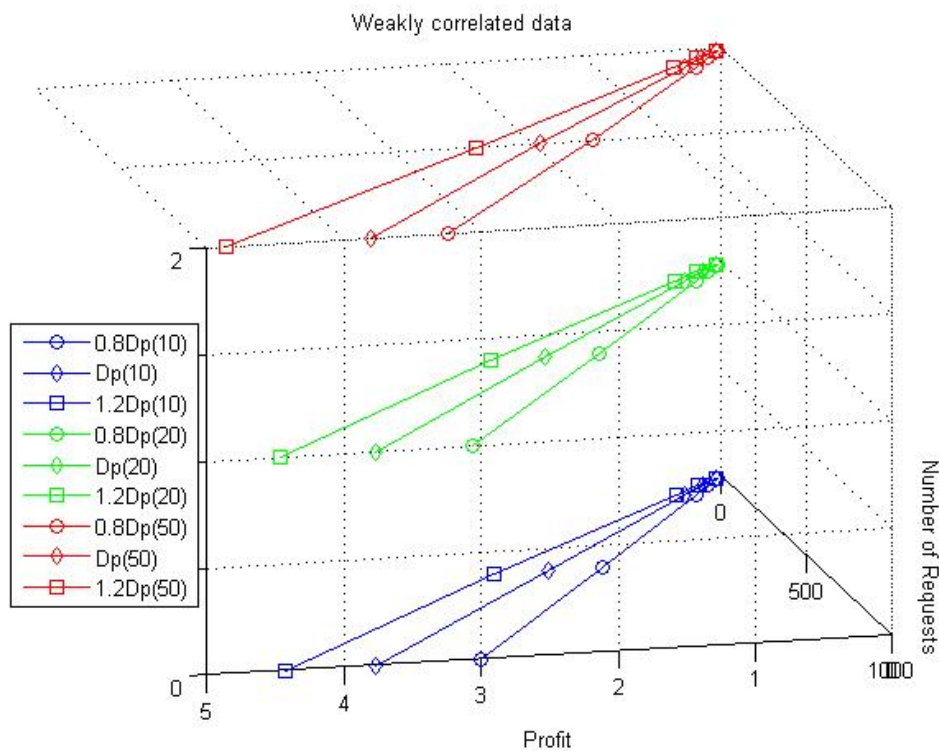
Σχήμα 5-9 Ποσοστό απόκλισης τελικού κέρδους σε σχέση με το αρχικό όταν η προσφερόμενη Καθυστερήση και το κέρδος έχουν ισχυρή σχέση συσχέτισης και πλήθος πελατών=[45-55]%



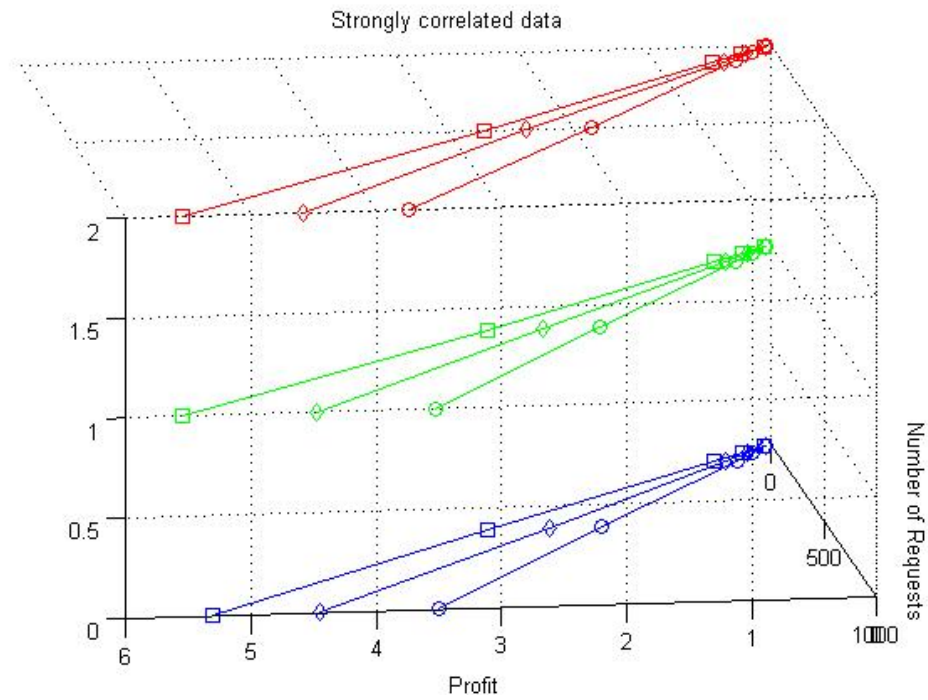
Σχήμα 5-10 Ποσοστό απόκλισης τελικού κέρδους σε σχέση με το αρχικό όταν η προσφερόμενη Καθυστερήση και το κέρδος έχουν ισχυρή σχέση συσχέτισης και πλήθος πελατών=[55-65]%



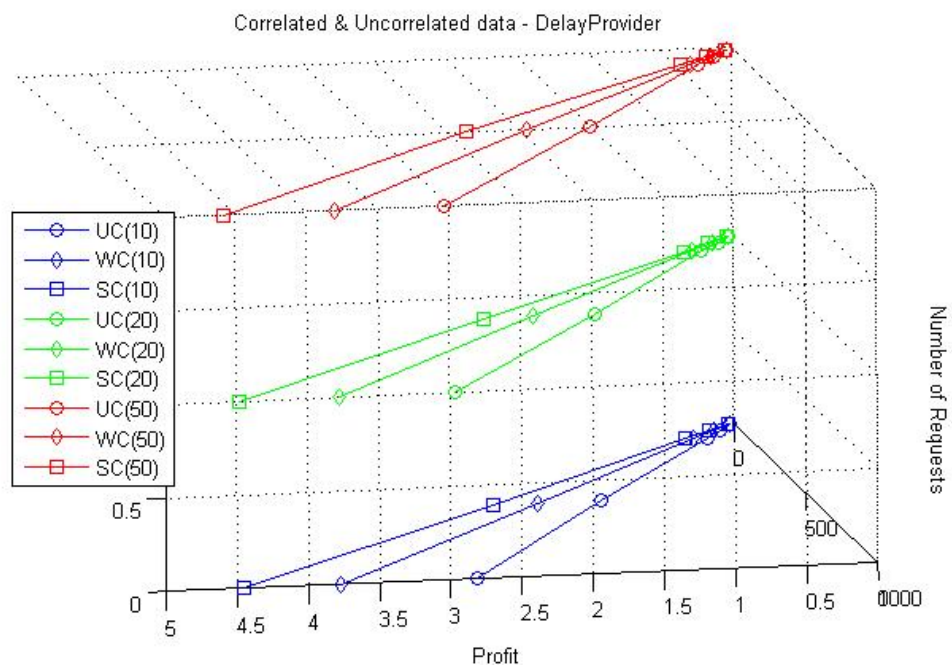
Σχήμα 5-11 Κέρδος παρόχου όταν η προσφερόμενη Καθυστέρηση και το κέρδος είναι ασυσχέτιστα μεταξύ τους



Σχήμα 5-12 Κέρδος παρόχου όταν η προσφερόμενη Καθυστέρηση και το κέρδος έχουν ασθενή σχέση συσχέτισης



Σχήμα 5-13 Κέρδος παρόχου όταν η προσφερόμενη Καθυστέρηση και το κέρδος έχουν ισχυρή σχέση συσχέτισης



Σχήμα 5-14 Κέρδος παρόχου για διάφορες σχέσεις συσχέτισης και πλήθος πελατών=[45-55]%

Το επόμενο μέγεθος που θα εξεταστεί είναι το «εύρος» της ελάχιστης *Καθυστέρησης*. Με τον όρο αυτό εννοείται η απόκλιση της τιμής της υπηρεσίας με την ελάχιστη *Καθυστέρηση* που επιλέγεται από τον αλγόριθμο σε σχέση είτε με την υπηρεσία που έχει τη μεγαλύτερη *Καθυστέρηση* είτε με άλλες υπηρεσίες που αφορούν την ίδια σχέση συσχέτισης.

Στους πίνακες που ακολουθούν (Πίνακας 5-6 - Πίνακας 5-8) περιλαμβάνονται οι ελάχιστες *Καθυστερήσεις* των υπηρεσιών που θα προσφερθούν, όπως αυτές υπολογίζονται από τον προτεινόμενο αλγόριθμο, ενώ οι τιμές μέσα στις παρενθέσεις παριστάνουν το πλήθος των πελατών που θα εξυπηρετηθούν κατά περίπτωση. Στον πίνακα 5-9 περιλαμβάνεται το εύρος των τιμών που περιλαμβάνονται στους πίνακες αυτούς, ενώ η τελευταία στήλη παριστάνει την απόκλιση της συνολικά μικρότερης *Καθυστέρησης* από τη συνολικά μεγαλύτερη για τη συγκεκριμένη μέτρηση ($\frac{D_{\max} - D_{\min}}{D_{\max}}$). Από τις πειραματικές μετρήσεις

που έγιναν, βρέθηκε ότι τις περισσότερες φορές η μέγιστη *Καθυστέρηση* ισούται με την επιτρεπόμενη μέγιστη, δηλαδή $D_{\max} = 1000$ (αφού το εύρος τιμών της *Καθυστέρησης* κυμαίνεται μεταξύ 1-1000). Η σκιαγράφηση των τιμών της στήλης αυτής περιλαμβάνονται στο Σχήμα 5-15. Όπως φαίνεται από τους πίνακες και το σχήμα που ακολουθούν, το προτεινόμενο μοντέλο για την *Καθυστέρηση* (§5.4) είναι αρκετά ακριβές, αφού παρατηρείται πολύ μικρή απόκλιση μεταξύ μικρότερης και μεγαλύτερης *Καθυστέρησης* όταν υπάρχει ακόμα και ασθενή σχέση συσχέτισης μεταξύ κέρδους και *Καθυστέρησης*. Όταν βέβαια τα μεγέθη αυτά είναι τελείως ασυσχέτιστα μεταξύ τους, η απόκλιση αυτή αυξάνει, όπως άλλωστε ήταν αναμενόμενο, χωρίς όμως ποτέ να γίνεται τόσο μεγάλη ώστε να απορρίπτει το προτεινόμενο μοντέλο.

N	0.8 * D _p (10 ⁵)			D _p (10 ⁵)			1.2 * D _p (10 ⁵)		
	k ₁	k ₂	k ₃	k ₁	k ₂	k ₃	k ₁	k ₂	k ₃
10	782(3)	782(3)	782(3)	773 (5)	834(5)	829(4)	781(5)	867(6)	854(5)
50	743(16)	779(16)	871(18)	749(20)	752(19)	839(21)	752(25)	794(25)	845(25)
100	779(33)	786(33)	793(33)	739(40)	778(40)	828(42)	755(50)	777(50)	798(48)
500	800(171)	785(162)	783(164)	804(211)	807(212)	803(207)	803(271)	842(263)	811(257)
1000	822(353)	802(332)	789(312)	812(440)	821(430)	836(420)	832(574)	829(569)	819(565)

Πίνακας 5-6 Ελάχιστη Καθυστέρηση και πλήθος πελατών που εξυπηρετούνται όταν η προσφερόμενη Καθυστέρηση και το κέρδος είναι ασυσχέτιστα μεταξύ τους

N	$0.8 * D_p (10^5)$			$D_p (10^5)$			$1.2 * D_p (10^5)$		
	k_1	k_2	k_3	k_1	k_2	k_3	k_1	k_2	k_3
10	946(4)	970(4)	988(4)	924 (5)	961(6)	983(6)	907(6)	951(6)	980(6)
50	948(21)	975(20)	989(21)	932(25)	966(25)	985(26)	909(30)	955(30)	981(31)
100	948(41)	974(41)	989(40)	929(51)	964(51)	986(50)	914(59)	954(60)	981(60)
500	950(199)	974(199)	989(202)	934(248)	965(249)	986(251)	912(300)	954(301)	980(307)
1000	950(399)	974(402)	989(407)	933(500)	965(499)	986(498)	915(584)	956(583)	980(601)

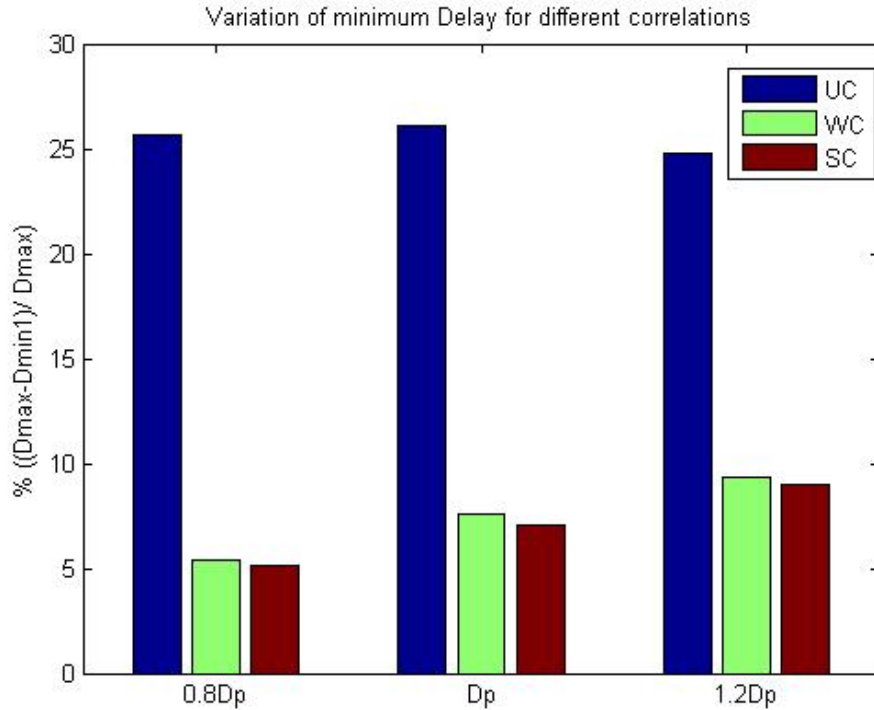
Πίνακας 5-7 Ελάχιστη Καθυστέρηση και πλήθος πελατών που εξυπηρετούνται όταν η προσφερόμενη Καθυστέρηση και το κέρδος έχουν ασθενή σχέση συσχέτισης

N	$0.8 * D_p (10^5)$			$D_p (10^5)$			$1.2 * D_p (10^5)$		
	k_1	k_2	k_3	k_1	k_2	k_3	k_1	k_2	k_3
10	949(4)	972(4)	987(4)	929 (5)	962(6)	985(6)	911(6)	952(6)	979(6)
50	950(21)	975(21)	989(21)	931(26)	963(25)	986(26)	912(30)	955(30)	981(31)
100	949(40)	974(40)	990(40)	933(50)	966(50)	986(51)	912(60)	955(60)	982(61)
500	950(200)	975(200)	990(202)	934(251)	966(251)	986(251)	911(304)	954(303)	983(301)
1000	950(399)	975(400)	990(401)	934(508)	966(506)	986(507)	910(605)	953(602)	983(604)

Πίνακας 5-8 Ελάχιστη Καθυστέρηση και πλήθος πελατών που εξυπηρετούνται όταν η προσφερόμενη Καθυστέρηση και το κέρδος έχουν ισχυρή σχέση συσχέτισης

N	$0.8 * D_p$			D_p			$1.2 * D_p$		
	$D \min_1$	$D \min_2$	%	$D \min_1$	$D \min_2$	%	$D \min_1$	$D \min_2$	%
UC	743	871	25.7	739	839	26.1	752	867	24.8
WC	946	989	5.4	924	986	7.6	907	981	9.3
SC	949	990	5.1	929	986	7.1	910	983	9.0

Πίνακας 5-9 Εύρος Ελάχιστης Καθυστέρησης για κάθε σχέση συσχέτισης



Σχήμα 5-15 Ποσοστό απόκλισης ελάχιστης και μέγιστης Καθυστέρησης για διάφορες σχέσεις συσχέτισης και πλήθους πελατών

Το τελευταίο μέγεθος που θα εξεταστεί είναι η χρονική απόκριση του προτεινόμενου αλγορίθμου, προκειμένου να διαπιστωθεί το πώς επιδρά η μεταβολή του πλήθους των πελατών και των προσφερόμενων υπηρεσιών για διάφορες σχέσεις συσχέτισης και να μελετηθεί αν η θεωρητικά υπολογισμένη πολυπλοκότητα επαληθεύεται και πειραματικά.

Όπως θα περίμενε κανείς η μεταβολή στη σχέση συσχέτισης μεταξύ κέρδους και Καθυστέρησης δεν επιδρά καθόλου στη χρονική απόκριση του αλγορίθμου, αφού δεν επηρεάζεται σε κάποιο βήμα ο αλγόριθμος από αυτήν. Επιπρόσθετα, όπως φαίνεται και στο σχήμα (Σχήμα 5-16) που ακολουθεί, η θεωρητικά υπολογισμένη πολυπλοκότητα (μεταξύ $O(n \log n)$ και $O(n^2)$) επαληθεύεται και πειραματικά, ενώ επισημαίνεται ότι η μεγαλύτερη επιβάρυνση στον αλγόριθμο όταν αυξάνεται το πλήθος των υπηρεσιών για σταθερό πλήθος πελατών προκαλείται από την εφαρμογή του πρώτου κριτηρίου επικράτησης.

N	$0.8 * D_p$			D_p			$1.2 * D_p$		
	k_1	k_2	k_3	k_1	k_2	k_3	k_1	k_2	k_3
10	0.016	0.018	0.019	0.018	0.019	0.020	0.018	0.018	0.019
50	0.042	0.065	0.098	0.045	0.067	0.154	0.048	0.068	0.155
100	0.085	0.114	0.435	0.088	0.126	0.442	0.103	0.164	0.444
500	0.987	1.977	23.632	1.089	2.125	26.111	1.232	2.763	32.162
1000	3.532	16.040	170.136	3.970	18.936	181.411	5.642	25.339	186.597

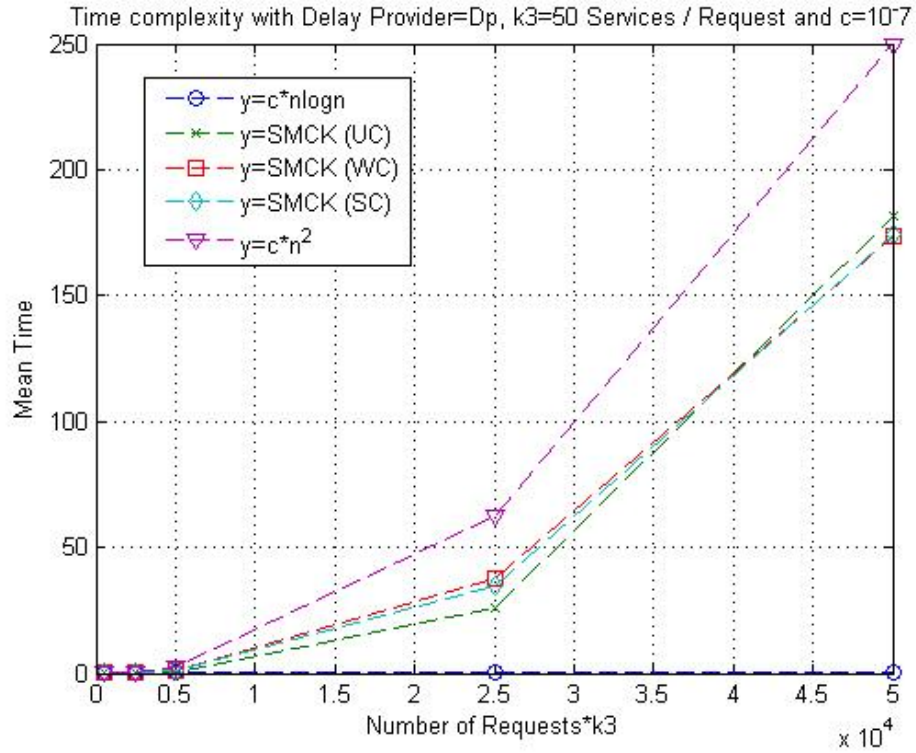
Πίνακας 5-10 Χρονική απόκριση SMCK όταν η προσφερόμενη Καθυστέρηση και το κέρδος είναι ασυσχέτιστα μεταξύ τους

N	$0.8 * D_p$			D_p			$1.2 * D_p$		
	k_1	k_2	k_3	k_1	k_2	k_3	k_1	k_2	k_3
10	0.009	0.010	0.018	0.008	0.011	0.018	0.007	0.011	0.018
50	0.026	0.052	0.248	0.027	0.046	0.251	0.026	0.049	0.242
100	0.069	0.156	0.969	0.067	0.153	0.937	0.065	0.146	0.972
500	1.294	4.470	37.193	1.370	3.837	37.306	1.434	4.038	37.674
1000	5.866	23.656	173.214	5.480	20.637	173.469	5.854	24.071	174.567

Πίνακας 5-11 Χρονική απόκριση SMCK όταν η προσφερόμενη Καθυστέρηση και το κέρδος έχουν ασθενή σχέση συσχέτισης

N	$0.8 * D_p$			D_p			$1.2 * D_p$		
	k_1	k_2	k_3	k_1	k_2	k_3	k_1	k_2	k_3
10	0.006	0.008	0.016	0.008	0.009	0.017	0.009	0.009	0.016
50	0.021	0.048	0.203	0.022	0.047	0.208	0.021	0.048	0.209
100	0.055	0.128	0.923	0.061	0.157	0.927	0.061	0.157	1.158
500	1.187	4.831	34.041	1.203	5.371	35.031	1.206	5.413	36.074
1000	5.171	21.640	172.152	5.247	22.440	174.106	6.061	23.847	175.964

Πίνακας 5-12 Χρονική απόκριση SMCK όταν η προσφερόμενη Καθυστέρηση και το κέρδος έχουν ισχυρή σχέση συσχέτισης



Σχήμα 5-16 Χρονική απόκριση προτεινόμενου αλγορίθμου με επιλεγμένη σταθερή Καθυστέρηση για τον πάροχο ίση με D_p και πλήθος υπηρεσιών / πελάτη = 50 για διάφορες σχέσεις συσχέτισης

Κεφάλαιο 6ο

6. Επίλογος

Στο κεφάλαιο αυτό θα γίνει μια ανασκόπηση των όσων αναφέρθηκαν στα προηγούμενα κεφάλαια, ενώ παράλληλα θα προταθούν πιθανές επεκτάσεις / παραλλαγές των προβλημάτων που εξετάστηκαν, οι οποίες δύναται να αποτελέσουν αντικείμενο μελλοντικής έρευνας.

Το θεωρητικό υπόβαθρο στο οποίο στηρίχθηκε η παρούσα διατριβή είναι το σημασιολογικό δίκτυο, οι διαδικτυακές υπηρεσίες και οι παράμετροι ποιότητας αυτών, ενώ η επίλυση των αλγοριθμικών προβλημάτων που εξετάστηκαν, εφάρμοσαν τεχνικές που ακολουθούνται κατά την επίλυση των προβλημάτων της οικογένειας του σακιδίου (knapsack family).

Πιο αναλυτικά, στο πρώτο κεφάλαιο έγινε μια σύντομη αναφορά στις τεχνολογίες του διαδικτύου και των διαδικτυακών υπηρεσιών. Μέσω κατάλληλων παραδειγμάτων έγιναν σαφείς οι αδυναμίες που παρουσιάζουν οι τεχνολογίες αυτές στην παρούσα μορφή τους, καθώς και το γεγονός ότι είναι επιτακτική ανάγκη η αντιμετώπιση των αδυναμιών αυτών με την υιοθέτηση νέων τεχνολογιών. Το σημασιολογικό δίκτυο αποτελεί ίσως τη μοναδική λύση για την επίτευξη του στόχου αυτού, αφού χρησιμοποιώντας κατάλληλες τεχνικές διαστρωμάτωσης και οντολογίες για την περιγραφή κάθε είδους πληροφορίας ξεπερνά με επιτυχία τις αδυναμίες αυτές. Το κεφάλαιο ολοκληρώνεται με την παρουσίαση ενός προβλήματος, η λύση του οποίου αποτελεί τη συμβολή της παρούσας διατριβής: *Έστω ένας πάροχος που προσφέρει διαδικτυακές υπηρεσίες, οι οποίες διαθέτουν παραμέτρους ποιότητας στην περιγραφή τους. Οι υπηρεσίες αυτές στηρίζουν την περιγραφή τους αποκλειστικά σε οντολογίες. Λαμβάνοντας ακόμη υπόψη το γεγονός ότι ο πάροχος μπορεί να εξυπηρετήσει συγκεκριμένο αριθμό πελατών που είναι απόρροια του γεγονότος ότι έχει περιορισμένους*

πόρους, ζητείται να βρεθούν μέθοδοι επιλογής πελατών και υπηρεσιών που θα προσφέρει στους πελάτες του που θα του αποφέρουν το μεγαλύτερο δυνατό κέρδος.

Ο ρόλος του δεύτερου κεφαλαίου εστιάζεται στην παρουσίαση των κατάλληλων θεωρητικών πληροφοριών που σχετίζονται με τις διαδικτυακές υπηρεσίες και το σημασιολογικό δίκτυο, δίνοντας ιδιαίτερη έμφαση στις οντολογίες που αποτελούν ένα από τα σημαντικότερα θέματα που μας απασχόλησε στη διατριβή αυτή. Αρχικά στο κεφάλαιο αυτό αναλύθηκαν οι τεχνολογίες που συμμετέχουν κατά την υλοποίηση και εκτέλεση των διαδικτυακών υπηρεσιών. Μεταξύ αυτών ξεχωρίζουν η αρχιτεκτονική SOA, το πρωτόκολλο SOAP, η γλώσσα WSDL και οι σελίδες UDDI. Ακόμη, παρουσιάστηκαν τα στρώματα στα οποία στηρίζεται το σημασιολογικό δίκτυο αναλύοντας ξεχωριστά το κάθε ένα από αυτά. Τα στρώματα αυτά είναι: Το στρώμα URI – Unicode, το στρώμα XML – XSD, το στρώμα RDF – RDFS, το στρώμα οντολογιών και τα στρώματα Λογικής, Απόδειξης και Εμπιστοσύνης. Όπως αναφέρθηκε στο κεφάλαιο αυτό, αρκετή έρευνα είναι απαραίτητο να γίνει ιδιαίτερα στα τελευταία 3 στρώματα, προτού γίνει πραγματικότητα το όραμα του σημασιολογικού δικτύου. Το κεφάλαιο ολοκληρώνεται με παρουσίαση μιας γλώσσας για σημασιολογικές διαδικτυακές υπηρεσίες: της OWL – S.

Ο ρόλος του τρίτου κεφαλαίου είναι διττός: Αφενός παρουσιάζει τα προβλήματα που προκαλούνται από την παροχή υπηρεσιών που δεν προσφέρουν καθορισμένο επίπεδο ποιότητας και αφετέρου προσφέρει κατάλληλα μοντέλα οντολογιών για παραμέτρους ποιότητας διαδικτυακών υπηρεσιών, η χρήση των οποίων μπορεί να αντιμετωπίσει τα προβλήματα αυτά με επιτυχία. Τα μοντέλα αυτά συμβάλουν στην επίλυση του πρώτου μέρους του προβλήματος, όπως αυτό παρουσιάστηκε στο πρώτο κεφάλαιο, αποτελώντας το ένα μέρος της συμβολής της παρούσας διατριβής. Πιο συγκεκριμένα, αναλύονται τα βήματα που ακολουθήθηκαν για το σχεδιασμό των οντολογιών αυτών, προτείνονται δύο οντολογίες παραμέτρων ποιότητας (οντολογία Γλώσσα και οντολογία Λεξιλόγιο) αναλύοντας εκτενώς όλους τους όρους που συμμετέχουν σε αυτές, ενώ παράλληλα γίνεται μια αποτίμηση των προτεινόμενων σχεδιασμών. Όπως αναλύεται στο κεφάλαιο αυτό, οι προτεινόμενες οντολογίες ικανοποιούν επαρκώς τα περισσότερα κριτήρια βάσει των οποίων εξετάστηκαν, ενώ ο ορθός σχεδιασμός τους επιβεβαιώνεται και από τη δημιουργία μεγάλου πλήθους αναφορών από άλλες ερευνητικές εργασίες.

Στην παρουσίαση θεωρητικών πληροφοριών εστιάζεται και το τέταρτο κεφάλαιο. Το πεδίο που επικεντρώνεται είναι οι αλγόριθμοι και πιο συγκεκριμένα η πολυπλοκότητα των αλγορίθμων, καθώς και οι τεχνικές που ακολουθούνται κατά την επίλυση του προβλήματος του σακιδίου. Στο πρώτο μέρος του κεφαλαίου αυτού παρουσιάζονται οι συμβολισμοί που χρησιμοποιούνται για να χαρακτηρίσουν την πολυπλοκότητα ενός αλγορίθμου, επισημαίνοντας τη διαφορά - όσον αφορά το χρόνο απόκρισης - μεταξύ πολωνυμικής και εκθετικής πολυπλοκότητας. Παράλληλα κατατάσσονται τα προβλήματα που καλούνται να

λύσουν οι διάφοροι αλγόριθμοι σε ορισμένες κατηγορίες. Οι σημαντικότερες από αυτές είναι οι κλάσεις P και NP , οι οποίες αποτελούν αντικείμενο μελέτης για τους μαθηματικούς εδώ και πολλές δεκαετίες. Υπενθυμίζεται ότι δεν έχει αποδειχθεί ακόμη αν η κλάση P είναι υποκλάση της NP ή όχι, αν και όπως επισημαίνεται στο κεφάλαιο αυτό υπάρχουν σαφείς ενδείξεις που ενισχύουν την άποψη ότι είναι. Στο κεφάλαιο αυτό παρουσιάζεται ακόμα η οικογένεια του σακιδίου, ενώ αναλύονται οι τεχνικές επίλυσης που ακολουθούνται για τα δύο σημαντικότερα μέλη της οικογένειας αυτής: το $0-1 KP$ και το $MCKP$.

Στο πέμπτο και τελευταίο κεφάλαιο της διατριβής προτείνονται κατάλληλοι αλγόριθμοι που επιλύουν το τελευταίο μέρος του προβλήματος, όπως αυτό παρουσιάστηκε στο πρώτο κεφάλαιο. Έχοντας ως βάση το θεωρητικό υπόβαθρο του προηγούμενου κεφαλαίου και χρησιμοποιώντας την οντολογία Λεξιλόγιο του κεφαλαίου 3, γίνεται σε πρώτο στάδιο μια αξιολόγηση των παραμέτρων ποιότητας που μπορούν να χρησιμοποιηθούν στους προτεινόμενους αλγόριθμους. Όπως έγινε φανερό στο κεφάλαιο αυτό, επειδή το πρόβλημα προσεγγίζεται από τη μεριά του παρόχου και το ενδιαφέρον εστιάζεται στις παραμέτρους εκείνες που η τιμή τους επηρεάζεται με τη μεταβολή του πλήθους των χρηστών / αιτήσεων, αρκετές από τις παραμέτρους που περιλαμβάνονται στην οντολογία αυτή δε μπορούν να συμμετέχουν ως περιορισμός στον προτεινόμενο αλγόριθμο, αφού η τιμή τους παραμένει σταθερή ανεξάρτητα από το πλήθος των χρηστών / αιτήσεων. Λαμβάνοντας αυτό υπόψη καθώς και το γεγονός ότι δεν είναι εφικτό να μοντελοποιηθούν όλες οι μη γραμμικές παράμετροι, προτείνεται κατάλληλος αλγόριθμος βελτιστοποίησης του κέρδους του παρόχου για τις εξής 4 παραμέτρους: το *Κόστος*, τη *Χωρητικότητα*, το *Εύρος Ζώνης* και την *Καθυστέρηση*. Εξάλλου επειδή το πρόβλημα αποτελεί παραλλαγή του προβλήματος $MCKP$ το οποίο ανήκει στην κατηγορία των προβλημάτων NP – Δύσκολα (NP - Hard), είναι απαραίτητο να ακολουθηθούν προσεγγιστικές μέθοδοι, όπως αυτές που παρουσιάστηκαν στο τέταρτο κεφάλαιο. Το κεφάλαιο ολοκληρώνεται με τη διεξαγωγή κατάλληλων πειραμάτων και μετρήσεων, προκειμένου να γίνει η αποτίμηση της αποτελεσματικότητας του προτεινόμενου αλγορίθμου $SMCK$. Δύο είναι οι παράμετροι των οποίων μελετήθηκε η συμπεριφορά: Η μεταβολή της τιμής της σταθερής *Καθυστέρησης* του παρόχου D_p και (ii) η μεταβολή του πλήθους των υπηρεσιών που εξυπηρετούν τον κάθε πελάτη. Από τα πειράματα που διεξήχθησαν, παρατηρήθηκε ότι η συμπεριφορά του προτεινόμενου αλγορίθμου είναι πολύ ικανοποιητική, κάτι που είναι και συνέπεια της πολυωνυμικής πολυπλοκότητάς του ($O(n \log n) \leq O(SMCK) \leq O(n^2)$).

Ορισμένες από τις πιθανές προεκτάσεις / παραλλαγές του προβλήματος που αντιμετωπίστηκε και οι οποίες μπορούν να αποτελέσουν αντικείμενο περαιτέρω έρευνας και μελέτης είναι:

- Να προταθεί μοντέλο για τη *Μεταβολή Καθυστέρησης* (jitter) και να τροποποιηθεί κατάλληλα ο αλγόριθμος, ώστε να δίνει λύση και σε αυτή την περίπτωση.
- Να εξεταστούν παράλληλα οι περιορισμοί των διαθέσιμων πόρων του παρόχου.
- Να χρησιμοποιηθούν πράκτορες (agents) και να προταθεί κατάλληλη αρχιτεκτονική ενσωμάτωσής τους καθώς και τεχνικές διαπραγμάτευσης των παρεχόμενων και αιτούμενων επιπέδων ποιότητας των προσφερόμενων υπηρεσιών.
- Να προσεγγιστεί το πρόβλημα από τη μεριά του πελάτη, λαμβάνοντας υπόψη την τοπολογία του δικτύου και τις τυχόν παρεκκλίσεις στην ποιότητα της προσφερόμενης υπηρεσίας - όταν αυτή οφείλεται σε δικτυακούς παράγοντες - σε σχέση με αυτή που απολαμβάνει τελικά ο πελάτης.

Παράρτημα Α

(πηγαίος κώδικας οντολογιών Γλώσσας και Λεξιλόγιο με χρήστη του Protégé)

Οντολογία Γλώσσα

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns="http://www.owl-ontologies.com/unnamed.owl#"
  xml:base="http://www.owl-ontologies.com/unnamed.owl">
  <owl:Ontology rdf:about=""/>
  <owl:Class rdf:ID="QoSParameter"/>
  <owl:Class rdf:ID="ConversionFormula"/>
  <owl:Class rdf:ID="Unit"/>
  <owl:Class rdf:ID="ImpactFactor"/>
  <owl:Class rdf:ID="Aggregated"/>
  <owl:Class rdf:ID="Metric"/>
  <owl:Class rdf:ID="QoSImpact"/>
  <owl:Class rdf:ID="Statistics">
    <rdfs:subClassOf rdf:resource="#Metric"/>
  </owl:Class>
  <owl:Class rdf:ID="Relationship"/>
  <owl:Class rdf:ID="Type"/>
  <owl:ObjectProperty rdf:ID="hasType">
    <rdfs:domain rdf:resource="#QoSParameter"/>
    <rdfs:range rdf:resource="#Type"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="hasQoSImpact">
    <rdfs:range rdf:resource="#QoSImpact"/>
    <rdfs:domain rdf:resource="#QoSParameter"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="hasConversionFormula">
    <rdfs:domain rdf:resource="#Unit"/>
    <rdfs:range rdf:resource="#ConversionFormula"/>
  </owl:ObjectProperty>
```

```

<owl:ObjectProperty rdf:ID="influentialParameter">
  <rdfs:domain rdf:resource="#Relationship"/>
  <rdfs:range rdf:resource="#QoSParameter"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasUnit">
  <rdfs:range rdf:resource="#Unit"/>
  <rdfs:domain rdf:resource="#Metric"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="isAggregated">
  <rdfs:range rdf:resource="#Aggregated"/>
  <rdfs:domain rdf:resource="#QoSParameter"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasMetric">
  <rdfs:range rdf:resource="#Metric"/>
  <rdfs:domain rdf:resource="#QoSParameter"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasRelationship">
  <rdfs:range rdf:resource="#Relationship"/>
  <rdfs:domain rdf:resource="#QoSParameter"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="convertsTo">
  <rdfs:domain rdf:resource="#ConversionFormula"/>
  <rdfs:range rdf:resource="#Unit"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasImpactFactor">
  <rdfs:range rdf:resource="#ImpactFactor"/>
  <rdfs:domain rdf:resource="#Relationship"/>
</owl:ObjectProperty>
<owl:DatatypeProperty rdf:ID="Nature">
  <rdfs:domain rdf:resource="#QoSParameter"/>
  <rdfs:range>
    <owl:DataRange>
      <owl:oneOf rdf:parseType="Resource">
        <rdf:rest rdf:parseType="Resource">
          <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >Static</rdf:first>
          <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
        </rdf:rest>
        <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >Dynamic</rdf:first>
      </owl:oneOf>
    </owl:DataRange>
  </rdfs:range>
</owl:DatatypeProperty>

```



```

    </owl:oneOf>
  </owl:DataRange>
</rdfs:range>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="Node">
  <rdfs:range>
    <owl:DataRange>
      <owl:oneOf rdf:parseType="Resource">
        <rdf:rest rdf:parseType="Resource">
          <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >Client</rdf:first>
          <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
        </rdf:rest>
        <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
          >Server</rdf:first>
      </owl:oneOf>
    </owl:DataRange>
  </rdfs:range>
  <rdfs:domain rdf:resource="#QoSParameter"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="MetricType">
  <rdfs:domain rdf:resource="#Metric"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="Value">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="#Metric"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="validityLevel">
  <rdfs:range>
    <owl:DataRange>
      <owl:oneOf rdf:parseType="Resource">
        <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
          >Strong</rdf:first>
        <rdf:rest rdf:parseType="Resource">
          <rdf:rest rdf:parseType="Resource">
            <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
              >Medium</rdf:first>
            <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
          </rdf:rest>
        </rdf:rest>
      </owl:oneOf>
    </owl:DataRange>
  </rdfs:range>

```

```

    <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Weak</rdf:first>
  </rdf:rest>
</owl:oneOf>
</owl:DataRange>
</rdfs:range>
<rdfs:domain rdf:resource="#ImpactFactor"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="IFType">
  <rdfs:range>
    <owl:DataRange>
      <owl:oneOf rdf:parseType="Resource">
        <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >Proportional</rdf:first>
        <rdf:rest rdf:parseType="Resource">
          <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
          <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
          >InverselyProportional</rdf:first>
        </rdf:rest>
      </owl:oneOf>
    </owl:DataRange>
  </rdfs:range>
  <rdfs:domain rdf:resource="#ImpactFactor"/>
</owl:DatatypeProperty>
</rdf:RDF>

```

Οντολογία Λεξιλόγιο

```

<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  <owl:Class rdf:ID="QoSParameter"/>
  <owl:Class rdf:ID="SupportedStandard">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Configuration"/>

```

```

    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Performance">
    <rdfs:subClassOf rdf:resource="#QoSParameter"/>
  </owl:Class>
  <owl:Class rdf:ID="Confidentiality">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Security"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Authorization">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#Security"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Cost">
    <rdfs:subClassOf rdf:resource="#QoSParameter"/>
  </owl:Class>
  <owl:Class rdf:ID="ErrorRate">
    <rdfs:subClassOf rdf:resource="#Performance"/>
  </owl:Class>
  <owl:Class rdf:ID="Stability">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#Configuration"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Capacity">
    <rdfs:subClassOf rdf:resource="#QoSParameter"/>
  </owl:Class>
  <owl:Class rdf:ID="ResponseTime">
    <rdfs:subClassOf rdf:resource="#Performance"/>
  </owl:Class>
  <owl:Class rdf:ID="Throughput">
    <rdfs:subClassOf rdf:resource="#Performance"/>
  </owl:Class>
  <owl:Class rdf:ID="NonRepudiation">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#Security"/>
    </rdfs:subClassOf>
  </owl:Class>

```

```

<owl:Class rdf:about="#Security">
  <rdfs:subClassOf rdf:resource="#QoSParameter"/>
</owl:Class>
<owl:Class rdf:ID="Jitter">
  <rdfs:subClassOf rdf:resource="#Performance"/>
</owl:Class>
<owl:Class rdf:ID="Delay">
  <rdfs:subClassOf rdf:resource="#Performance"/>
</owl:Class>
<owl:Class rdf:ID="MTTR">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Reliability"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="MTBF">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Reliability"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Scalability">
  <rdfs:subClassOf rdf:resource="#QoSParameter"/>
</owl:Class>
<owl:Class rdf:ID="Availability">
  <rdfs:subClassOf rdf:resource="#QoSParameter"/>
</owl:Class>
<owl:Class rdf:ID="Integrity">
  <rdfs:subClassOf rdf:resource="#QoSParameter"/>
</owl:Class>
<owl:Class rdf:ID="Authentication">
  <rdfs:subClassOf rdf:resource="#Security"/>
</owl:Class>
<owl:Class rdf:ID="Auditability">
  <rdfs:subClassOf rdf:resource="#Security"/>
</owl:Class>
<owl:Class rdf:ID="Regulatory">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Configuration"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="DataEncryption">

```

```
<rdfs:subClassOf rdf:resource="#Security"/>
</owl:Class>
<owl:Class rdf:about="#Reliability">
  <rdfs:subClassOf rdf:resource="#QoSParameter"/>
</owl:Class>
<owl:Class rdf:about="#Configuration">
  <rdfs:subClassOf rdf:resource="#QoSParameter"/>
</owl:Class>
<owl:Class rdf:ID="Accessibility">
  <rdfs:subClassOf rdf:resource="#QoSParameter"/>
</owl:Class>
</rdf:RDF>
```

Παράρτημα Β

(αρχεία πηγαίου κώδικα Matlab - με τη σειρά που αυτά καλούνται - για τον αλγόριθμο βελτιστοποίησης *SMCK*. Τα αρχεία δημιουργίας των διαγραμμάτων δεν περιλαμβάνονται για λόγους εργονομικούς.

create_uncorrelated_services.m

```
Services=[];  
Total_Number_Services=10000;  
R1=1000;  
for i=1:Total_Number_Services  
    %h prwth sthlh einai to Delay kai h deuterh to Profit  
    Services(i,1)=ceil(R1*(rand(1,1)));  
    Services(i,2)=ceil(R1*(rand(1,1)));  
end
```

create_weakly_correlated_services.m

```
Services=[];  
Total_Number_Services=10000;  
R1=1000;  
for i=1:Total_Number_Services  
    %h prwth sthlh einai to Delay kai h deuterh to Profit  
    Services(i,1)=ceil(R1*(rand(1,1)));  
    Services(i,2)=ceil(0.5*Services(i,1))+ceil(0.5*R1*(rand(1,1)));  
end
```

create_strongly_correlated_services.m

```
Services=[];  
Total_Number_Services=10000;  
R1=1000;  
for i=1:Total_Number_Services  
    %h prwth sthlh einai to Delay kai h deuterh to Profit  
    Services(i,1)=ceil(R1*(rand(1,1)));  
    Services(i,2)=ceil(0.8*Services(i,1))+ceil(0.2*R1*(rand(1,1)));  
end
```

create_customers.m

```
NumberCust=100;  
NumberServ=10;  
index=1;
```

```

for i=1:NumberCust
    for j=1:NumberServ
        item=ceil(Total_Number_Services*(rand(1,1)));
        Customers(index,1)=i;
        Customers(index,2)=j;
        Customers(index,3)=Services(item,1);
        Customers(index,4)=Services(item,2);
        index=index+1;
    end
end

```

execution.m

```

Customers=[];
Initial_Solution=[];
Solution=[];
Slopes=[];
New_Slopes=[];
Results=[];
iterations=50;
profile on
for n=1:iterations
    first_dominance;
    second_dominance;
    Step2;
    Step3;
    Step4;
    Step5;
    Results(n,1)=LB1;
    Results(n,2)=LB2;
    Results(n,3)=LB3;
    Results(n,4)=Profit;
    Results(n,5)=min(Solution(:,3));
    Results(n,6)=max(Solution(:,3));
    Results(n,7)=length(Solution);
    Customers=[];
    Initial_Solution=[];
    Solution=[];
    Slopes=[];
    New_Slopes=[];
end

```

```

end
profile viewer
sum(Results)/iterations

```

first_dominance.m

```

%calculate delay provider
DelayProvider=ceil(0.8*NumberCust/2-NumberCust/20+ceil(NumberCust*rand(1,1))/10);

Customers=sortrows(Customers,[1,-3]);
item=1;
index=NumberCust*NumberServ;
while item<index
    if ((Customers(item,1)== Customers(item+1,1)) &&
(Customers(item,4)>=Customers(item+1,4)))
        Customers(item+1,:)=[];
        index=index-1;
    elseif ((Customers(item,1)== Customers(item+1,1)) &&
(Customers(item,3)==Customers(item+1,3)) &&
(Customers(item,4)<=Customers(item+1,4)))
        Customers(item,:)=[];
        index=index-1;
    else
        item=item+1;
    end
end

```

second_dominance.m

```

item=1;
index=length(Customers);
while item<=index-2
    i=item+1;
    while i<=index-1
        if (Customers(item,1)== Customers(i,1))
            j=i+1;
            while j<=index-1
                if (Customers(item,1)== Customers(j,1))
                    if (((Customers(j,4)-Customers(i,4))/(Customers(i,3)-
Customers(j,3)))>=((Customers(i,4)-Customers(item,4))/(Customers(item,3)-
Customers(i,3))))
                        Customers(i,:)=[];

```



```

        index=index-1;
    else
        j=j+1;
    end
    else
        break;
    end
end
i=i+1;
else
    break;
end
end
item=item+1;
end

```

Step2.m

```

Initial_Solution(1,:)=Customers(1,:);
item=2;
for i=1:length(Customers)
    if (Customers(i,1)== item)
        Initial_Solution(item,:)=Customers(i,:);
        item=item+1;
    end
end
Initial_Solution=sortrows(Initial_Solution,-3);
Solution(1:DelayProvider,:)=Initial_Solution(1:DelayProvider,:);
minDelay=min(Solution(:,3));
DelayProvider=minDelay/length(Solution);
LBI=sum(Solution(:,4));

```

Step3.m

```

item=1;
for i=1:(length(Customers)-1)
    if (Customers(i+1,1)==Customers(i,1))
        Slopes(item,1)=Customers(i,1);
        Slopes(item,2)=Customers(i+1,2);
        Slopes(item,3)=Customers(i+1,3);
        Slopes(item,4)=Customers(i+1,4);
    end
end

```

```

        Slopes(item,5)=(Customers(i+1,4)-Customers(i,4))/((Customers(i,3)-
Customers(i+1,3)));
        item=item+1;
    end;
end
if (length(Slopes)>0)
    Slopes=sortrows(Slopes,-5);
    item=length(Solution);
    for i=1:length(Slopes)
        %check if this customer is already served from another service
        served=0;
        for k=1:length(Solution)
            if (Slopes(i,1)==Solution(k,1))
                served=k;
                break;
            end
        end
        if (served>0)
            if (Slopes(i,3)>=minDelay && Slopes(i,4)>Solution(served,4))
                Solution(served,2)=Slopes(i,2);
                Solution(served,3)=Slopes(i,3);
                Solution(served,4)=Slopes(i,4);
            elseif (item/Slopes(i,3)<=1/DelayProvider && Slopes(i,4)>Solution(served,4))
                minDelay=Slopes(i,3);
                Solution(served,2)=Slopes(i,2);
                Solution(served,3)=Slopes(i,3);
                Solution(served,4)=Slopes(i,4);
            else
                break;
            end
        else %new customer
            if ((Slopes(i,3)>=minDelay) && ((item+1)/minDelay)<=(1/DelayProvider))
                item=item+1;
                Solution(item,1)=Slopes(i,1);
                Solution(item,2)=Slopes(i,2);
                Solution(item,3)=Slopes(i,3);
                Solution(item,4)=Slopes(i,4);
            elseif ((Slopes(i,3)<minDelay) && ((item+1)/Slopes(i,3))<=(1/DelayProvider))
                minDelay=Slopes(i,3);
                item=item+1;
            end
        end
    end
end

```

```

        Solution(item,1)=Slopes(i,1);
        Solution(item,2)=Slopes(i,2);
        Solution(item,3)=Slopes(i,3);
        Solution(item,4)=Slopes(i,4);
    else
        break;
    end
end
end
New_Slopes=Slopes(i:length(Slopes),:);
end
LB2=sum(Solution(:,4));

```

Step4.m

```

for i=1:length(Solution)
    for k=1:length(Customers)
        if (Solution(i,1)==Customers(k,1))
            if (Customers(k,3)>=minDelay && Customers(k,4)>=Solution(i,4))
                Solution(i,:)=Customers(k,:);
            end
        else
            continue;
        end
    end
end
LB3=sum(Solution(:,4));

```

Step5.m

```

if (length(New_Slopes)>0)
    New_Slopes=sortrows(New_Slopes,-5);
    item=length(Solution);
    for i=1:length(New_Slopes)
        %this is added for the "extra" functionality with regards to MCKP
        Solution=sortrows(Solution,-4);
        %check if this customer is already served from another service
        served=0;
        for k=1:length(Solution)
            if (New_Slopes(i,1)==Solution(k,1))
                served=k;
                break;
            end
        end
    end
end

```

```

        end
    end
    if (served>0)
        %this is added for the "extra" functionality with regards to MCKP
        if (item/New_Slopes(i,3)>1/DelayProvider)
            break_item=ceil(New_Slopes(i,3)/DelayProvider)-1;
            sumSolution=sum(Solution(:,4));
            if (break_item>=served)
                newSumSolution=sum(Solution(1:break_item,4))+New_Slopes(i,4)-
Solution(served,4);
                if (newSumSolution>=sumSolution)
                    minDelay=New_Slopes(i,3);
                    Solution((break_item+1):length(Solution),:)=[];
                    Solution(served,2)=New_Slopes(i,2);
                    Solution(served,3)=New_Slopes(i,3);
                    Solution(served,4)=New_Slopes(i,4);
                    item=break_item;
                end
            else
                newSumSolution=sum(Solution(1:(break_item-1),4))+New_Slopes(i,4);
                if (newSumSolution>=sumSolution)
                    minDelay=New_Slopes(i,3);
                    Solution(break_item:length(Solution),:)=[];
                    item=break_item;
                    Solution(item,2)=New_Slopes(i,2);
                    Solution(item,3)=New_Slopes(i,3);
                    Solution(item,4)=New_Slopes(i,4);
                end
            end
        end
    end
    else %new customer
        %this is added for the "extra" functionality with regards to MCKP
        if ((New_Slopes(i,3)<minDelay) && (item+1)/New_Slopes(i,3)>1/DelayProvider)
            break_item=ceil(New_Slopes(i,3)/DelayProvider)-1;
            sumSolution=sum(Solution(:,4));
            newSumSolution=sum(Solution(1:(break_item-1),4))+New_Slopes(i,4);
            if (newSumSolution>=sumSolution)
                minDelay=New_Slopes(i,3);
                Solution((break_item):length(Solution),:)=[];
                item=break_item;
            end
        end
    end
end

```

```
        Solution(item,2)=New_Slopes(i,2);
        Solution(item,3)=New_Slopes(i,3);
        Solution(item,4)=New_Slopes(i,4);
    end
end
end
end
Profit=sum(Solution(:,4));
```

Βιβλιογραφία

- [1] G. G. Antoniou, and F. van Harmelen. 2004. *A Semantic Web Primer*. MIT Press
- [2] 2007, “Ontology”, <http://en.wikipedia.org/wiki/Ontology>
- [3] 2000, “Extensible Markup Language (XML)”, <http://www.w3.org/XML/>
- [4] 2000, “RDFS Specification”, <http://www.w3.org/TR/2000/CR-rdf-schema-20000327/>
- [5] 2004, “OWL: Web Ontology Language”, <http://www.w3.org/TR/owl-features/>
- [6] T. Berners-Lee, J. Hendler, and O. Lassila. 2001. The Semantic Web. *Scientific American Magazine*: 34:43
- [7] R. Khalaf, S. Mukhi, S. Tai, and S. Weerawarana. 2003. The next steps in Web Services. *Communications of the ACM* 46(10): 29-34
- [8] W. Iverson. 2004. *Real World Web Services*. O'Reilly
- [9] D. Tidwell. 2000. “Web Services: The Web’s next revolution”, <http://www.ibm.com>
- [10] T. Erl. 2004. *Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services*. Prentice Hall PTR
- [11] 2000, “Simple Object Access Protocol (SOAP) 1.1”, <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>
- [12] 2007, “Web Services Description Language (WSDL) Version 2.0 Part 1:Core Language”, <http://www.w3.org/TR/wsdl20/>
- [13] A. Ankolekar, M. Burstein, J. R. Hobbs, O. Lassila, D. L. Martin, D. McDermott, A. McIlraith, S. Narayanan, M. Paolucci, T. R. Payne, and K. Sycara. 2002. DAML-S: Web Service Description for the Semantic Web. Paper presented at the 1st International Semantic Web Conference (ISWC), June 9-12, in Sardinia, Italy
- [14] 2004, “OWL-S: Semantic Markup for Web Services”, <http://www.w3.org/Submission/OWL-S/>
- [15] M. Govern, and S. Weagraff. 2003. *Web Services: Provider Threat or Opportunity?*. White paper, Convergys Corporation
- [16] D. T. Tsesmetzis, I. G. Roussaki, I. V. Papaioannou, and M. E. Anagnostou. 2006. QoS awareness support in Web – Service semantics. Paper presented at the International Conference on Internet and Web Applications and Services / Advances International Conference (AICT – ICIW 2006), February 19-25, in Guadeloupe, French Caribbean
- [17] I. V. Papaioannou, D. T. Tsesmetzis, I. G. Roussaki, and M. E. Anagnostou. 2006. A QoS Ontology Language for Web - Services. Paper presented at the 20th International Conference on Advanced Information Networking and Applications (AINA2006), April 18-20, in Vienna, Austria
- [18] J. Kalaoja, J. Kantorovitch, I. Roussaki, D. T. Tsesmetzis, and I. V. Papaioannou. 2007. Ontology Modeling for Ambient Intelligent Home Environments. Paper presented at the 2007 ODBASE (Ontologies, Databases, and Applications of Semantics) International Conference, November 25-30, in Vilamoura, Portugal
- [19] N. Georgantas, J. Kantorovitch, I. Roussaki, V. Issarny, J. Kalaoja, I. Papaioannou, and D. Tsesmetzis. 2008. Amigo: Interoperable semantic services for the smart home environment in “At your service: An overview of results of projects in the field of service engineering of the IST programme”. *MIT Press Series on Information Systems* (in press)

- [20] X. Wang, T. Vitvar, M. Kerrigan, and I. Toma. 2006. A QoS-aware Selection Model for Semantic Web Services. Paper presented at the 4th International Conference on Service Oriented Computing (ICSOC06), December 4-7, in Chicago, USA
- [21] R. Afandi, J. Zhang, and C. A. Gunter. 2006. AMPol-Q: Adaptive Middleware to Support QoS. *Lecture Notes in Computer Science* 4294: 165-178
- [22] R. Jurca, W. Binder, and B. Faltings. 2007. Reliable QoS Monitoring Based on Client Feedback. Paper presented at the 16th International World Wide Web Conference, May 8-12, in Banff, Canada
- [23] G. Dobson, and A. Sanchez-Macian. 2006. Towards Unified QoS / SLA Ontologies. *IEEE Services Computing Workshops*: 169-174
- [24] E. Giallonardo, and E. Zimeo. 2007. More Semantics in QoS Matching. Paper presented at the IEEE International Conference on Service – Oriented Computing and Applications, June 19-20, in Newport Beach, California
- [25] Z. Xu, P. Martin, W. Powley, and F. Zulkernine. 2007. Reputation-Enhanced QoS-based Web Services Discovery. Paper presented at the IEEE International Conference on Web Services, 9-13 July, in Utah, USA
- [26] H. L. Truong, R. Samborski, and T. Fahringer. 2006. Towards a Framework for Monitoring and Analyzing QoS Metrics of Grid Services. Paper presented at the 2nd IEEE International Conference on e-Science and Grid Computing, 4-6 December, in Amsterdam, Netherlands
- [27] W. Zhang, Y. Yang, S. Tang, and L. Fang. 2007. QoS-driven Service Selection Optimization Model and Algorithms for Composite Web Services. Paper presented at the 31st Annual IEEE International Computer Software and Applications Conference (COMPSAC 2007), July 23-27, in Beijing, China
- [28] N. Kokash, and V. D. Andrea. 2006. Evaluating Quality of Web Services: A risk-driven approach. Technical Report DIT-06-093
- [29] M. Naseri, and A. Towhidi. 2007. QoS-Aware Automatic Composition of Web Services using AI Planners. Paper presented at the 2nd International Conference on Internet and Web Applications and Services, 13-19 May, in Mauritius
- [30] C. Wan, and H. Wang. 2007. Uncertainty-aware QoS Description and Selection Model for Web Services. Paper presented at the International Conference on Services Computing, 9-13 July, in Utah, USA
- [31] Y. Yang, S. Tang, Y. Xu, W. Zhang, and L. Fang. 2007. An Approach to QoS-aware Service Selection in Dynamic Web Service Composition. Paper presented at the International Conference on Networking and Services, 19-25 June, in Athens, Greece
- [32] L. Li, and I. Horrocks. 2003. A software framework for matchmaking based on semantic web technology. Paper presented at the 12th International World Wide Web Conference (WWW 2003), May 20-24, in Budapest, Hungary
- [33] D. Tsesmetzis, I. Roussaki, and E. Sykas. 2007. QoS-aware service evaluation and selection. *European Journal of Operational Research* (in press)
- [34] D. Tsesmetzis, I. Roussaki, and E. Sykas. 2007. Modeling and Simulation of QoS-aware Web Service Selection for Provider Profit Maximization. *Simulation* 83(1): 93-106
- [35] E. Cerami. 2002. *Web Services Essentials*. O'Reilly
- [36] 2006, "Web Services", http://en.wikipedia.org/wiki/Web_Services
- [37] 2003, "UDDI Version 3.0.1 UDDI Spec Technical Specification", <http://www.uddi.org/pubs/uddi-tech-wp.pdf>

- [38] D. Gourley, and B. Totty. 2002. *HTTP: The Definitive Guide*. O'Reilly
- [39] W. Rubin, and M. Brain. 1998. *Understanding DCOM*. Prentice Hall PTR
- [40] E. Gamma, R. Helm, R. Johnson, and J. M. Vlissides. 2000. *Design Patterns: Elements of Reusable Object – Oriented Software*. Addison – Wesley
- [41] 1982, “Simple Mail Transfer Protocol”, <http://www.faqs.org/rfcs/rfc821.html>
- [42] R. Lewis. 1999. *Advanced Messaging Applications with MSMQ and MQSeries*. Que
- [43] S. S. Laurent, E. Dumbill, and J. Johnston. 2002. *Programming Web Services with XML – RPC*. O'Reilly
- [44] D. Chappell, and T. Jewell. 2002. *Java Web Services*. O'Reilly
- [45] F. Curbera, M. Duffler, R. Khalaf, W. Nagy, N. Mukhi, and S. Weerawarana. Unraveling the Web services web: an introduction to SOAP, WSDL, and UDDI. *IEEE Internet Computing* 6(2): 86-93
- [46] F. P. Coyle. 2002. *XML, Web Services, and the Data Revolution*. O'Reilly
- [47] C. Musciano, and B. Kennedy. 2007. *HTML: The Definitive Guide, Third Edition*. O'Reilly
- [48] J. Cardoso. 2007. *Semantic Web Services: Theory, Tools, and Applications*. IGI Global
- [49] C. J. Date. 2000. *The Database Relational Model: A Retrospective and Analysis: A Historical Account and Assessment of E. F. Codd's Contribution to the Field of Database Technology*. Addison Wesley Longman
- [50] N. Balani. 2005. “The future of the Web is Semantic”, <http://www.ibm.com/developerworks/web/library/wa-semweb/>
- [51] T. Berners-Lee, R. T. Fielding, and L. Masinter. 2005. Uniform Resource Identifier (URI): Generic Syntax, <http://gbiv.com/protocols/uri/rfc/rfc3986.html>
- [52] S. Powers. 2003. *Practical RDF*. O'Reilly
- [53] M. C. Daconta, L. J. Obrst, and K. T. Smith. 2003. *The Semantic Web: A Guide to the Future of XML, Web Services and Knowledge Management*. Jonh Wiley & Sons
- [54] T. Budd. 2001. *An Introduction to Object-Oriented Programming*. Addison Wesley
- [55] M. E. Dutra. 2002. Ontologies for Web Services. Paper presented at Object Management Group's Web Services: From Technology to Reality; a Workshop on Modeling, Architectures, Infrastructures and Standards for Business Collaboration, March 4-7, in San Jose, CA
- [56] N. F. Noy, and D. L. McGuinness. 2001. “Ontology Development 101: A Guide to Creating your First Ontology”, http://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html
- [57] 2006, “Pelet: The Open Source OWL DL Reasoner”, <http://pellet.owldl.com/>
- [58] 2007, “KAON2 – Ontology Management for the Semantic Web”, <http://kaon2.semanticweb.org/>
- [59] 2008, “RacerPro”, <http://www.racer-systems.com/products/racerpro/index.phtml>
- [60] 2007, “Fact++”, <http://owl.man.ac.uk/factplusplus/>
- [61] M. Paolucci, T. Kawamura, T. R. Payne, and K. P. Sycara. 2002. Semantic Matching of Web Services Capabilities. Paper presented at the 1st International Semantic Web Conference on the Semantic Web, June 9-12, in Sardinia, Italy
- [62] K. Sycara, M. Paolucci, A. Ankolekar, and N. Srinivasan. 2003. Automated Discovery, Interaction and Composition of Semantic Web Services. *Journal of Web Semantics* 1(1): 27-46

- [63] L. Taher, H. El Khatib, and R. Basha. 2005. A Framework and QoS Matchmaking Algorithm for Dynamic Web Services Selection. Paper presented at the 2nd International Conference on Innovations in Information Technology (ITT), September 26-28, in Dubai, UAE
- [64] C. Zhou, L. T. Chia, and B.S. Lee. 2005. Web Services Discovery with DAML-QoS Ontology. *International Journal of Web Services Research* 2(2): 43-66
- [65] A. Mani, and A. Nagarajan. 2002. "Understanding quality of service for Web services". <http://www.ibm.com/developerworks/library/ws-quality.html>
- [66] V. Cardellini, M. Colajanni, and P. S. Yu. 1999. Dynamic Load Balancing on Web – Server Systems. *IEEE Internet Computing* 3(3): 28-39
- [67] D. A. Menascé. 2002. QoS Issues in Web Services. *IEEE Internet Computing* 6(6): 72-75
- [68] A. Keller, and H. Ludwig. 2003. The WSLA Framework: Specifying and Monitoring Service Level Agreements for Web Services. *Journal of Network and Systems Management* 11(1): 57-81
- [69] S. Ran. 2003. A Model for Web Services Discovery with QoS. *ACM SIGecom Exchanges* 4(1): 1-10
- [70] M. Maximilien, and P. Singh. 2004. A Framework and Ontology for Dynamic Web Services Selection. *IEEE Internet Computing* 8(5): 84-93
- [71] C. Patel, K. Superkar, and Y. Lee. 2003. A QoS Oriented Framework for Adaptive Management of Web Service based Workflows. Paper presented at the 14th International Database and Expert Systems Applications Conference (DEXA 2003), September 1-5, in Prague, Czech Republic
- [72] J. Meng, S. Y. W. Su, H. Lam, and A. Helal. 2002. Achieving Dynamic Inter-organizational Workflow Management by Integrating Business Processes, Events, and Rules. Paper presented at the 35th Hawaii International Conference on System Sciences (HICSS-35), January 7-10, in Hilton Waikoloa Village, Hawaii
- [73] A. Sánchez-Macián, L. Bellido, and E. Pastor. 2005. Paper presented at the V Jornadas de Ingeniería Telemática (Jitel 2005), September 12-14, in Pontevedra, Spain
- [74] V. A. Zeithaml, A. Parasuraman, and L. L. Berry. 1990. *Delivering Quality Service. Balancing Customer Perceptions and Expectations*. The Free Press, Macmillan, Inc. New York
- [75] M. Tian, A. Gramm, T. Naumowisz, H. Ritter, and J. Schiller. 2003. A concept for QoS Integration in Web Services. Paper presented at the 1st Web Services Quality Workshop (WQW 2003), December 13, in Rome, Italy
- [76] J. H. Gennari, M. A. Musen, R. W. Fergeson, W. E. Grosso, M. Crubézy, H. Eriksson, N. F. Nov, and S. W. Tu. 2003. The evolution of Protégé: an environment for knowledge – based systems development. *International Journal of Human – Computer Studies* 58(1): 89-123
- [77] K. Lee, J. Jeon, W. Lee, S. Jeong, and S. W. Park. 2003. "QoS for Web Services: Requirements and Possible Approaches", <http://www.w3c.or.kr/kr-office/TR/2003/NOTE-ws-qos-20031125/>
- [78] J. Brank, M. Grobelnik, and D. Mladenic. 2005. A survey of ontology evaluation techniques. Paper presented at the Conference on Data Mining and Data Warehouses, 10-17 October, in Ljubljana, Slovenia
- [79] A. Burton-Jones, V. C. Storey, V. Sugumaran, and P. Ahluwalia. 2005. A semiotic metrics suite for assessing the quality of ontologies. *Data and Knowledge*

- Engineering* 55(1): 84-102
- [80] 2008, Amigo-Ambient intelligence for the networked home environment, <http://www.amigo-project.org>
 - [81] E. Aarts, R. Harwig, and M. Schuurmans. 2001. *Ambient Intelligence in The Invisible Future: The Seamless Integration Of Technology Into Everyday Life*. McGraw – Hill Companies
 - [82] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. 1990. *Introduction to Algorithms*. McGraw-Hill, New York
 - [83] M. R. Garey, and D.S. Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, San Francisco
 - [84] D. E. Knuth. 1997. *The art of Computer Programming (Volumes 1-4)*. Addison-Wesley Professional
 - [85] A. Cobham. 1964. The intrinsic computational difficulty of functions. Paper presented at the International Congress for Logic Methodology and Philosophy of Science, 26 August – 2 September, in Jerusalem, Israel
 - [86] J. Edmonds. 1965. Minimum partition of a matroid into independent subsets. *Journal of Research of the National Bureau of Standards* 69B: 67-72
 - [87] A. Turing. 1936. On computable numbers with an application to the Entscheidungs Problem, *London Mathematics Society* 2(42): 230 – 265
 - [88] S. A. Cook. 1971. The complexity of theorem proving procedures Paper presented at the 3rd Annual ACM Symposium on Theory of Computing, May 3-5, in Ohio, USA
 - [89] S. Hochbaum. 1997. *Approximation algorithms for NP-hard problems*. PWS Publishing Company, Boston
 - [90] S. Martello, and P. Toth. 1990. *Knapsack Problems: Algorithms and Computer Implementations*. John Wiley & Sons, Inc
 - [91] K. Dudzinski, and S. Walukiewicz. 1987. Exact Methods for the Knapsack Problem and its Generalizations. *European Journal of Operational Research* 28: 3-21
 - [92] T. Ibaraki. 1987. Enumerative Approaches to Combinatorial Optimization, Part I, II. *Annals of Operations Research* 10,11
 - [93] O. H. Ibarra, and C. E. Kim. 1975. Fast approximation algorithms for the Knapsack and sum of subset problem. *Journal of ACM* 22: 463-468
 - [94] G. P. Ingrargiola, and J. F. Korsh. 1977. A general algorithm for the one-dimensional knapsack problem. *Operations Research* 25: 752-759
 - [95] D. Pisinger. 1995. *Algorithms for Knapsack Problems*. PhD Thesis
 - [96] R. E. Bellman. 1957. *Dynamic Programming*. Princeton University Press
 - [97] P. Toth. 1980. Dynamic programming algorithms for the zero-one knapsack problem. *Computing* 25: 29-45
 - [98] T. L. Morin, and R. E. Marsten. 1976. Branch and bound strategies for dynamic programming. *Operations Research* 24: 611-627
 - [99] P. J. Kolesar. 1967. A branch and bound algorithm for the knapsack problem. *Management Science* 13: 723-735
 - [100] G. T. Ross, and R. M. Soland. 1975. A branch and bound algorithm for the generalized assignment problem. *Mathematical Programming* 8: 91-103
 - [101] R. L. Bulfin, R. G. Parker, and C. M. Shetty. 1979. Computational results with a branch and bound algorithm for the general knapsack problem. *Naval Research Logistics Quarterly* 26: 41-46
 - [102] H. M. Salkin. 1975. *Integer Programming*. Addison-Wesley
 - [103] D. Fayard, and G. Plateau. 1994. An exact algorithm for the 0-1 collapsing knapsack

- problem. *Discrete Applied Mathematics* 49: 175-187
- [104] G. Kozanidis, and E. Melachrinoudis. 2004. A branch and bound algorithm for the 0-1 mixed integer knapsack problem with linear multiple choice constraints. *Computer and Operations Research* 31: 695-711
 - [105] G. B. Dantzig. 1957. Discrete Variable Extremum Problems. *Operations Research* 5: 266-277
 - [106] H. Kellerer, U. Pferschy, and D. Pisinger. 2005. *Knapsack Problems*. Springer Verlag
 - [107] R. D. Armstrong, D. S. Kung, P. Sihna, and A. A. Zoltners. 1983. A Computational Study of a Multiple-Choice Knapsack Algorithm. *ACM Transactions on Mathematical Software* 9(2): 184-198
 - [108] K. Klamroth, and M. M. Wiecek. 2000. *Dynamic Programming approaches to the multiple criteria knapsack problem*. John Wiley and Sons Inc
 - [109] A. Sinha, and A. A. Zoltners. 1979. The multiple-choice knapsack problem. *Operations Research* 27: 503-515
 - [110] M. E. Dyer. 1984. An O(n) algorithm for the LMCK program. *Mathematical Programming* 29: 57-63
 - [111] E. Zemel. 1980. The LMCKP. *Operations Research* 28(6): 1412-1423
 - [112] 2006, "The MathWorks - MATLAB - The Language of Technical Computing", <http://www.mathworks.com/products/matlab/>
 - [113] T. Yu, and K. J. Lin. 2005. Service Selection Algorithms for Web Services with End-to-end QoS Constraints. *Journal of Information Systems and E-Business Management* 3(2): 129-136
 - [114] T. Yu, and K.J. Lin. 2005. Service Selection Algorithms for Composing Complex Services with Multiple QoS Constraints. Paper presented at the 3rd International Conference on Service Oriented Computing (ICSOC2005), December 12-15, in Amsterdam, The Netherlands
 - [115] F. Harary, and E. Palmer. 1973. *Graphical Enumeration*. New York: Academic Press
 - [116] S. Khan, K. F. Li, E. G. Manning, and M. Akbar. 2002. Solving the knapsack problem for adaptive multimedia systems. *Studia Informatica Universalis* 2(1): 157-178
 - [117] R. Widjono. 1994. *The design and evaluation of routing algorithms for real-time channels*. Technical Rep. TR-94-024
 - [118] T. Korkmaz, and M. Krunz. 2001. Multi-Constrained Optimal Path Selection. Paper presented at the 20th Joint Conference IEEE Computer and Communication Societies, April 22-26, in Anchorage, Alaska
 - [119] D. Roman, H. Lausen, and U. Keller. 2005. *Web Service Modeling Ontology (WSMO)*. WSMO Final Draft
 - [120] S. Liu, Y. Liu, N. Jing, G. Tang, and Y. Tang. 2005. A dynamic web service selection strategy with QoS global optimization based on multi-objective genetic algorithm. Paper presented at the 4th Grid and Cooperative Computing International Conference (GCC 2005), November 30 - December 3, in Beijing, China
 - [121] X. P. Wang, and L.M. Cao. 2002. *Genetic Algorithms – Theory, Application and Software Implement*. XiAn: XiAn Jiao Tong University Press
 - [122] L. Zeng, B. Benatallah, M. Dumas, J. Kalagnanam, and Q.Z. Sheng. 2003. Quality Driven Web Services Composition. Paper presented at the 12th International Conference on the World Wide Web (WWW), May 20-24, in Budapest, Hungary

- [123] D. Harel, and A. Naamad. 1996. The STATEMATE Semantics of Statecharts. *ACM Transactions on Software Engineering and Methodology* 24(6): 429-456
- [124] M. Kkasal, and S. Zionts. 2001. *Multiple Criteria Decision Making in the New Millenium*. Springer – Verlag
- [125] H. C-L. Yoon, and K. Yoon. 1981. *Multiple Criteria Decision Making*. Lecture Notes in Economics and Mathematical Systems. Springer –Verlag
- [126] J. Pathak, N. Koul, D. Caragea, and V. Honavar. 2005. A Framework for Semantic Web Services Discovery. Paper presented at the 7th ACM International Workshop on Web Information and Data Management (WIDM 2005), November 5, in Bremen, Germany
- [127] A. S. Tanenbaum. 2003. *Computer Networks (Fourth Edition)*. Prentice Hall
- [128] R. S. Dembo, and P. L. Hammer. 1980. A Reduction Algorithm for Knapsack Problems. *Methods of Operations Research* 36: 49-60
- [129] E. Horowitz, and S. Sahni. 1974. Computing partitions with applications to the Knapsack Problem. *Journal of ACM* 21: 277-292
- [130] E. Balas, and E. Zemel. 1980. An Algorithm for Large Zero-One Knapsack Problems. *Operations Research* 28: 1130-1154