



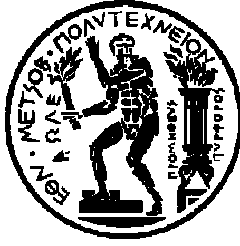
ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

Αντικειμενοστραφής Μεθοδολογία και Εργαλεία για
την Ολοκλήρωση Δομημένης και Ημι-Δομημένης
Πληροφορίας με Χρήση Οντολογιών

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

του
Γιάννη Γ. Σταυρούλα

Αθήνα, Νοέμβριος 2006



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ
ΠΛΗΡΟΦΟΡΙΚΗΣ

Αντικειμενοστραφής Μεθοδολογία και Εργαλεία για
την Ολοκλήρωση Δομημένης και Ημι-Δομημένης
Πληροφορίας με Χρήση Οντολογιών

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

ΤΟΥ

Ιωάννη Γ. Σταυρούλα

Συμβουλευτική Επιτροπή : Θ. Βαρβαρίγου
 Ε. Μ. Πρωτονοτάριος
 Ι. Παπανάνος

Εγκρίθηκε από την επταμελή εξεταστική επιτροπή την 27η Νοεμβρίου 2006.

.....
Θ. Βαρβαρίγου
Αναπληρώτρια
Καθηγήτρια Ε.Μ.Π.

.....
Ε. Μ. Πρωτονοτάριος
Καθηγητής Ε.Μ.Π.

.....
Ι. Παπανάνος
Καθηγητής Ε.Μ.Π.

.....
Γ. Στασινόπουλος
Καθηγητής Ε.Μ.Π.

.....
Α. Σταφυλοπάτης
Καθηγητής Ε.Μ.Π.

.....
Μ. Θεολόγου
Καθηγητής Ε.Μ.Π.

.....
Γ. Σταμούλης
Καθηγητής Ο.Π.Α.

Αθήνα, Νοέμβριος 2006

.....

Ιωάννης Γ. Σταυρούλας

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

© 2006 – All rights reserved

ΠΕΡΙΛΗΨΗ

Η εργασία που παρουσιάζεται στην ανά χείρας διατριβή μελετά πτυχές του προβλήματος της επικοινωνίας και συνεργασίας ετερογενών συστημάτων, στο επίπεδο της μορφολογίας, της σημασιολογίας και της προγραμματιστικής διαχείρισης της πληροφορίας που ανταλλάσσεται. Το πρόβλημα αυτό έχει συγκεντρώσει το επιστημονικό ενδιαφέρον από τις αρχές της δεκαετίας του 1990 και υπάρχουν πλέον αρκετές δόκιμες λύσεις για καθένα από τα επιμέρους προβλήματα που το απαρτίζουν. Στην παρούσα εργασία εξετάζονται οι δυνατότητες, οι περιορισμοί και οι τεχνικές για τον συγκερασμό αυτών των επιμέρους λύσεων σε μια ολοκληρωμένη μέθοδο αντιμετώπισης του συνολικού προβλήματος. Ως βασικός γνώμονας τίθεται η δυνατότητα πρακτικής εκμετάλλευσης της παραγόμενης μεθόδου σε ευρεία κλίμακα, και δίνεται έμφαση στην εξέταση των συνδετικών μηχανισμών που απαιτούνται για την συνένωση και συλλειτουργία των επιμέρους λύσεων σε ένα ολοκληρωμένο σύστημα.

Αναλυτικά, επιλέγεται η προσέγγιση της χρήσης οντολογιών για πληροφοριακά συστήματα, ως επιμέρους λύση στο πρόβλημα της σημασιολογικής ετερογένειας. Τα συνδεδεμένα προβλήματα της αναπαράστασης δεδομένων και γνώσης και της διασύνδεσης των εμπλεκόμενων συστημάτων σε

ένα λειτουργικό σύνολο, εξετάζονται υπό αυτό το πρίσμα. Προτείνεται μια αρχιτεκτονική για τη δόμηση συστημάτων που μοιράζονται πληροφορία με χρήση οντολογιών για τη σημασιολογική ταύτιση της μεταξύ τους. Παρουσιάζονται μέθοδοι και εργαλεία για την υποστήριξη της κατασκευής ενός τέτοιου συστήματος, στο οποίο μπορούν να συμμετέχουν πηγές δεδομένων δομημένης, ή ημι-δομημένης μορφής, με τον βασικό περιορισμό ότι το σύστημα αποσκοπεί σε ανάγνωση των δεδομένων και όχι σε εγγραφή τους.

Στη συνέχεια, εξετάζεται το ειδικότερο πρόβλημα της αναζήτησης δεδομένων εμπιστευτικού χαρακτήρα με χρήση αλγορίθμων αναζήτησης που δεν είναι εκ των προτέρων διαθέσιμοι στην πηγή. Αναπτύσσεται ένα σύστημα διαμεσολάβησης που επιτρέπει την ασφαλή εκτέλεση αναζητήσεων με άγνωστους στην πηγή αλγορίθμους, χωρίς διαρροή των εμπιστευτικών δεδομένων. Αναλύονται οι δυνατότητες εφαρμογής αλλά και οι περιορισμοί του συστήματος.

ABSTRACT

The work presented in the thesis at hand focuses on the problem of communication and interoperation of heterogeneous systems, at the levels of representation, semantics and programmatic management of the exchanged information. This problem has drawn the attention of the scientific community since the early 1990s, resulting in several well-tested solutions for each of its constituent problems. In this work we examine the possibilities, constraints and techniques related to the combination of such point solutions to an integrated solution addressing the complete problem. We focus on the applicability of the resulting method in wide scale rather than in the laboratory, and we elaborate on the connection mechanisms needed to enable the interoperation of these point solutions in a comprehensive system.

We select the ontologies for information systems as our preferred point solution to the semantic integration problem. The subsequent problems of knowledge and data representation and of the interconnection of constituent systems into an integrated entity are then addressed under this assumption. We propose an architecture for integrated systems that use ontologies to semantically resolve the information they share. Methods and tools are presented for supporting the construction of such an integrated system, which

can include sources of structured or semi-structured data, under the limitation that read-only access to the information is desired.

Finally, we address the problem of querying confidential data over an integrated system, using algorithms that are unknown to – and thus not approved by – the data provider. We present a middleware that allows untrusted search algorithms to be mediated and executed on the data source, without posing a data leakage risk. We discuss the application possibilities as well as the limitations of this system.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΠΕΡΙΛΗΨΗ	5
ABSTRACT	7
1 ΕΙΣΑΓΩΓΗ	13
2 ΟΝΤΟΛΟΓΙΑ ΚΑΙ ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ	18
2.1 Η ΣΥΛΛΗΨΗ ΚΑΙ ΕΞΕΛΙΞΗ ΤΗΣ ΟΝΤΟΛΟΓΙΑΣ	19
2.2 ΟΝΤΟΛΟΓΙΕΣ ΓΙΑ ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ	29
2.3 Η ΕΠΙΔΡΑΣΗ ΤΗΣ ΟΝΤΟΛΟΓΙΑΣ ΣΤΗ ΣΧΕΔΙΑΣΗ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ	36
3 ΣΧΕΔΙΑΣΗ ΟΝΤΟΛΟΓΙΩΝ ΓΙΑ ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ	44
3.1 ΔΟΜΙΚΑ ΣΥΣΤΑΤΙΚΑ.....	45
3.2 ΣΧΕΔΙΑΣΤΙΚΕΣ ΑΡΧΕΣ	47
3.3 ΜΕΘΟΔΟΛΟΓΙΕΣ ΑΝΑΠΤΥΞΗΣ ΟΝΤΟΛΟΓΙΩΝ	49
3.4 ΜΕΘΟΔΟΛΟΓΙΕΣ ΑΝΑΣΧΕΔΙΑΣΜΟΥ ΟΝΤΟΛΟΓΙΩΝ.....	73
3.5 ΣΥΝΟΨΗ ΚΑΙ ΣΥΜΠΕΡΑΣΜΑΤΑ.....	77
4 ΓΛΩΣΣΕΣ ΚΩΔΙΚΟΠΟΙΗΣΗΣ ΟΝΤΟΛΟΓΙΩΝ	81
4.1 ΠΛΑΙΣΙΟ ΑΞΙΟΛΟΓΗΣΗΣ	81
4.2 ΚΛΑΣΣΙΚΕΣ ΓΛΩΣΣΕΣ ΠΕΡΙΓΡΑΦΗΣ ΟΝΤΟΛΟΓΙΩΝ	89
4.3 ΓΛΩΣΣΕΣ ΟΝΤΟΛΟΓΙΩΝ ΒΑΣΙΣΜΕΝΕΣ ΣΕ XML	110
4.4 ΣΥΓΚΡΙΣΗ – ΣΥΜΠΕΡΑΣΜΑΤΑ	124
5 ΟΛΟΚΛΗΡΩΣΗ ΠΗΓΩΝ ΠΛΗΡΟΦΟΡΙΑΣ ΜΕ ΟΝΤΟΛΟΓΙΕΣ.....	137
5.1 ΥΠΑΡΧΟΥΣΕΣ ΠΡΟΣΕΓΓΙΣΕΙΣ ΣΤΗΝ ΟΝΤΟΛΟΓΙΚΗ ΟΛΟΚΛΗΡΩΣΗ	140

5.2	ΠΡΟΒΛΗΜΑΤΑ	144
5.3	ΤΟ ΟΚΒC.....	149
6	ΟΖΟΝΕ	151
6.1	ΑΝΑΠΑΡΑΣΤΑΣΗ ΓΝΩΣΗΣ	154
6.2	ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΚΑΙ ΛΕΙΤΟΥΡΓΙΑ	164
6.3	ΤΥΠΟΠΟΙΗΣΗ ΤΩΝ INTERFACES	177
6.4	ΚΑΤΑΣΚΕΥΗ INTERFACES ΠΡΟΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ.....	182
6.5	ΟΛΟΚΛΗΡΩΣΗ ΗΜΙ-ΔΟΜΗΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΑΣ	192
6.6	ΠΕΙΡΑΜΑΤΙΚΗ ΥΛΟΠΟΙΗΣΗ	193
7	ΑΣΦΑΛΗΣ ΑΝΑΖΗΤΗΣΗ ΗΜΙ-ΔΟΜΗΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΑΣ.....	199
7.1	ΤΟ ΠΡΟΒΛΗΜΑ.....	201
7.2	ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΚΑΙ ΑΣΦΑΛΕΙΑ	207
7.3	ΕΦΑΡΜΟΓΗ ΤΗΣ ΜΕΘΟΔΟΥ ΩΣ ΑΣΦΑΛΗΣ ΥΠΗΡΕΣΙΑ.....	220
7.4	ΕΦΑΡΜΟΓΕΣ	232
8	ΣΥΜΠΕΡΑΣΜΑΤΑ.....	240
	ΒΙΒΛΙΟΓΡΑΦΙΑ.....	246

ΠΙΝΑΚΑΣ ΣΧΗΜΑΤΩΝ

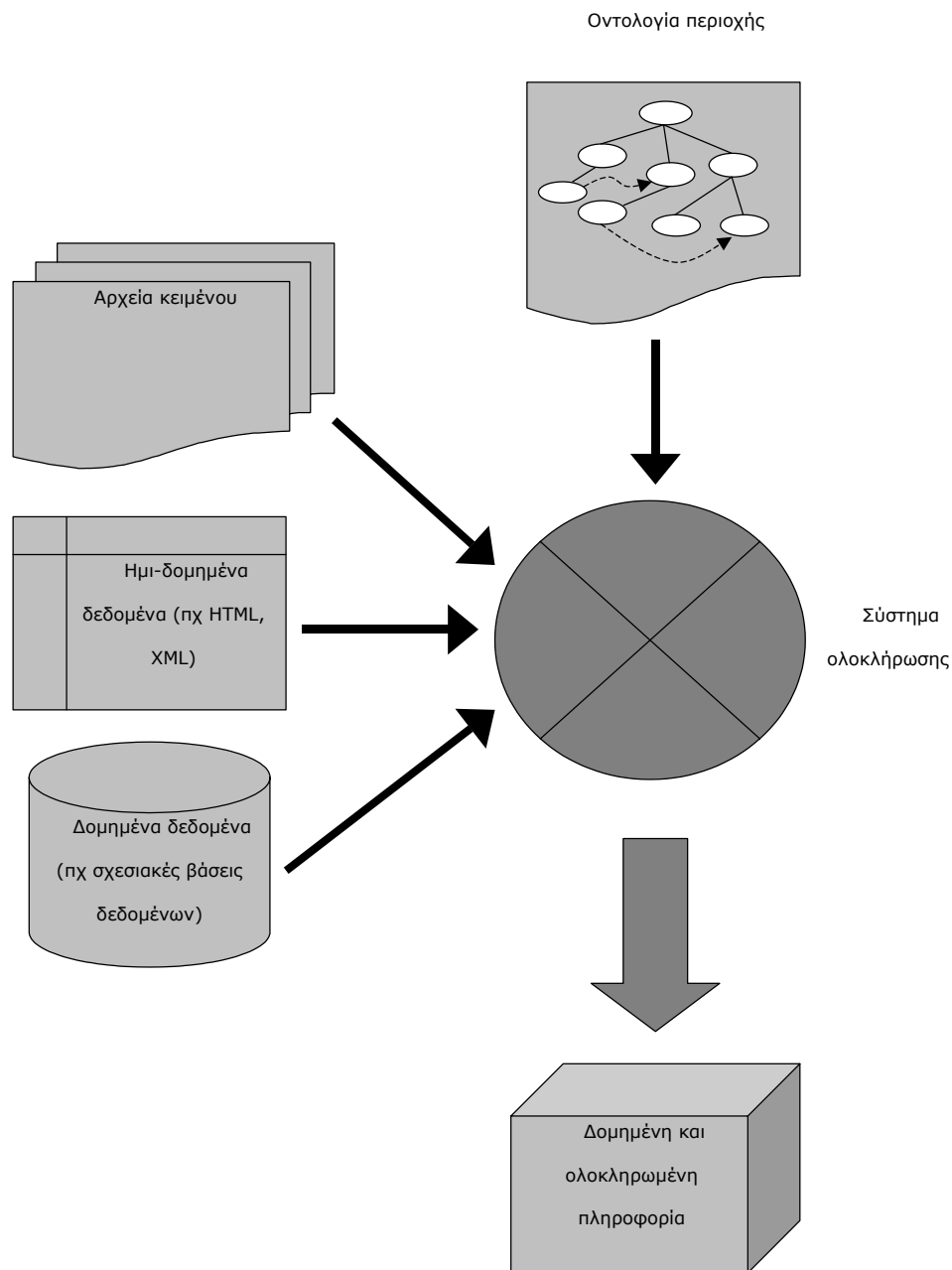
Σχήμα 1: Εννοιολογική ολοκλήρωση πληροφορίας από δομημένες, ημι-δομημένες και αδόμητες πηγές, με χρήση οντολογίας.....	14
Σχήμα 2: Ταξινόμηση διαφορετικών ερμηνειών του όρου οντολογία, ανάλογα με την πολυπλοκότητα.....	25
Σχήμα 3: Είδη οντολογιών ανάλογα με τη γενικότητα, τη λεπτομέρεια και την εξάρτηση από συγκεκριμένες εφαρμογές [Guarino, 1997]	35
Σχήμα 4: Η διαδικασία ανασχεδιασμού οντολογιών	74
Σχήμα 5: Τα στάδια του ανασχεδιασμού οντολογιών [Gomez Perez, 1999]	77
Σχήμα 6. Το πλαίσιο αξιολόγησης γλωσσών ανάπτυξης οντολογιών του CommonKADS	82
Σχήμα 7: Σύνοψη των χαρακτηριστικών μοντελοποίησης γνώσης	126
Σχήμα 8: Υποστήριξη χαρακτηριστικών κλάσεων	127
Σχήμα 9: Δυνατότητες ορισμού ταξινομιών.....	129
Σχήμα 10: Υποστήριξη σχέσεων και συναρτήσεων	130
Σχήμα 11: Χαρακτηριστικά αξιωμάτων	132
Σχήμα 12: Υποστήριξη παραγωγικών κανόνων.....	134
Σχήμα 13: Δυνατότητες μηχανής λογικής.....	135
Σχήμα 14: Ο αντίκτυπος της μεθοδολογίας Ozone στην προσπάθεια και τις γνώσεις που απαιτούνται για να ενταχθεί μια βάση δεδομένων σε μια ομοσπονδία βάσεων	153
Σχήμα 15: Σύγκριση δυνατοτήτων αναπαράστασης μεταξύ Ozone και Ontolingua.....	156
Σχήμα 16: Οι βασικές κλάσεις του Ozone σε UML, με την μετακλάση (RawConceptData) στο πάνω μέρος.....	159
Σχήμα 17: Η ιεραρχία της κλάσης ValueRange σε UML	161

Σχήμα 18: Η ορατότητα των οντολογιών και η έκταση χρήσης γλώσσας περιγραφής οντολογίας στην κλασική προσέγγιση ολοκλήρωσης με οντολογίες (αριστερά) και στο Ozone (δεξιά).....	165
Σχήμα 19: Η αρχιτεκτονική του Ozone.....	170
Σχήμα 20: Τα interfaces ανάμεσα στο Ozone και τα υπόλοιπα συστατικά του ολοκληρωμένου συστήματος σε UML.....	178
Σχήμα 21: Αντιστοίχιση μιας οντολογικής έννοιας σε οντότητες της τοπικής βάσης	186
Σχήμα 22: Αντιστοίχιση οντολογικών ιδιοτήτων στην τοπική βάση.....	187
Σχήμα 23: Αντιστοίχιση οντολογικών σχέσεων στην τοπική βάση	190
Σχήμα 24: Σχηματική αναπαράσταση της βασικής εφαρμογής της μεθόδου.....	206
Σχήμα 25: Η αρχιτεκτονική του PIVOTS	209
Σχήμα 26: Απλοποιημένος κώδικας Java των βασικών τμημάτων του συστήματος	215
Σχήμα 27: Σχηματική αναπαράσταση της λειτουργίας ασφαλούς υπηρεσίας αναζήτησης, βασισμένης στην προτεινόμενη μέθοδο.....	221
Σχήμα 28: Παράδειγμα εφαρμογής της μεθόδου παραμόρφωσης κειμένου για αποτροπή ανάγνωσης του από μηχανή	227

1 ΕΙΣΑΓΩΓΗ

Οι υπολογιστικές πηγές πληροφοριών αποθηκεύουν και παρέχουν τέτοια ποσά στοιχείων ώστε η πρόσβαση, εύρεση ή σύνοψη πληροφοριών παραμένει ένας δύσκολος στόχος δεδομένου του μεγάλου όγκου πληροφοριών που βρίσκονται σε κάθε πηγή και του μεγάλου αριθμού και της ποικιλίας των διαθέσιμων πηγών μέσω των τρεχουσών τεχνολογιών, όπως ο Παγκόσμιος Ιστός (WWW). Οι λόγοι που κρύβονται πίσω από αυτό το πρόβλημα είναι πολλαπλοί, εντούτοις μια από τις σημαντικότερες αιτίες βρίσκεται στο μεγάλο χάσμα ανάμεσα στην αντίληψη των πληροφοριών, όπως αυτή πραγματοποιείται από τον τελικό χρήστη, και την οργάνωση, αποθήκευση και διάθεση των πληροφοριών στο υπολογιστικό σύστημα που τις παρέχει. Σε γενικές γραμμές, η ερώτηση κατά πόσο το χάσμα αυτό μπορεί να γεφυρωθεί παραμένει ανοιχτή, πόσο μάλλον η ερώτηση πώς να το γεφυρώσει κανείς με έναν τρόπο που να μειώνει το τεχνικά επίπονο έργο της παροχής μεθόδων εξαγωγής πληροφοριών για μεγάλους αριθμούς και διαφορετικά είδη πηγών πληροφοριών. Το απλό κείμενο (free text), οι ημι-δομημένες πληροφορίες (semi-structured information), π.χ. XML, και οι πληροφορίες σχεσιακών βάσεων δεδομένων παρουσιάζουν όλες παρόμοια προβλήματα όταν επιχειρείται η διαμόρφωση ενός

κοινού, γενικευμένου σχήματος για την κατανόηση και ερμηνεία της πληροφορίας.



Σχήμα 1: Εννοιολογική ολοκλήρωση πληροφορίας από δομημένες, ημι-δομημένες και αδόμητες πηγές, με χρήση οντολογίας

Οι οντολογίες περιγράφουν κοινές εννοιολογικές αντιλήψεις (conceptualisations) για συγκεκριμένες περιοχές ενδιαφέροντος σε ένα υψηλό επίπεδο αφαίρεσης, όσο αφορά τις τεχνικές συνιστώσες του προβλήματος. Αντί

να εξετάσουν τα προβλήματα εφαρμογής και υλοποίησης, οι οντολογίες περιοχών ενδιαφέροντος (domain ontologies) περιγράφουν ακριβώς τις σχετικές με κάθε περιοχή έννοιες, τις σχέσεις τους, και τους κανόνες για αυτές τις σχέσεις που επιβάλλουν μια καθορισμένη με σαφήνεια σημασιολογία στην αντίληψη και ερμηνεία των πληροφοριών. Όσον αφορά το πρόβλημα εξαγωγής και ολοκλήρωσης πληροφοριών που αναφέρθηκε παραπάνω, οι τυποποιημένες (formal) οντολογίες επιτρέπουν την ακριβή περιγραφή μιας σημασιολογίας κοινής για ποικίλες πηγές πληροφοριών. Κατά συνέπεια, οι οντολογίες προσφέρονται ως (μερικές) λύσεις στα προβλήματα της εξαγωγής πληροφορίας και παρουσίασης της στο χρήστη, ή της εισαγωγής της σε μια άλλη, καλά γνωστή βάση δεδομένων με σαφή τρόπο (Σχήμα 1).

Παρακάτω παρουσιάζεται μια ανασκόπηση σε τεχνικές που, χρησιμοποιώντας οντολογίες, επιτρέπουν την εξαγωγή και την ολοκλήρωση πληροφοριών από διαφορετικούς τύπους πηγών δεδομένων. Παραδείγματος χάριν, η εξαγωγή πληροφοριών με οντολογίες από ελεύθερο κείμενο ερευνάται στο πρόγραμμα GETESS [Staab, 1999] και λύσεις για την εξαγωγή και την ολοκλήρωση πληροφοριών από ημι-δομημένες πηγές πληροφοριών έχουν αναπτυχθεί στο πρόγραμμα OntoBroker[Decker, 1999]. Για την εξαγωγή και την ολοκλήρωση πληροφοριών συνήθως στηριζόμαστε σε διάφορα εργαλεία οντολογιών, συμπεριλαμβανομένου ενός συντακτικού προγράμματος (ontology editor) και ενός εξυπηρετητή οντολογιών (ontology server).

Ωστόσο, πέρα από αυτά τα εργαλεία για τη δημιουργία και την πρόσβαση σε οντολογίες, η δόμηση ενός ολοκληρωμένου πληροφοριακού συστήματος βασισμένου σε οντολογίες απαιτεί την ανάπτυξη και διασύνδεση αρκετών ακόμα υποσυστημάτων. Έτσι, στην παρακάτω μελέτη επιχειρώ μια κριτική θεώρηση των οντολογικών τεχνικών ολοκλήρωσης, προσπαθώντας να εντοπίσω τα πλεονεκτήματα και μειονεκτήματα τους. Στην συνέχεια συνοψίζω τις βασικές αδυναμίες των τυποποιημένων μεθόδων και ορίζω το πλαίσιο εργασίας για τη

διατριβή μου, με σκοπό να βελτιώσω αυτές τις αδυναμίες. Η δομή της εργασίας έχει ως εξής:

Αρχικά γίνεται μια ανασκόπηση στην χρήση των οντολογιών σε πληροφοριακά συστήματα, με έμφαση στα σχεδιαστικά και μεθοδολογικά προβλήματα που η χρήση τους εισάγει. Έτσι, στο κεφάλαιο 2 εξετάζονται οι διαφορετικές προσεγγίσεις στην χρήση οντολογιών σε πληροφοριακά συστήματα και η επίδραση τους στη δόμηση και λειτουργία τους.

Στο κεφάλαιο 3 εξετάζεται το γενικό πρόβλημα σχεδίασης οντολογιών, αναλύονται οι βασικές σχεδιαστικές αρχές και γίνεται μια συγκριτική μελέτη των μεθοδολογιών σχεδίασης οντολογιών, με στόχο να καταστεί προφανής η έλλειψη ωριμότητας του πεδίου, όπως αυτή αντικατοπτρίζεται στην έλλειψη σύγκλισης ανάμεσα σε διαφορετικές μεθοδολογίες και μοντέλα, και η δυσκολία εφαρμογής των προτεινόμενων μεθοδολογιών σε βιομηχανικά περιβάλλοντα.

Στο κεφάλαιο 4 εξετάζονται υπό αντίστοιχο πρίσμα οι γλώσσες κωδικοποίησης οντολογιών. Υιοθετείται ένα πλαίσιο αξιολόγησης για τη συγκριτική μελέτη τους, βάσει του οποίου αναλύονται τα χαρακτηριστικά, οι δυνατότητες, τα πλεονεκτήματα και τα μειονεκτήματα κάθε γλώσσας. Καθίσταται προφανής η αδυναμία μιας γλώσσας να ικανοποιήσει όλες τις παραμέτρους του πλαισίου αξιολόγησης ταυτόχρονα, καθώς είναι αντικρουόμενες μεταξύ τους, και αναπτύσσεται το επιχείρημα της ανάγκης συνύπαρξης περισσότερων τις μίας γλωσσών σε ένα ολοκληρωμένο σύστημα, για την κάλυψη διαφορετικών αναγκών.

Στο κεφάλαιο 5 γίνεται βιβλιογραφική ανασκόπηση των προσεγγίσεων στην ολοκλήρωση συστημάτων με χρήση οντολογιών. Καταδεικνύεται αφενός η χρησιμότητα αυτών των προσεγγίσεων, δηλαδή η ικανότητα τους να παράσχουν ικανοποιητικές λύσεις στο πρόβλημα της ολοκλήρωσης πληροφορίας, και αφετέρου το κενό που υπάρχει στον τομέα της αρχιτεκτονικής και της μεθοδολογίας ανάπτυξης τέτοιων ολοκληρωμένων συστημάτων, που πηγάζει

από τα προβλήματα και τις ιδιαιτερότητες των οντολογικών γλωσσών και μεθοδολογιών. Εντοπίζονται τα συγκεκριμένα σημεία που πρέπει να αντιμετωπιστούν για να καλυφθεί αυτό το κενό.

Στο κεφάλαιο 6 περιγράφεται η μεθοδολογία Ozone, όπως επίσης η αρχιτεκτονική και το μοντέλο αναπαράστασης γνώσης στο οποίο στηρίζεται. Αναλύονται τα πλεονεκτήματα και η συμβολή στην αντιμετώπιση των προβλημάτων που εντοπίστηκαν στο προηγούμενο κεφάλαιο. Περιγράφεται το πλαίσιο της πειραματικής υλοποίησης και τα συμπεράσματα από την αξιολόγηση της.

Στο κεφάλαιο 7 παρουσιάζεται μια μέθοδος που αντιμετωπίζει το πρόβλημα αναζήτησης και ανάκτησης πληροφορίας όταν το περιεχόμενο δεν είναι ελεύθερα διαθέσιμο. Εξηγείται η ανάγκη για μια τεχνική αξιολόγησης πληροφορίας χωρίς την πλήρη αποκάλυψη της, τα εγγενή προβλήματα ασφαλείας που εισέρχονται και οι διαθέσιμοι τρόποι αντιμετώπισης τους. Εξετάζεται η εφικτότητα χρήσης της μεθόδου σε διαφορετικές εφαρμογές.

Τέλος, στο κεφάλαιο 8 συνοψίζονται τα ευρήματα της εργασίας, τα πλεονεκτήματα των προτεινόμενων λύσεων και τα γενικά συμπεράσματα. Επίσης γίνεται αναφορά στις κατευθύνσεις που μπορούν να ακολουθηθούν για την μελλοντική ανάπτυξη των λύσεων και την προσαρμογή τους στα βιομηχανικά δεδομένα, σε σχέση με τις τελευταίες τεχνολογικές τάσεις.

2 ΟΝΤΟΛΟΓΙΑ ΚΑΙ ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ

Σήμερα η έρευνα για την οντολογία γίνεται όλο και περισσότερο διαδεδομένη στην κοινότητα της πληροφορικής. Παρά το ότι αυτός ο κλάδος έχει περιοριστεί στη φιλοσοφική σφαίρα στο παρελθόν, σήμερα κερδίζει έναν συγκεκριμένο ρόλο στην τεχνητή νοημοσύνη, την υπολογιστική γλωσσολογία και τη θεωρία βάσεων δεδομένων. Ειδικότερα, η σημασία της αναγνωρίζεται σε διαφορετικούς ερευνητικούς τομείς όπως την μηχανική γνώσης [Gruber, 1993], [Uschold, 1996], [Gaines, 1997], [Gomez Perez, 1998], την αναπαράσταση γνώσης [Artale, 1996], τη γλωσσική τεχνολογία [Lang, 1991], [Bateman, 1995], το σχεδιασμό βάσεων δεδομένων [Burg, 1997], [Van de Riet, 1998], τη μοντελοποίηση πληροφορίας [Ashenhurst, 1996], την ολοκλήρωση πληροφοριών [Wiederhold, 1996], [Bergamaschi, 1998], [Mena, 1998], την αντικειμενοστραφή ανάλυση [Wand, 1989], [Pazzi, 1998], την εύρεση και εξαγωγή πληροφοριών [Benjamins, 1998], [Guarino, 1997], [McGuinness, 1998], τη διαχείριση και οργάνωση γνώσης [Poli, 1996], το σχεδιασμό συστημάτων βασισμένων σε ευφυείς πράκτορες, και άλλους. Οι υπάρχοντες τομείς εφαρμογών είναι ανάμοιοι, για παράδειγμα επιχειρησιακή ολοκλήρωση (enterprise integration) [Gruninger, 1995], μετάφραση φυσικής

γλώσσας [Knight, 1994], [Mahesh, 1996], ιατρική [Gangemi, 1998], μηχανολογία [Borst, 1997], ηλεκτρονικό εμπόριο [Lehmann, 1995], γεωγραφικά συστήματα πληροφοριών [Casati, 1998]. Στη συνέχεια χρησιμοποιούμε τον όρο συστήματα πληροφοριών (Information Systems, IS) με τη γενικότερη έννοια του, για να αναφερθούμε σε όλες αυτές τις διαφορετικές περιοχές εφαρμογών. Στις επόμενες παραγράφους δίνεται μια εποπτική περιγραφή της οντολογίας ως φιλοσοφικό εργαλείο, και της σχέσης και χρησιμότητας της στα πληροφοριακά συστήματα.

2.1 Η ΣΥΛΛΗΨΗ ΚΑΙ ΕΞΕΛΙΞΗ ΤΗΣ ΟΝΤΟΛΟΓΙΑΣ

Η φιλοσοφική οντολογία είναι η επιστήμη των όντων, των ειδών και των δομών των αντικειμένων, των ιδιοτήτων, των γεγονότων, των διαδικασιών και των σχέσεων σε κάθε τομέα της πραγματικότητας. Η φιλοσοφική οντολογία εμφανίστηκε στις πρώιμες μορφές της στη μεταφυσική του Αριστοτέλη. Μετά τον Αριστοτέλη, ο όρος «οντολογία» χρησιμοποιήθηκε για πρώτη φορά το 1613, ανεξάρτητα από δύο φιλοσόφους, το Rudolf Göckel, στο έργο του *Lexicum philosophicum*, και τον Jacob Lorhard στο έργο του *Theatrum philosophicum*. Η πρώτη εμφάνιση του στα αγγλικά όπως καταγράφεται από το Oxford English Dictionary εμφανίζεται στο λεξικό Bailey του 1721, το οποίο καθορίζει την οντολογία ως «ένας αφηρημένος απολογισμός της ύπαρξης» [Guarino, 1998].

Πέρα από την ιστορία του όρου, αυτό που σήμερα αναφέρουμε ως φιλοσοφική οντολογία είναι η επιδίωξη για την σαφή και εξαντλητική ταξινόμηση των οντοτήτων σε όλες τις σφαίρες της ύπαρξης. Μπορεί έτσι να συλληφθεί ως είδος γενικευμένης χημείας. Οι ταξινομήσεις που προκύπτουν από τη φιλοσοφική οντολογία έχουν προοριστεί για να είναι σαφείς υπό την έννοια ότι θα μπορούσαν να χρησιμεύσουν ως οι απαντήσεις σε ερωτήσεις όπως: Ποιες κατηγορίες οντοτήτων απαιτούνται για μια πλήρη περιγραφή και εξήγηση όλων των αλληλεπιδράσεων του κόσμου; Ή: Ποιες κατηγορίες οντοτήτων απαιτούνται

για να δώσουν μια περιγραφή αυτού που καθιστά αληθινές όλες τις αλήθειες; Έχουν ως σκοπό να είναι εξαντλητικές υπό την έννοια ότι όλοι οι τύποι οντοτήτων πρέπει να περιληφθούν, συμπεριλαμβάνων επίσης των τύπων σχέσεων με τους οποίους οι οντότητες είναι συνδεδεμένες.

Για περίπου 2000 έτη μετά τον Αριστοτέλη, η οντολογία δεν αναπτύχθηκε σχεδόν καθόλου, σε τέτοιο βαθμό που αποτέλεσε ένα κεντρικό μέρος αυτού που αναφέρουμε σήμερα ως αρχαιοελληνική φιλοσοφία. Με την επιστημονική επανάσταση, εντούτοις, οι φιλοσοφικές ταξινομήσεις άρχισαν να απεικονίζουν μερικές φορές ριζικές εξελίξεις στην κατανόησή του κόσμου μας, και οι γενικά αποδεκτές ταξινομήσεις άρχισαν να αλλάζουν. Οι διαφορετικές σχολές της φιλοσοφίας έχουν προσφέρει διαφορετικές προσεγγίσεις στην παροχή μιας τέτοιας ταξινόμησης. Ένα μεγάλο χάσμα υπάρχει μεταξύ των φιλοσόφων που επιδίωξαν μια οντολογία βασισμένη στα υποκείμενα και εκείνων που επιδίωξαν μια οντολογία βασισμένη στα γεγονότα ή τις διαδικασίες.

Ακριβώς όπως οι ρίζες της οντολογίας συνδυάστηκαν με την πρόωρη ανάπτυξη της φιλοσοφίας και αναπτύχθηκαν μαζί της, έτσι τα τελευταία χρόνια, η οντολογία έχει συσχετιστεί έντονα με την ανάπτυξη της τεχνητής νοημοσύνης και της επιστήμης συστημάτων πληροφοριών [Smith, 2001]. Από την ίδια την αρχή της, η τεχνητή νοημοσύνη έστρεψε την προσοχή της στα συστήματα που κατέχουν γνώση, ή έχουν τη δυνατότητα να μιμηθούν τη γνώση, μέσω της χρήσης αυτοματοποιημένων μηχανισμών συμπερασμάτων. Δεδομένου ότι αυτοί οι μηχανισμοί έγιναν περισσότερο τυποποιημένοι με την πάροδο του χρόνου, οι θεωρίες που εκφράστηκαν με βάση αυτούς έγιναν εστία της επιστημονικής προσοχής. Αυτές οι θεωρίες, που αποκλήθηκαν βάσεις γνώσεων προτού ακόμα αρχίσει ο όρος να συγχέεται έντονα με τα έμπειρα συστήματα, αναφέρονται σε συλλογές όρων συνοδευόμενες από τα αντίστοιχα αξιώματα, σχεδιασμένες με σκοπό να περιορίσουν τις ανεπιθύμητες ερμηνείες

των όρων αυτών και να επιτρέψουν την εξαγωγή νέων πληροφοριών από υπάρχοντα δεδομένα.

Η λειτουργία αυτοματοποιημένων μηχανισμών συλλογισμού βασισμένων σε τέτοιες βάσεις γνώσεων θεωρήθηκε παράδειγμα τεχνητής νοημοσύνης. Οι βάσεις γνώσεων κατασκευάστηκαν συχνά με τρόπους που και απεικόνισαν την ανθρώπινη γνώση κοινής λογικής με έναν δηλωτικό (declarative) τρόπο, αλλά και εκμεταλλεύθηκαν τις δυνάμεις του ιδιαίτερου αυτοματοποιημένου συστήματος συλλογισμού που χρησιμοποιήθηκε.

Είναι σημαντικό να σημειωθεί ότι μεγάλη μερίδα της επιστημονικής κοινότητας θεώρησε, και θεωρεί ακόμα, ότι οι μηχανισμοί συλλογισμού οι ίδιοι είναι η πλέον σημαντική επιστημονική πρόκληση, και οι βάσεις γνώσεων δεν είναι τίποτα περισσότερο από βοηθητικά εργαλεία που δεν έχουν καμία εγγενή σημασία. Άλλοι, εντούτοις, υποστήριξαν ότι οι βάσεις γνώσεων οι ίδιες αποτελούν σημαντικό θέμα για επιστημονική έρευνα. Οι εργασίες σε αυτή την κατεύθυνση έδειξαν τον υψηλό βαθμό αυθαιρεσίας που χαρακτήριζε τις υπάρχουσες βάσεις γνώσεων, και την έλλειψη ακαμψίας που υπήρξε εγγενής στην ανάπτυξή τους. Κατ' αυτό τον τρόπο ο τομέας της εφαρμοσμένης μηχανικής γνώσης γεννήθηκε [Welty, 1999], όχι όμως εγκαίρως για να αποτρέψει τις αποτυχημένες προσπάθειες να καθιερωθούν και να εμπορευματοποιηθούν τα έμπειρα συστήματα, τα οποία αποτέλεσαν ένα υπεραπλουστευμένο προϊόν της έρευνας στην περιοχή αυτή.

Ανεξάρτητα από αυτές τις εξελίξεις, ερευνητές στο πεδίο των συστημάτων διαχείρισης βάσεων δεδομένων (DBMS) επίσης διαπίστωναν ότι, μετά από την ορολογία βάσεων δεδομένων που η τεχνολογία είχε αρχίσει να σταθεροποιεί, το πολύ σημαντικότερο και λεπτό πρόβλημα εννοιολογικής μοντελοποίησης (Conceptual modelling) ακόμα παρέμενε άλυτο [Jardine, 1997]. Η πρώτη περίοδος της εννοιολογικής μοντελοποίησης βάσεων δεδομένων χαρακτηρίστηκε ως επί το πλείστον από την ειδική, κατά περίπτωση και άτυπη

μοντελοποίηση, οδηγώντας στα πολλά πρακτικά προβλήματα ολοκλήρωσης βάσεων δεδομένων που αντιμετωπίζουμε σήμερα.

Παράλληλα ακόμα ένα πεδίο της πληροφορικής, η τεχνολογία λογισμικού, ιδιαίτερα μετά από τις προόδους στις αντικειμενοστραφείς γλώσσες, άρχισε να αναγνωρίζει τη σημασία αυτού που πλέον αναφέρουμε ως μοντελοποίηση περιοχών (domain modelling) [Arango, 1991]. Η εργασία του Arango απεικόνισε μια κατάσταση στην οποία πολύ συγκεκριμένα πρακτικά προβλήματα έπρεπε να αντιμετωπιστούν από τους υπεύθυνους για την ανάπτυξη λογισμικού, όπως το αυξανόμενο μέγεθος και η πολυπλοκότητα των προγραμμάτων, που προκαλούσαν στη συνέχεια δυσκολίες στη διατήρηση τέτοιων προγραμμάτων και την αξιοποίηση τους σε νέες χρήσεις. Η προτεινόμενη λύση ήταν να χτιστούν δηλωτικές αναπαραστάσεις των διαδικασιών που κάποιος προσπαθεί να μοντελοποιήσει – παραδείγματος χάριν επιχειρησιακές διαδικασίες ταξινόμησης ή χρονοδρομολόγησης – με έναν τρόπο που θα επέτρεπε στα συστήματα εφαρμογής να επαναχρησιμοποιήσουν στοιχεία κώδικα [Frank, 1997]. Αυτός ο τομέας, αντίστοιχα με αυτούς που αναφέρθηκαν παραπάνω, επηρεάστηκε σοβαρά από την έλλειψη συγκεκριμένων και τυποποιημένων πλαισίων για τη λήψη αποφάσεων μοντελοποίησης.

2.1.1 Η Εισαγωγή της Οντολογίας στην Πληροφορική

Η μετάβαση από κάθε μια από αυτές τις τρεις αφετηρίες στην οντολογία είναι έπειτα σχετικά εύκολη. Ο μηχανικός γνώσης, ο υπεύθυνος εννοιολογικού σχεδιασμού, ή ο domain modeller αντιλαμβάνονται την ανάγκη για δηλωτικές αναπαραστάσεις που πρέπει να έχουν όσο το δυνατόν περισσότερη γενικότητα για να εξασφαλίσουν ικανότητα επαναχρησιμοποίησης, αλλά αντιστοιχούν συγχρόνως στα πράγματα και τις διαδικασίες που πρέπει να αντιπροσωπεύσουν. Κατά συνέπεια αναγκάζονται να θέσουν ερωτήσεις όπως: Τι είναι ένα αντικείμενο, διαδικασία, χαρακτηριστικό ή σχέση; Τι είναι μια συναλλαγή, ένα

πρόσωπο, μια οργάνωση; Πώς εξαρτώνται ο ένας από τον άλλον; Πώς σχετίζονται;

Η μετάβαση στην οντολογία έγινε στην πράξη από απομονωμένα άτομα από κάθε μια από αυτές τις περιοχές ανεξάρτητα. Κυρίως, οι θεμελιώδεις ιδέες της οντολογίας συστημάτων πληροφορικής αναπτύχθηκαν από φιλοσόφους που ξεκίνησαν κυριολεκτικά εκ του μηδενός. Ο John McCarthy αναγνώρισε αρχικά την επικάλυψη μεταξύ της εργασίας στη φιλοσοφική οντολογία και της δραστηριότητας οικοδόμησης των λογικών θεωριών των συστημάτων τεχνητής νοημοσύνης. Ο McCarthy επιβεβαίωσε ήδη το 1980 ότι οι σχεδιαστές των βασισμένων στη λογική ευφυών συστημάτων πρέπει πρώτα "να απαριθμήσουν όλα αυτά που υπάρχουν, χτίζοντας μια οντολογία του κόσμου μας"[McCarthy, 1980]. Αυτή η άποψη, που εμπνεύστηκε ο McCarthy έχοντας επηρεαστεί σε κάποιο βαθμό από τον Quine[Quine, 1969], υποστηρίχτηκε στη συνέχεια και από τον Hayes[Hayes, 1985].

Οι περισσότερες προσπάθειες της τεχνητής νοημοσύνης εστίασαν στη σύλληψη πληροφοριών για τον κόσμο που είναι συμβατές με την προοπτική της ανθρώπινης κοινής λογικής, και αυτές οι προσπάθειες συνδέθηκαν πολύ με την έρευνα για το θέμα του συλλογισμού κοινής λογικής[Davis, 1990]. Μια παρόμοια προοπτική, αλλά με ευρύτερες φιλοδοξίες και με μια ακόμα προφανέστερη αναγνώριση της επικάλυψης με τη φιλοσοφία προτάθηκε από John Sowa[Sowa, 1984], ο οποίος αναφέρεται σε «μια οντολογία για έναν πιθανό κόσμο, δηλαδή έναν κατάλογο όλων όσων αποτελούν αυτόν τον κόσμο, πώς αλληλοεξαρτώνται και πώς συνδυάζονται».

Παρά την ώθηση από αυτές τις προσωπικότητες, το μεγαλύτερο μέρος της τεχνητής νοημοσύνης επέλεξε να μην εξετάσει τα αποτελέσματα του κατά πολύ παλαιότερου τομέα της φιλοσοφικής οντολογίας, προτιμώντας αντί αυτού να χρησιμοποιήσει τον όρο οντολογία ως ένα εξωτικό όνομα αυτού ακριβώς που έκαναν ούτως ή άλλως – της μηχανικής γνώσης. Αυτό οδήγησε δυστυχώς σε

μια διαστροφή της έννοιας του όρου, όπως χρησιμοποιείται στους τομείς τεχνητής νοημοσύνης και συστημάτων πληροφοριών, καθώς η εργασία υπό τον τίτλο της οντολογίας στην πράξη παρουσιάστηκε πιο κοντά στη λογική θεωρία, και ειδικά στη λογική σημασιολογία, και αποδείχτηκε αντίστοιχα πιο απομακρυσμένη από οτιδήποτε έχει άμεση σχέση με την ύπαρξη ή την πραγματικότητα και την περιγραφή τους. Κάποιοι μπορούν να υποστηρίξουν ότι αυτή η έννοια είναι πιο κατάλληλη για ένα υπολογιστικό σύστημα, δεδομένου ότι μια λογικο-σημασιολογική θεωρία θα καθορίσει, στην πράξη, τα είδη και τις δομές των αντικειμένων, των ιδιοτήτων, των γεγονότων, των διαδικασιών και των σχέσεων που υπάρχουν στο σύστημα[Smith, 2001].

Αφ' ετέρου, πολλοί πλέον υποστηρίζουν ότι η ίδια η έλλειψη θεμελίωσης στην εξωτερική πραγματικότητα είναι ακριβώς αυτό που δημιούργησε το πρόβλημα, που τόσο πιέζει τη βιομηχανία πληροφορίας σήμερα, της ολοκλήρωσης legacy systems. Πώς μπορούμε να κάνουμε παλαιότερα συστήματα με διαφορετικά εννοιολογικά πρότυπα αλλά επικαλυπτόμενη σημασιολογία να εργαστούν μαζί, εάν όχι με την αναφορά στον κοινό κόσμο στον οποίο όλα αναφέρονται; Από τις πρώτες χρήσεις του όρου οντολογία υπό τη νέα έννοια του χώρου ΑΙ, όπως παραδείγματος χάριν στο[Alexander, 1986], η πολλαπλότητα ερμηνειών του όρου αυξήθηκε, και καθώς οι απομονωμένοι τομείς της μηχανικής γνώσης, της εννοιολογικής μοντελοποίησης, και του domain modelling άρχισαν να συγκλίνουν και να ανακαλύπτουν ο ένας τον άλλο, αντίστοιχα μεγάλωσε ο αριθμός των παραλλαγών της έννοιας του όρου οντολογία. Μέχρι το 1993 η χρήση του όρου ήταν ήδη αρκετά διαδεδομένη σε κάθε ένα από αυτά τα πεδία της πληροφορικής.

2.1.2 Προσπάθειες Καθιέρωσης Ορολογίας

Αν και ο Tom Gruber πιστώνεται συχνά με την έναρξη της χρήσης του όρου σε στους διάφορους τομείς της πληροφορικής, η συμβολή του το 1993 ήταν πραγματικά αυτή της πρώτης αξιόπιστης προσπάθειας για τον ορισμό του όρου

οντολογία[Gruber, 1993]. Αλλά ο προτεινόμενος ορισμός του, «μια οντολογία είναι μια προδιαγραφή μιας σύλληψης» (an ontology is a specification of a conceptualization), αφήνει περιθώριο για πάρα πολλές πιθανές ερμηνείες, και παρά μια προσπάθεια να διευκρινιστεί και να τυποποιηθεί ο ορισμός περαιτέρω από τον Guarino[Guarino, 1998], οι νέες σημασίες του όρου οντολογία συνέχισαν να πολλαπλασιάζονται. Οι Welty, Lehmann, Gruninger και Uschold αναφέρθηκαν το 1999 σε ένα ευρύ φάσμα πληροφοριακών αντικειμένων που σε κάποια χρονική στιγμή καταχωρήθηκαν ως οντολογίες[Welty, 1999]. Στο Σχήμα 2 φαίνεται μια γραφική απεικόνιση των αποτελεσμάτων τους.

Κατάλογος	Συλλογή αρχείων	Γλωσσάριο	Θησαυρός	Ταξινόμια	Σύνολο πλαισίων (frames)	Σύνολο γενικών λογικών περιορισμών
Χωρίς δυνατότητα reasoning				Με δυνατότητα reasoning		
Μικρή	ΠΟΛΥΠΛΟΚΟΤΗΤΑ			Μεγάλη		

Σχήμα 2: Ταξινόμηση διαφορετικών ερμηνειών του όρου οντολογία, ανάλογα με την πολυπλοκότητα

Συστήματα πληροφοριών όπως απλοί κατάλογοι, στους οποίους κάθε τύπος προϊόντων έχει έναν μοναδικό κώδικα (π.χ. ο αριθμός στοιχείου), έχουν συχνά επονομαστεί οντολογίες. Ένας κατάλογος είναι, από μία άποψη, η οντολογία των πραγμάτων που μια επιχείρηση εμπορεύεται. Ένα ελαφρώς πιο σύνθετο σύστημα πληροφοριών μπορεί να παρέχει απλά ονόματα φυσικής γλώσσας και να επιτρέπει την αντιστοίχιση ονομάτων. Τα γλωσσάρια (glossaries ή lexicons) είναι τέτοια συστήματα πληροφοριών που παρέχουν τις περιγραφές φυσικής γλώσσας των όρων, συχνά μαζί με γραμματικές ή συντακτικές πληροφορίες, επιβάλλοντας κατά συνέπεια κάποια δομή σε κείμενα. Βρίσκουν εφαρμογή σε μια μεγάλη ποικιλία συστημάτων επεξεργασίας φυσικής γλώσσας με ανάγκες

μορφολογικής ανάλυσης, για παράδειγμα συστήματα μετατροπής κειμένου σε ομιλία [Stavroulos , 2000]. Οι θησαυροί είναι τυποποιημένα συστήματα πληροφοριών που παρέχουν, εκτός από τις περιγραφές των όρων, επίσης τις σχέσεις με άλλους γενικότερους ή ειδικότερους όρους, στα πλαίσια μιας κοινής ιεραρχίας όρων.

Οι τομείς της αναπαράστασης γνώσης (knowledge representation), της ανάπτυξης βάσεων δεδομένων, και της αντικειμενοστραφούς τεχνολογίας λογισμικού χρησιμοποιούν οντολογίες που γίνονται αντιληπτές ως ταξινομίες στις οποίες οι ιδιότητες των γενικότερων κατηγοριών κληρονομούνται από τις πιο ειδικές. Τα βασισμένα σε πλαίσια συστήματα (frame-based systems) παρέχουν, εκτός από την ταξινομική δομή, τις σχέσεις μεταξύ των αντικειμένων και τους περιορισμούς στο ποιες κατηγορίες αντικειμένων και πώς μπορούν να σχετίζονται μεταξύ τους. Τέλος, οι πιο εκφραστικές και σύνθετες οντολογίες συστημάτων πληροφοριών χρησιμοποιούν τα αξιώματα της πλήρους πρώτης τάξεως, ή και υψηλότερων τάξεων. Όλοι αυτοί οι τύποι συστημάτων πληροφοριών ικανοποιούν τον ορισμό του Gruber, και βρίσκονται πλέον όλα είναι κάτω από την κοινή ομπρέλα του όρου οντολογία.

Από αυτό το ορολογικό χάος, κάποια τάξη αρχίζει να αναδύεται τα τελευταία χρόνια. Βαθμιαία, οι επιστήμονες υπολογιστών αρχίζουν να αναγνωρίζουν ότι η καθιέρωση, μια για πάντα, μιας κοινής, συνεπούς οντολογίας αναφοράς – δηλαδή μιας κοινής ταξινομίας των οντοτήτων – παρέχει σημαντικά πλεονεκτήματα σε σύγκριση με τις ειδικές, κατά περίπτωση μεθόδους που χρησιμοποιούνται μέχρι τώρα. Τέλος, όλο και περισσότεροι επιστήμονες και μηχανικοί πληροφορικής συνειδητοποιούν ότι ίσως η φιλοσοφία έχει κάτι χρήσιμο προσφέρει στα αντικείμενα της εργασίας τους.

Η άνοδος της οντολογίας στην πληροφορική απεικονίζει μια νίκη του περιεχομένου πάνω στη διαδικασία, μια νίκη που, παράδοξα, έχει ενισχυθεί ως αποτέλεσμα του γεγονότος ότι, δεδομένου ότι το λογισμικό το ίδιο γίνεται

διαρκώς περιπλοκότερο, οι υπεύθυνοι για την ανάπτυξη λογισμικού και οι θεωρητικοί υπολογιστών βρίσκουν ολοένα ευκολότερο να εστιάσουν στα στοιχεία πάνω στα οποία τα συστήματά τους λειτουργούν, παρά στη λειτουργία και τις διαδικαστικές πτυχές των συστημάτων. Η σημασία αυτής της αλλαγής αποτυπώθηκε σε μια εύγλωττη παρατήρηση του Dijkstra το 1986, ο οποίος έγραψε στο *The Mathematical Intelligencer* ότι το να ονομάζουμε αυτό που κάνουν οι επιστήμονες υπολογιστών «επιστήμη υπολογιστών» είναι σαν να ονομάζουμε αυτό που κάνουν οι χειρουργοί «επιστήμη του νυστεριού»! Κι αυτό γιατί ο όρος επιστήμη υπολογιστών εστιάζει στενά στο εργαλείο, παρά σε αυτό που το εργαλείο κάνει ή στα αντικείμενα, τις σχέσεις, τις διαδικασίες και, πάνω από όλα, τους σκοπούς που το εργαλείο σχεδιάζεται για να επιτελέσει. Παρατηρούμε επιπλέον ότι, καθώς τα συστήματα πληροφοριών εισέρχονται ολοένα και περισσότερο σε περισσότερες περιοχές και διαστάσεις της ζωής μας, η περιοχή που πρέπει να καλυφθεί από τον όρο «επιστήμη υπολογιστών» γίνεται διαρκώς μεγαλύτερη – μέχρι το σημείο που ο όρος αυτός πιθανώς να πρέπει πλέον να απορριφθεί ως ακυριολεξία.

Η άνοδος της οντολογίας μπορεί να ειπωθεί υπό αυτό το πρίσμα, ως το αποτέλεσμα των προσπαθειών μιας μερίδας επιστημόνων υπολογιστών και πληροφορικής, με σκοπό να ξεπεράσουν τα τεχνητά αντικείμενα της πληροφορίας και του υπολογισμού, και να εστιάσουν στον πραγματικό κόσμο από τον οποίο αυτά τα αντικείμενα πηγάζουν. Η εργασία των Guarino και Welty [Guarino, 2000], για παράδειγμα, εισήγαγε την φιλοσοφική αντιμετώπιση των εννοιών της ταυτότητας και της ενότητας ως τυποποιημένα εργαλεία για την ανάλυση των οντολογικών διεργασιών. Η εργασία τους, μαζί με άλλες, είναι βασισμένη στην ιδέα ότι το πρόγραμμα της ανάπτυξης μιας σταθερής κοινής οντολογίας αναφοράς μπορεί να ωφεληθεί από τις θεωρίες που αναπτύχθηκαν από φιλοσόφους σε πάνω από 2000 έτη οντολογικής έρευνας. Ένα βασικό χαρακτηριστικό των συστημάτων που προκύπτουν ως αποτέλεσμα τέτοιων

προσπαθειών είναι ότι είναι πολύ απλούστερα στη συντήρηση και απολαμβάνουν το πλεονέκτημα της διαλειτουργικότητας (interoperability). Ήδη οι επιτυχίες αυτής της προσέγγισης φαίνονται σε επιχειρήσεις και προγράμματα που ασχολούνται με σημασιολογικές εφαρμογές.

Αυτό δεν σημαίνει, φυσικά, ότι η φιλοσοφία δεν έχει τίποτα να κερδίσει από την θεώρηση της οντολογίας από την πλευρά του επιστήμονα υπολογιστών. Οι πρόσφατες εξελίξεις σε διάφορα πεδία της λογικής (modal, temporal logic) όπως επίσης και σε άλλες συγγενείς περιοχές, έχουν καταδείξει το βαθμό στον οποίο οι πρόοδοι στην πληροφορική μπορούν να προάγουν την έρευνα και να δώσουν οφέλη στον τομέα της λογικής – οφέλη όχι μόνο αυστηρά τεχνικής φύσης, αλλά και ευρύτερης φιλοσοφικής σημασίας. Η εμπειρία των επιτυχιών και των αποτυχιών της οντολογίας στη λύση προβλημάτων από τον τομέα των συστημάτων πληροφορικής μπορεί αφενός να υποβοηθήσει τις υπάρχουσες τάσεις στη φιλοσοφική οντολογία ως προς το άνοιγμα νέων περιοχών έρευνας για τους φιλοσόφους, παραδείγματος χάριν η περιοχή των προτύπων (patterns) [Johansson, 1989], της εξάρτησης [Mertz, 1996] κλπ. Αφετέρου, μπορεί να ρίξει νέο φως σε πολλές υπάρχουσες συνεισφορές στην οντολογία, των οποίων η σημασία παραμελήθηκε για πολύ από τους φιλοσόφους που αναπτύχθηκαν στη σκιά του Kant και άλλων πολεμίων της μεταφυσικής. Επιπλέον, εάν η φιλοσοφική οντολογία επιχειρηθεί να γίνει αντιληπτή ως ένα είδος γενικευμένης χημείας, τότε τα συστήματα πληροφοριών μπορούν να βοηθήσουν να καλυφθεί ένα σημαντικό κενό στην οντολογία όπως αυτή έχει διαμορφωθεί μέχρι τώρα, το οποίο εντοπίζεται στην έλλειψη οποιουδήποτε ανάλογου του χημικού πειραματισμού.

Τέλος, τα μαθήματα που διδασκόμαστε από την οντολογία συστημάτων πληροφοριών μπορούν να υποστηρίξουν τις προσπάθειες των φιλοσόφων που ασχολούνται όχι μόνο με την ανάπτυξη των οντολογικών θεωριών, αλλά και με την εφαρμογή τέτοιων θεωριών σε περιοχές όπως η νομική [Koepsell, 2000],

ή το εμπόριο [Grassl, 1999], η ιατρική, η γεωγραφία [Navratil, 1998] και άλλες, δημιουργώντας ένα πεδίο που ονομάζεται εφαρμοσμένη οντολογία. Τα εργαλεία της φιλοσοφικής οντολογίας έχουν εφαρμοστεί για να λύσουν πρακτικά προβλήματα, παραδείγματος χάριν σχετικά με τη φύση της πνευματικής ιδιοκτησίας ή σχετικά με την ταξινόμηση του ανθρώπινου εμβρύου στα διαφορετικά στάδια ανάπτυξής του. Η συνεργασία και η υιοθέτηση ιδεών από το πεδίο της οντολογίας συστημάτων πληροφορικής μπορεί να υποβοηθήσει τέτοιες προσπάθειες με διάφορους τρόπους, πρωτίστως επειδή τα αποτελέσματα σε συγκεκριμένους τομείς εφαρμογής μπορούν να λειτουργήσουν ως ερεθίσματα για τους φιλοσόφους. Ακόμα σημαντικότερα, τα πληροφοριακά συστήματα αποτελούν από μόνα τους ένα γιγάντιο τομέα πρακτικής εφαρμογής της οντολογίας, ο οποίος προσφέρεται ιδιαίτερα για διερεύνηση με φιλοσοφικές μεθόδους.

2.2 ΟΝΤΟΛΟΓΙΕΣ ΓΙΑ ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ

Σε μερικές περιπτώσεις, ο όρος «οντολογία» είναι απλώς ένα εξωτικό όνομα που αντιπροσωπεύει το αποτέλεσμα καλά γνωστών δραστηριοτήτων όπως η εννοιολογική ανάλυση και η μοντελοποίηση περιοχών, που πραγματοποιούνται με τη βοήθεια τυποποιημένων μεθοδολογιών. Σε πολλές περιπτώσεις, εντούτοις, οι αποκαλούμενες οντολογίες παρουσιάζουν μεθοδολογικές και αρχιτεκτονικές ιδιαιτερότητες. Στη μεθοδολογική πλευρά, η κύρια ιδιαιτερότητα είναι η υιοθέτηση μιας κατ' εξοχήν διεπιστημονικής προσέγγισης, όπου η φιλοσοφία και η γλωσσολογία διαδραματίζουν έναν θεμελιώδη ρόλο στην ανάλυση της δομής μιας υφιστάμενης οντότητας σε μεγάλο βαθμό γενικότητας, και στη διατύπωση ενός σαφούς και αυστηρού λεξιλογίου.

Στην αρχιτεκτονική πλευρά, η πιο ενδιαφέρουσα πτυχή είναι η θεμελιώδης σημασία του ρόλου που μια οντολογία μπορεί να διαδραματίσει σε ένα σύστημα πληροφοριών, που οδηγεί στην προοπτική συστημάτων πληροφοριών

βασισμένων σε οντολογίες. Έχει τονιστεί στη βιβλιογραφία [Gruninger, 1995], [Guarino, 1995], η σημασία της διεπιστημονικής προσέγγισης στην εξάσκηση της οντολογικής μηχανικής, ειδικότερα το ρόλο που διαδραματίζει η κλασική φιλοσοφική οντολογία. Ωστόσο ένα ανεπιθύμητο επακόλουθο αυτής της διεπιστημονικής εξελικτικής διαδικασίας, είναι η δημιουργία μεγάλης σύγχυσης και ανομοιογένειας στη σχετική ορολογία.

Παρά τη σπουδαιότητα που χαρακτηρίζει το θέμα αυτό, υπάρχει ακόμα έντονη ορολογική και εννοιολογική σύγχυση. Στην παρούσα εργασία υιοθετούμε την ορολογία που χρησιμοποιεί η πλειοψηφία της διεθνούς επιστημονικής κοινότητας τεχνητής νοημοσύνης. Στη συνέχεια δίνονται μερικοί ορισμοί βασικών εννοιών που χρησιμοποιούνται εδώ, και εξηγούνται τα νοήματα και οι διαφορές τους.

Πρώτα από όλα, ο όρος «οντολογία» χρησιμοποιείται τόσο στον ενικό ως κύριο όνομα (Οντολογία), όσο και σε ενικό ή πληθυντικό ως απλό ουσιαστικό (π.χ. μια οντολογία περιοχής, η οντολογία Χ κλπ). Ενώ η πρώτη περίπτωση είναι αρκετά σαφής, αναφερόμενη σε μια ιδιαίτερη περιοχή φιλοσοφικής μελέτης, δύο διαφορετικές ερμηνείες δίνονται στη δεύτερη περίπτωση, από τη φιλοσοφική κοινότητα και την κοινότητα τεχνητής νοημοσύνης (και γενικότερα ολόκληρη την κοινότητα πληροφορικής) αντίστοιχα. Υπό τη φιλοσοφική έννοια, μπορούμε να αναφερθούμε σε μια οντολογία ως ένα ιδιαίτερο σύστημα κατηγοριών που αποδίδουν μια συγκεκριμένη σύλληψη του κόσμου. Υπό αυτήν τη μορφή, αυτό το σύστημα δεν εξαρτάται από μια ιδιαίτερη γλώσσα ή περιγραφική μέθοδο. Η Οντολογία του Αριστοτέλη είναι πάντα η ίδια, ανεξάρτητα από τη γλώσσα που χρησιμοποιείται για να σε περιγράψει.

Ωστόσο, στην πιο επικρατούσα χρήση της στην τεχνητή νοημοσύνη, μια οντολογία αναφέρεται σε ένα τεχνητό αντικείμενο, αποτελούμενο από ένα συγκεκριμένο λεξιλόγιο που χρησιμοποιείται για να περιγράψει μια ορισμένη πραγματικότητα, και ένα σύνολο ρητών υποθέσεων σχετικά με την επιθυμητή

έννοια των λέξεων που απαρτίζουν αυτό το λεξιλόγιο. Αυτό το σύνολο υποθέσεων έχει συνήθως τη μορφή θεωρίας λογικής πρώτης τάξεως, όπου οι λέξεις του λεξιλογίου εμφανίζονται ως μοναδιαία ή δυαδικά predicates, αποκαλούμενα αντίστοιχα έννοιες και σχέσεις. Στην απλούστερη περίπτωση, μια οντολογία περιγράφει μια ιεραρχία εννοιών που ορίζεται με σχέσεις υπαγωγής (subsumption). Σε περιπλοκότερες περιπτώσεις, κατάλληλα αξιώματα προστίθενται προκειμένου να εκφραστούν άλλες σχέσεις μεταξύ των εννοιών και να περιορίσουν την προοριζόμενη ερμηνεία τους.

Οι δύο ερμηνείες του όρου «οντολογία» που περιγράφονται παραπάνω συσχετίζονται προφανώς μεταξύ τους. Ωστόσο προκειμένου να λυθεί το ορολογικό αδιέξοδο θα χρησιμοποιήσουμε την ερμηνεία της τεχνητής νοημοσύνης. Στη συνέχεια, θα αναφερόμαστε στην ερμηνεία της φιλοσοφικής κατεύθυνσης χρησιμοποιώντας τον όρο «φιλοσοφική οντολογία» (philosophical ontology). Στη διεθνή βιβλιογραφία χρησιμοποιείται για τον ίδιο σκοπό και ο όρος conceptualisation, που μεταφράζεται στο παρόν σαν αντίληψη.

2.2.1 Ορισμός της Οντολογίας

Μπορούμε τώρα να διευκρινίσουμε το ρόλο μιας οντολογίας, που θεωρείται ως σύνολο λογικών αξιωμάτων με σκοπό να αποτελέσουν την επιθυμητή έννοια ενός λεξιλογίου. Ο διαχωρισμός μεταξύ οντολογίας και φιλοσοφικής οντολογίας παρέχει τη δυνατότητα για ένα συμπαγέστερο ορισμό του όρου οντολογία, σε σχέση με τον ορισμό του Gruber ("an ontology is a specification of a conceptualisation") που είναι ο συνηθέστερα χρησιμοποιούμενος. Πράγματι ένας τέτοιος ορισμός έχει δοθεί από τον Guarino [Guarino, 1998]. Σύμφωνα με αυτόν, μια οντολογία είναι μια λογική θεωρία που απεικονίζει την επιθυμητή έννοια ενός formal λεξιλογίου, εκφράζοντας έτσι την συγκεκριμένη αντίληψη του κόσμου που δίνεται από το λεξιλόγιο αυτό.

Η βασική διαφορά έναντι του ορισμού του Gruber βρίσκεται στο γεγονός ότι ο ορισμός του Guarino δηλώνει ρητά ότι μια οντολογία μόνο έμμεσα εκφράζει μια αντίληψη του κόσμου (δηλ. conceptualisation). Ο ορισμός αυτός είναι πρακτικά πολύ χρησιμότερος, διότι δίνει έμφαση στη σημασιολογική χρήση των οντολογιών, στην οποία βέβαια βασίζεται η πρακτική τους εκμετάλλευση στο χώρο των υπολογιστικών και πληροφοριακών συστημάτων. Σύμφωνα με τον Guarino η οντολογία εκφράζει μια αντίληψη του λεξιλογίου. Η έκφραση της αντίληψης του κόσμου είναι μια ιδιότητα που την αποκτά επαγωγικά: το λεξιλόγιο περικλείει μια τέτοια αντίληψη και, δεδομένου ότι η οντολογία αναπαριστά τη σημασιολογία του λεξιλογίου, έπεται ότι κωδικοποιεί και την αντίληψη του κόσμου του λεξιλογίου. Επομένως, το λεξιλόγιο μεσολαβεί ανάμεσα στην οντολογία και τον κόσμο και η περιγραφή του κόσμου από την οντολογία περιορίζεται από τον ορισμό του λεξιλογίου.

Δεδομένου λοιπόν ότι μια οντολογία μόνο έμμεσα εκφράζει μια αντίληψη του κόσμου, έχει νόημα να αναρωτηθούμε πόσο η οντολογία μπορεί να πλησιάσει την αντίληψη αυτή. Στη συνέχεια, μπορούμε να ταξινομήσουμε τις οντολογίες σύμφωνα με την ακρίβειά τους στο χαρακτηρισμό της αντίληψης από όπου πηγάζουν. Υπάρχουν δύο πιθανές μέθοδοι με τις οποίες μια οντολογία μπορεί να συγκλίνει στην αντίστοιχη αντίληψη του κόσμου: Είτε με την ανάπτυξη πλουσιότερου πυρήνα αξιωμάτων, είτε με την υιοθέτηση μιας πλουσιότερης περιοχής ή ενός πλουσιότερου συνόλου αντίστοιχων εννοιολογικών σχέσεων. Στην πρώτη περίπτωση, η διαφορά μεταξύ του συνόλου των οντολογικών μοντέλων και του συνόλου των πραγματικών μοντέλων μειώνεται.

Στη δεύτερη περίπτωση, είναι δυνατό, τουλάχιστον στις περισσότερες περιπτώσεις, να περιληφθούν στο σύνολο των εννοιολογικών σχέσεων κάποιες από εκείνες τις σχέσεις που χαρακτηρίζουν κάθε υφιστάμενη κατάσταση του κόσμου, επεκτείνοντας συγχρόνως την περιοχή που καλύπτει η οντολογία προκειμένου να περιληφθούν οι οντότητες που εισέρχονται σε αυτές τις σχέσεις.

Δεδομένου ότι κάθε οντολογικό μοντέλο περιέχει τώρα τις πληροφορίες σχετικά με την κατάσταση του κόσμου όπου αναφέρεται, η αντίστοιχη αντίληψη του κόσμου μπορεί να ανοικοδομηθεί από το σύνολο των μοντέλων αυτών. Σε αυτήν την περίπτωση, εάν μια οντολογία μπορεί να κατασκευαστεί με τέτοιο τρόπο ώστε τα μοντέλα που εμπεριέχει να είναι πλήρως ακριβή, μπορούμε να πούμε ότι είναι μια τέλεια οντολογία.

Σε αντίθεση με τις λεπτομερείς οντολογίες, μια γενική και αφηρημένη οντολογία μπορεί να κατασκευασθεί από ένα ελάχιστο σύνολο αξιωμάτων εκφρασμένων σε κάποια γλώσσα χαμηλής εκφραστικότητας, με σκοπό να υποστηρίξουν μόνο ένα περιορισμένο σύνολο συγκεκριμένων υπηρεσιών, που προορίζεται για κοινή χρήση μεταξύ χρηστών που ήδη έχουν διαμορφώσει και συμφωνήσει σε μια κοινή αντίληψη του κόσμου. Μπορούμε να διακρίνουμε επομένως μεταξύ των λεπτομερών οντολογιών αναφοράς και των αφηρημένων κοινών οντολογιών, ως εξής: οι πρώτες χρησιμοποιούνται σπάνια και μόνο για λόγους αναφοράς, ενώ οι δεύτερες υποστηρίζουν τις θεμελιώδεις λειτουργίες του συστήματος.

2.2.2 Ολοκλήρωση Οντολογιών

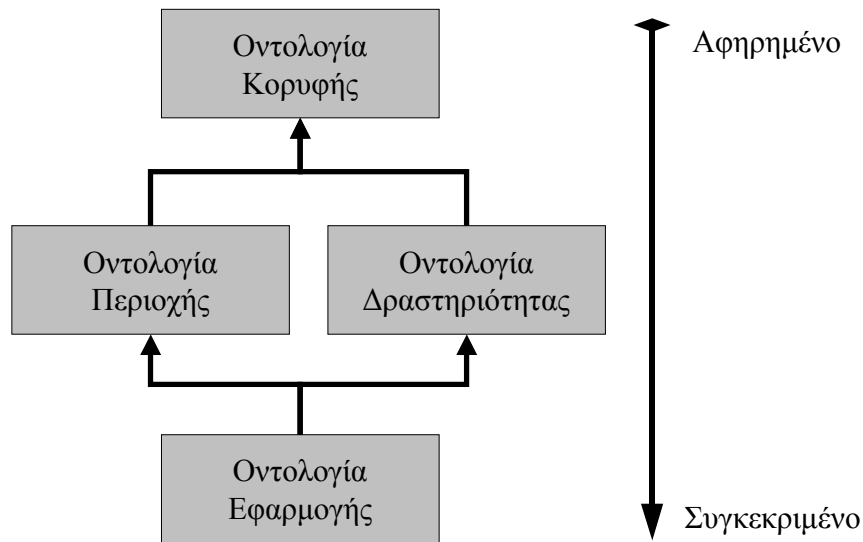
Όπως έχει ήδη αναφερθεί, η ολοκλήρωση πληροφοριών είναι ένας από τους σημαντικότερους τομείς εφαρμογής των οντολογιών. Όπως είναι γνωστό, ακόμα κι αν δύο συστήματα υιοθετούν το ίδιο λεξιλόγιο, δεν υπάρχει καμία εγγύηση ότι μπορούν να συμφωνήσουν στην ορθότητα ή μη οποιασδήποτε πρότασης, εκτός αν μοιράζονται την ίδια αντίληψη του κόσμου. Υποθέτοντας ότι κάθε σύστημα έχει τη δική του αντίληψη, ένας αναγκαίος όρος προκειμένου να κατασταθεί η συμφωνία πιθανή, είναι ότι τα κοσμικά μοντέλα των δύο διαφορετικών αντιλήψεων έχουν κάποια επικάλυψη.

Υποθέτοντας τώρα ότι αυτά τα δύο σύνολα μοντέλων προσεγγίζονται από δύο διαφορετικές οντολογίες, είναι δυνατό να έχουμε το φαινόμενο οι δύο

οντολογίες να επικαλύπτονται, ενώ τα αρχικά μοντέλα όχι. Το φαινόμενο εμφανίζεται στην απλή περίπτωση που τουλάχιστον μία από τις δύο οντολογίες παρουσιάζει υπερπαραγωγή μοντέλων σε σχέση με την αντίστοιχη της αντίληψη του κόσμου. Κάτι τέτοιο είναι δυνατό, αφού στον ορισμό της οντολογίας χρησιμοποιείται μια προσέγγιση από πάνω προς τα κάτω: ζητείται από μια οντολογία να περιγράψει σωστά ένα δεδομένο λεξιλόγιο, αν όμως η οντολογία χρησιμοποιηθεί αντίστροφα για να παράγει ένα λεξιλόγιο, ο ορισμός δεν δίνει εγγύηση για τη μη εμφάνιση υπερπαραγωγής.

Αυτό σημαίνει ότι μια από κάτω προς τα επάνω προσέγγιση στην ολοκλήρωση συστημάτων βασισμένη στην ολοκλήρωση πολλαπλών επιμέρους οντολογιών μπορεί να μην λειτουργήσει, ειδικά εάν οι επιμέρους οντολογίες στρέφονται μόνο στις εννοιολογικές σχέσεις σχετικές με ένα συγκεκριμένο θέμα, και επομένως μόνο κάτω από πολύ ειδικές συνθήκες προσεγγίζουν τα πραγματικά μοντέλα. Ως εκ τούτου, φαίνεται ευκολότερο να συμφωνήσουν όλα τα συστήματα σε μία ενιαία κεντρική οντολογία, παρά να στηριχθούν σε επιμέρους συμφωνίες βασισμένες στην τομή των διαφορετικών οντολογιών.

Οι παραπάνω παρατηρήσεις ανοίγουν το δρόμο για την ανάπτυξη διαφορετικών ειδών οντολογιών, ανάλογα με το επίπεδο αφαίρεσης ή λεπτομέρειας που τις χαρακτηρίζει (Σχήμα 3).



Σχήμα 3: Είδη οντολογιών ανάλογα με τη γενικότητα, τη λεπτομέρεια και την εξάρτηση από συγκεκριμένες εφαρμογές [Guarino, 1997]

Οι οντολογίες κορυφής περιγράφουν πολύ γενικές έννοιες όπως το διάστημα, το χρόνο, το αντικείμενο, το γεγονός, τη δράση κλπ, οι οποίες είναι ανεξάρτητες από ένα ιδιαίτερη πρόβλημα ή περιοχή: φαίνεται επομένως λογικό, τουλάχιστον θεωρητικά, να χρησιμοποιούνται ενοποιημένες οντολογίες κορυφής για μεγάλο εύρος εφαρμογών και χρηστών. Οι οντολογίες περιοχών και οι οντολογίες δραστηριότητας περιγράφουν, αντίστοιχα, το λεξιλόγιο το σχετικό με μια γενική περιοχή (όπως την ιατρική) ή μια συγκεκριμένη δραστηριότητα ή εργασία, εξειδικεύοντας τους όρους που περιγράφονται στην κορυφαία οντολογία. Οι οντολογίες εφαρμογών περιγράφουν έννοιες που εξαρτώνται και από μια συγκεκριμένη περιοχή και από συγκεκριμένη δραστηριότητα. Αυτές είναι συχνά ειδικεύσεις και των δύο αντίστοιχων οντολογιών περιοχής και δραστηριότητας. Οι έννοιες που περιέχουν αντιστοιχούν συχνά στους ρόλους που διαδραματίζονται από τις οντότητες περιοχών όταν εμπλέκονται σε μια ορισμένη δραστηριότητα.

Τέλος, είναι πιθανώς σκόπιμο να τονιστεί η διαφορά μεταξύ μιας οντολογίας εφαρμογής και μιας βάσης γνώσεων (knowledge base). Η διαφορά σχετίζεται με το σκοπό μιας οντολογίας, η οποία είναι μια πολύ ιδιαίτερη βάση γνώσεων που περιέχει προτάσεις που υποτίθεται ότι είναι πάντα αληθείς για μια συγκεκριμένη ομάδα χρηστών, βάσει της προσυμφωνημένης εννοιολογικής ερμηνείας ενός κοινού λεξιλογίου. Σε αντίθεση, μια γενική βάση γνώσεων μπορεί να είναι πολύ πιο εκτεταμένη ώστε να περιγράψει επίσης προτάσεις και αξιώματα σχετικά με μια συγκεκριμένη κατάσταση του κόσμου. Μέσα σε μια γενική βάση γνώσεων, μπορούμε να διακρίνουμε επομένως δύο συστατικά: την οντολογία, που περιέχει πληροφορίες ανεξάρτητες από την κατάσταση του κόσμου, και την κεντρική βάση γνώσεως που περιέχει τις πληροφορίες που εξαρτώνται από την κατάσταση.

2.3 Η ΕΠΙΔΡΑΣΗ ΤΗΣ ΟΝΤΟΛΟΓΙΑΣ ΣΤΗ ΣΧΕΔΙΑΣΗ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Σε κάθε σύστημα πληροφοριών (IS) μπορεί να αντιστοιχισθεί μια οντολογία, δεδομένου ότι το σύστημα αποδίδει συγκεκριμένη έννοια σε σύμβολα που χρησιμοποιούνται σύμφωνα με μια ιδιαίτερη αντίληψη του κόσμου. Αυτό όμως που έχει πρακτικό ενδιαφέρον, είναι η διερεύνηση του ρόλου που μπορεί να παίξει μια ρητά διατυπωμένη οντολογία. Μια τέτοια οντολογία μπορεί να δομηθεί χρησιμοποιώντας μια αρκετά εκφραστική γλώσσα επομένως είναι χρήσιμο να δούμε κατά πόσο η εισαγωγή της σε ένα πληροφοριακό σύστημα μπορεί να αποφέρει πλεονεκτήματα. Οι ερευνητές της τεχνητής νοημοσύνης τάσσονται υπέρ μιας αρχιτεκτονικής προοπτικής όπου αυτός ο ρόλος είναι κεντρικός, και η οντολογία «οδηγεί» όλες τις πτυχές και όλα τα συστατικά του πληροφοριακού συστήματος, έτσι ώστε μπορούμε να μιλήσουμε για συστήματα πληροφοριών πραγματικά βασισμένα σε οντολογίες.

Ένα πληροφοριακό σύστημα γενικά αποτελείται από συστατικά τριών διαφορετικών ειδών: προγράμματα εφαρμογών, πηγές πληροφοριών όπως βάσεις δεδομένων ή βάσεις γνώσεων, και διαπροσωπείες με τον χρήστη. Αυτά τα συστατικά είναι διασυνδεδεμένα με τέτοιο τρόπο ώστε να επιτευχθεί ένας συγκεκριμένος επιχειρησιακός σκοπός. Μελετώντας τον αντίκτυπο που μια οντολογία μπορεί να έχει στο πληροφοριακό σύστημα, μπορούμε να διακρίνουμε δύο ορθογώνιες διαστάσεις. Μια χρονική διάσταση, η οποία έχει να κάνει με το πότε η οντολογία χρησιμοποιείται, σε χρόνο ανάπτυξης (δηλ. για την παραγωγή του πληροφοριακού συστήματος) ή σε χρόνο εκτέλεσης (δηλ. για την λειτουργία του). Επίσης, μια δομική διάσταση, η οποία σχετίζεται με τον συγκεκριμένο τρόπο που μια τέτοια οντολογία μπορεί να αλληλεπιδράσει με τα συστατικά στοιχεία του πληροφοριακού συστήματος.

Πριν εστιάσουμε την επίδραση της οντολογίας σε συγκεκριμένα συστατικά του πληροφοριακού συστήματος, θα αναλύσουμε το γενικότερο ρόλο που μπορεί η οντολογία να διαδραματίσει στο χρόνο ανάπτυξης και στο χρόνο εκτέλεσης ενός συστήματος. Πρέπει να σημειωθεί ότι αυτοί οι χρόνοι αφορούν τον κύκλο ζωής του πληροφοριακού συστήματος του ίδιου, κι όχι της οντολογίας, η οποία υποθέτουμε ότι σε κάθε περίπτωση είναι ανεπτυγμένη και ολοκληρωμένη. Ωστόσο, στο χρόνο ανάπτυξης η «έτοιμη» οντολογία θα πρέπει να προσαρμοστεί στις ανάγκες του συγκεκριμένου πληροφοριακού συστήματος. Όταν η οντολογία χρησιμοποιείται από πληροφοριακό σύστημα στο χρόνο εκτέλεσης, μπορούμε να μιλήσουμε για πληροφοριακό σύστημα που οδηγείται από οντολογία. Αντίστοιχα, όταν χρησιμοποιείται στο χρόνο ανάπτυξης, μιλάμε για ανάπτυξη που οδηγείται από την οντολογία.

2.3.1 Χρησιμοποίηση Οντολογίας κατά την Ανάπτυξη του Συστήματος

Στη δεύτερη περίπτωση, μπορούμε να διακρίνουμε δύο διαφορετικά σενάρια. Στο πρώτο σενάριο, οργανώνουμε ένα σύνολο επαναχρησιμοποιήσιμων οντολογιών που έχουμε στη διάθεσή μας, σε μια βιβλιοθήκη οντολογιών που περιέχει τις οντολογίες περιοχών (domain ontologies) και τις οντολογίες δραστηριοτήτων (task ontologies) [Van Heijst, 1997]. Στο δεύτερο σενάριο, έχουμε μόνο μια πολύ γενική οντολογία, στην οποία περιλαμβάνονται διακρίσεις των βασικών εννοιών σε επίπεδο περιοχής, καθώς και αντίστοιχες διακρίσεις στις ταξινομίες και τις σχέσεις. Εδώ ο βαθμός δυνατότητας επαναχρησιμοποίησης είναι πολύ περιορισμένος, δεδομένου ότι η γενική αυτή οντολογία είναι ιδιαίτερα άκαμπτη τόσο όσο αφορά την περίληψη νέων περιοχών, όσο και στην περίπτωση που μόνο ένα μέρος της είναι αναγκαίο να χρησιμοποιηθεί.

Στο πρώτο σενάριο, το σημασιολογικό περιεχόμενο που εκφράζεται από την οντολογία (ή τις οντολογίες) που επιλέγεται από τη βιβλιοθήκη οντολογιών μετασχηματίζεται σε συστατικό του πληροφοριακού συστήματος, που μειώνει τις δαπάνες της εννοιολογικής ανάλυσης του συστήματος και διασφαλίζει – υποθέτοντας μια σωστή οντολογία – την εννοιολογική επάρκεια του πληροφοριακού συστήματος. Εάν το αναπτυσσόμενο πληροφοριακό σύστημα είναι συμβατικό, το εννοιολογικό περιεχόμενο θα ενσωματωθεί στα τυποποιημένα συστατικά του. Αν πρόκειται να για πληροφοριακό σύστημα οδηγούμενο από οντολογία, το αποτέλεσμα αυτής της φάσης ανάπτυξης θα είναι ένα διακριτό υποσύστημα, συγκεκριμένα μια οντολογία εφαρμογής όπως αυτή που απεικονίζεται στο Σχήμα 3, η οποία μπορεί να θεωρηθεί ως ειδίκευση και μιας οντολογίας περιοχής και μιας οντολογίας δραστηριότητας ταυτόχρονα.

Ένα σημαντικό όφελος της χρήσης μιας οντολογίας στο χρόνο ανάπτυξης είναι ότι επιτρέπει στον υπεύθυνο για την ανάπτυξη για να πετύχει ένα

υψηλότερο επίπεδο επαναχρησιμοποίησης από ότι συνήθως συμβαίνει στην τεχνολογία λογισμικού. Η συνήθης περίπτωση στην τεχνολογία λογισμικού είναι η επαναχρησιμοποίηση γνώσης αντί της επαναχρησιμοποίησης λογισμικού, ωστόσο το γεγονός ότι η οντολογία είναι ένας αποτελεσματικός τρόπος να κωδικοποιήσει κανείς τη γνώση καθεαυτή τείνει να εξαλείψει τη διαφορά ανάμεσα στα δύο, εμφυτεύοντας τη γνώση στο ίδιο το λογισμικό και αυξάνοντας έτσι τις δυνατότητες επαναχρησιμοποίησης του. Επιπλέον, επιτρέπει στον υπεύθυνο για την ανάπτυξη να επαναχρησιμοποιήσει και να μοιραστεί τη γνώση περιοχών εφαρμογής χρησιμοποιώντας ένα κοινό λεξιλόγιο ακόμα και ανάμεσα σε ετερογενείς πλατφόρμες λογισμικού. Τέλος του δίνει τη δυνατότητα να επικεντρωθεί στη δομή του συστήματος χωρίς να αποσπάται από τις λεπτομέρειες της υλοποίησης.

Δυστυχώς, η διαθεσιμότητα έτοιμων οντολογιών που να μπορούν να χρησιμοποιηθούν κατ' αυτό τον τρόπο είναι σήμερα εξαιρετικά περιορισμένη. Ο λόγος είναι ότι αυτές οι οντολογίες δεν είναι αρκετά γενικές να ώστε να μπορούν να ειδικευτούν αποτελεσματικά για τις διάφορες εφαρμογές. Υπάρχει δηλαδή μια έλλειψη αφηρημένων οντολογιών που να μπορούν να καλύψουν πολλαπλές περιοχές. Η ανάπτυξη τέτοιων οντολογιών ξεφεύγει από τα πλαίσια μεμονωμένων εφαρμογών και μπορεί να επιτευχθεί μόνο σαν αυτοσκοπός. Η απαιτούμενη επένδυση σε προσπάθεια σε συνδυασμό με την έλλειψη άμεσου οφέλους λόγω της αφηρημένης φύσεως τους, αλλά και η έλλειψη σύγκλισης στις γλώσσες, εργαλεία και τεχνικές ανάπτυξης οντολογιών, είναι οι δύο κυριότεροι ανασταλτικοί παράγοντες για την παγίωση μιας κοινά αποδεκτής υποδομής γενικών οντολογιών. Τέλος, σε εκείνες τις περιπτώσεις όπου η ολοκλήρωση διάφορων off-the-shelf οντολογιών θα ήταν απαραίτητη, θα αντιμετωπίζαμε τα προβλήματα ολοκλήρωσης που συζητούνται στην συνέχεια.

Το δεύτερο σενάριο που απεικονίζεται ανωτέρω εμφανίζεται επομένως ρεαλιστικότερο, παρά τα εγγενή του μειονεκτήματα. Στην πράξη, ο ρόλος της

οντολογίας δεν είναι σε αυτήν την περίπτωση αυτός μιας δομικής μονάδας που προορίζεται να προσαρμοστεί και να επαναχρησιμοποιηθεί, αλλά μάλλον αυτός ενός ισχυρού εργαλείου – ανάλογου με τα υπάρχοντα εργαλεία CASE (Computer Aided Software Engineering) – που μπορεί να αυξήσει την απόδοση της διαδικασίας ανάλυσης. Με άλλα λόγια, σε τέτοιες περιπτώσεις προσανατολιζόμαστε στην ανάπτυξη μιας οντολογίας εφαρμογής με κατ' ευθείαν εξειδίκευση μιας (σχετικά μικρής) οντολογίας κορυφής, χωρίς απαραίτητως την μεσολάβηση μιας προϋπάρχουσας οντολογίας περιοχής ή δραστηριότητας.

Στην ανάπτυξη πληροφοριακών συστημάτων με τη βοήθεια οντολογιών, πρέπει να έχει κανείς υπόψη ότι η οντολογία δεν μπορεί να χρησιμοποιηθεί μόνο για την εκ του μηδενός ανάπτυξη ενός νέου συστήματος. Αντίθετα, έχουν γίνει προσπάθειες για την μελέτη διαδικασιών ανασχεδιασμού (re-engineering) συστημάτων που βρίσκονται ήδη εγκατεστημένα, με χρήση οντολογιών, προκειμένου να βελτιστοποιηθούν οι δυνατότητες επαναχρησιμοποίησης και να διευκολυνθεί η συντήρηση του συστήματος [Partridge, 1996]. Κατ' αυτό τον τρόπο μεγάλες επενδύσεις λογισμικού που έχουν γίνει στο παρελθόν μπορούν να προστατευθούν και να αποκτήσουν νέα αξία.

2.3.2 Χρησιμοποίηση Οντολογίας κατά την Επιχειρησιακή Λειτουργία του Συστήματος

Στην περίπτωση που έχουμε χρησιμοποίηση μιας οντολογίας στο χρόνο εκτέλεσης, πρέπει κατ' αρχήν να διακρίνουμε την περίπτωση ενός πληροφοριακού συστήματος που έχει απλά πρόσβαση σε κάποια οντολογία από αυτή του συστήματος που πραγματικά οδηγείται από οντολογία: στην πρώτη περίπτωση, το πληροφοριακό σύστημα περιέχει ένα συστατικό που γνωρίζει την ύπαρξη μιας (ενδεχομένως απομακρυσμένης) οντολογίας και μπορεί να την χρησιμοποιήσει για κάποιο συγκεκριμένο σκοπό, αναλόγως της εφαρμογής.

Στη δεύτερη περίπτωση, η οντολογία είναι ουσιαστικά ένα υποσύστημα, συνήθως ενσωματωμένο στο πληροφοριακό σύστημα, που αποτελεί αναπόσπαστο μέρος του και προσφέρει προς την επίτευξη του γενικότερου σκοπού του συστήματος. Ένας σημαντικός λόγος ύπαρξης για μια οντολογία στο χρόνο εκτέλεσης, είναι ότι αυτή επιτρέπει την επικοινωνία μεταξύ πρακτόρων λογισμικού (software agents). Οι πράκτορες λογισμικού επικοινωνούν ο ένας με τον άλλον μέσω μηνυμάτων που περιέχουν εκφράσεις που διατυπώνονται χρησιμοποιώντας τους όρους κάποιας οντολογίας (ontology-driven communication) [FIPA, 2000]. Για να κατανοήσει ένας πράκτορας λογισμικού την έννοια αυτών των εκφράσεων, χρειάζεται να έχει πρόσβαση στην οντολογία στην οποία αυτές παραπέμπουν.

Όσο αφορά τη δομική διάσταση, κάθε ένα από τα συστατικά του πληροφοριακού συστήματος – προγράμματα εφαρμογής, βάσεις δεδομένων, οι διαπροσωπείες με τον χρήστη – μπορεί να χρησιμοποιήσει μια οντολογία με συγκεκριμένο τρόπο. Παρακάτω, εξετάζουμε κάθε ένα από αυτά τα συστατικά και περιγράφουμε το συγκεκριμένο ρόλο που μια οντολογία μπορεί να διαδραματίσει, τόσο σε χρόνο ανάπτυξης όσο και σε χρόνο εκτέλεσης.

Η προφανέστερη χρήση μιας οντολογίας είναι στο τμήμα βάσεων δεδομένων. Στην πραγματικότητα, μια οντολογία μπορεί να συγκριθεί με το σχήμα μιας βάσης δεδομένων. Στο χρόνο ανάπτυξης, μια οντολογία μπορεί να διαδραματίσει έναν σημαντικό ρόλο στην ανάλυση απαιτήσεων και την φάση εννοιολογικής μοντελοποίησης. Για παράδειγμα, οντολογίες που είναι συνδεδεμένες με λεξικολογικούς πόρους όπως το WordNet [Miller, 1995] έχουν χρησιμοποιηθεί για την αυτόματη ανάλυση προδιαγραφών σε κείμενο φυσικής γλώσσας [Horpenbrouwers, 1996]. Το προκύπτον εννοιολογικό πρότυπο μπορεί να αναπαρασταθεί ως οντολογία επεξεργάσιμη από υπολογιστή, και από εκεί να μεταφραστεί σε σχήμα βάσεως για τις συγκεκριμένες πλατφόρμες-στόχους. Τέτοια θέματα έχουν μελετηθεί εκτενώς στο πρόγραμμα του ESPRIT IDEA [Ceri,

1997], ειδικά όσο αφορά τη μετάβαση από την οντολογία στα σχήματα για πολλούς διαφορετικούς τύπους βάσεων δεδομένων (relational, object-oriented κλπ).

Μια σημαντικότερη χρήση οντολογιών στο χρόνο ανάπτυξης είναι η ολοκλήρωση πληροφοριών: ένα κοινό εννοιολογικό σχήμα που χρησιμοποιείται παραδείγματος χάριν σε μια εφαρμογή αποθήκευσης στοιχείων (data warehousing) μπορεί να στηριχτεί σε ημιαυτόματη αντιστοίχιση μεταξύ ετερογενών σχημάτων και μιας κεντρικής οντολογίας. Στην περίπτωση που δεν υπάρχει ζήτημα ολοκλήρωσης βάσεων, αλλά η ανάγκη σχεδιασμού πολλών βάσεων που λειτουργούν στην ίδια περιοχή, τότε μια οντολογία που προκύπτει από την εννοιολογική ανάλυση μπορεί να χρησιμεύσει σαν ένα εύχρηστο μέσο εξάλειψης της ετερογένειας, εξασφαλίζοντας ότι όλα τα επιμέρους σχήματα είναι συμβατά με την κεντρική οντολογία, και άρα μεταξύ τους.

Στο χρόνο εκτέλεσης, υπάρχουν πολλοί τρόποι με τους οποίους οι οντολογίες και οι βάσεις δεδομένων μπορούν να συνεργαστούν. Η διαθεσιμότητα των οντολογιών που περιγράφουν τις πηγές πληροφοριών είναι ο πυρήνας της μεθόδου ολοκλήρωσης που βασίζεται στη μεσολάβηση [Wiederhold, 1996]. Οι οντολογίες μπορούν να απαντήσουν ερωτήματα σχετικά με το περιεχόμενο μιας συγκεκριμένης βάσης δεδομένων ή να υποστηρίξουν τη δυναμική διαχείριση των ερωτήσεων που περιλαμβάνουν πολλαπλές βάσεις δεδομένων.

Ίσως όχι τόσο προφανής, αλλά εντούτοις πολύ σημαντική, είναι η χρήση οντολογίας για να υποβοηθήσει τη διαπροσωπεία με τον χρήστη. Η δυνατότητα διασύνδεσης οντολογιών με λεξικολογικό περιεχόμενο μπορεί να αξιοποιηθεί για να συνδέσει το χρήστη με τις διαδικασίες, τις υπηρεσίες και το περιεχόμενο του συστήματος, με χρήση φυσικής γλώσσας [Tsiara, 2002]. Στο χρόνο τρεξίματος, ο πρώτος ρόλος που μια οντολογία μπορεί να διαδραματίσει μέσα στη διαπροσωπεία με τον χρήστη είναι να ανακτηθεί και να εξετασθεί από το χρήστη. Σε αυτήν την περίπτωση, ο χρήστης γνωρίζει την οντολογία, και τη

χρησιμοποιεί ως τμήμα της κανονικής χρήσης του πληροφοριακού συστήματος. Έτσι, ο χρήστης μπορεί να συμβουλευθεί την οντολογία προκειμένου να βρει πληροφορίες για το λεξιλόγιο που χρησιμοποιείται από πληροφοριακό σύστημα, έτσι ώστε να μπορέσει να διατυπώσει τα ερωτήματα του (queries).

Τα προγράμματα εφαρμογής είναι ένα σημαντικό μέρος πολλών πληροφοριακών συστημάτων. Περιέχουν συνήθως μεγάλο όγκο γνώσης της συγκεκριμένης περιοχής, η οποία, για διάφορους λόγους, δεν αποθηκεύεται ρητά στη βάση δεδομένων. Μερικά μέρη αυτής της γνώσης κωδικοποιούνται στο στατικό μέρος του προγράμματος υπό μορφή τύπων ή κλάσεων, ενώ άλλα μέρη, όπως παραδείγματος χάριν οι επιχειρησιακοί κανόνες (business rules) αποθηκεύονται έμμεσα στο, συχνά πολύπλοκο, διαδικαστικό μέρος του προγράμματος.

Σε χρόνο ανάπτυξης, ένας προγραμματιστής του πληροφοριακού συστήματος μπορεί, σε γενικές γραμμές, να παράγει το στατικό μέρος ενός προγράμματος με τη βοήθεια μιας οντολογίας. Στο χρόνο εκτέλεσης, μπορούμε να αποφασίσουμε να αντιπροσωπεύσουμε ρητά όλη τη γνώση περιοχών που κωδικοποιείται σιωπηρά στο πρόγραμμα εφαρμογής, που γυρίζει το πρόγραμμα σε μια γνωσιακή βάση. Όπως καλά είναι γνωστό, αυτό έχει μεγάλα οφέλη από την άποψη της ευκολίας συντήρησης, της επεκτασιμότητας και της ευελιξίας. Σε αυτήν την περίπτωση, η βάση γνώσεων θα μπορούσε να διασπασθεί σε μια οντολογία και μια βάση γνώσης, όπως εξηγήθηκε παραπάνω.

3 ΣΧΕΔΙΑΣΗ ΟΝΤΟΛΟΓΙΩΝ ΓΙΑ ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ

Το 1991, στα πλαίσια του Knowledge Sharing Effort της DARPA [Neches, 1991] αναπτύχθηκε μια καινοτόμος προσέγγιση στην κατασκευή ευφυών πληροφοριακών συστημάτων. Η νέα πρόταση ανέφερε: «Η κατασκευή συστημάτων βασισμένων στη γνώση (knowledge based systems) σήμερα βασίζεται στην ανάπτυξη νέων βάσεων γνώσης εκ του μηδενός. Το ίδιο αποτέλεσμα θα μπορούσε να επιτευχθεί συνδυάζοντας επαναχρησιμοποίηση τμήματα. Η ανάπτυξη του συστήματος θα μπορούσε τότε να επικεντρωθεί στην δημιουργία της εξειδικευμένης γνώσης και των εργαλείων εξαγωγής συμπερασμάτων (inference engines) που αποτελούν ιδιαιτερότητα του νέου συστήματος. Αυτό το νέο σύστημα θα μπορούσε να συνεργαστεί με υπάρχοντα συστήματα και να τα χρησιμοποιήσει για να πραγματοποιήσει μέρος των απαιτούμενων λογικών συνειρμών (reasoning). Έτσι, η γνώση, οι τεχνικές λύσης προβλημάτων και οι υπηρεσίες εξαγωγής συμπερασμάτων θα μπορούσαν να είναι κοινά για όλα τα υποσυστήματα. Αυτή η προσέγγιση θα μπορούσε να επιτρέψει την φθηνή κατασκευή μεγαλύτερων και καλύτερων συστημάτων...»

Την ίδια περίπου περίοδο, πραγματοποιήθηκαν διάφορα ερευνητικά προγράμματα με σκοπό την ανάπτυξη μεθοδολογιών δόμησης συστημάτων βασισμένων στη γνώση. Αν και τα περισσότερα δεν χρησιμοποίησαν ρητά τον όρο οντολογία, έθεσαν όμως τα θεμέλια για τη σχετική αντίληψη στην κοινότητα των ερευνητών της μηχανικής της γνώσης. Στα προγράμματα αυτά περιλαμβάνονται το Task Structures, [Chandrasekaran, 1992], το Role-Limiting Methods [Marcus, 1988], το CommonKADS [Schreiber, 1999], το Protege [Musen, 1993], το MIKE [Angele, 1998], το Components of Expertise [Steels, 1990], το EXPECT [Swartout, 1995], το GDM [Terpstra, 1993] και το VITAL [Domingue, 1998].

Από τότε, σημαντική πρόοδος έχει επιτευχθεί στην ανάπτυξη των βάσεων για την δημιουργία τεχνολογίας που επιτρέπει την επικοινωνία, διαλειτουργικότητα και επαναχρησιμοποίηση συστημάτων και υποσυστημάτων γνώσης. Στη συνέχεια αυτού του κεφαλαίου, παρουσιάζουμε τις πλέον σημαντικές προσεγγίσεις στην ανάπτυξη μεθοδολογιών, γλωσσών και εργαλείων για την ανάπτυξη οντολογιών.

3.1 ΔΟΜΙΚΑ ΣΥΣΤΑΤΙΚΑ

Οι οντολογίες επιχειρούν να περιγράψουν μια συγκεκριμένη γνωστική περιοχή (domain) με τρόπο γενικό ώστε να μπορεί να αποτελέσει μια κοινά αποδεκτή αντίληψη της περιοχής αυτής, η οποία να μπορεί να χρησιμοποιηθεί από πολλαπλά συστήματα [Chandrasekaran, 1999]. Έτσι μια οντολογία παρέχει μια τυποποίηση της έννοιας των όρων που απαρτίζουν το λεξιλόγιο που απαιτείται για την έκφραση της γνώσης που περιλαμβάνεται σε μια γνωστική περιοχή, καθώς και των μεταξύ τους σχέσεων. Συνήθως είναι βασισμένες σε μια ταξινομική οργάνωση και χρησιμοποιούν έννοιες μοντελοποίησης όπως κλάσεις (classes), σχέσεις (relations), συναρτήσεις (functions), αξιώματα (axioms) και αντικείμενα (objects, instances) [Gruber, 1993].

Βασισμένες στον αφηρημένο ορισμό της οντολογίας του Gruber «μια οντολογία αποτελεί την ρητή έκφραση μιας αντίληψης των εννοιών» [Gruber, 1993], έχουν αναπτυχθεί αρκετοί άλλοι ορισμοί που επιχειρούν να προσδιορίσουν περαιτέρω την έννοια της οντολογίας. Το 1995, οι Guarino και Giaretta [Guarino, 1995] συγκέντρωσαν επτά διαφορετικούς ορισμούς και διαμόρφωσαν αντίστοιχες συντακτικές και σημασιολογικές ερμηνείες. Το 1997, ο Borst [Borst, 1997] εισήγαγε μια μικρή διαφοροποίηση στον ορισμό του Gruber: «μια οντολογία ορίζεται ως ο τυποποιημένος προσδιορισμός μιας κοινής αντίληψης».

Αυτοί οι δύο ορισμοί αναλύθηκαν λίγο αργότερα από τον Studer [Studer, 1998], ο οποίος υποστήριξε ότι «... ο όρος αντίληψη (conceptualisation) αναφέρεται σε ένα αφηρημένο μοντέλο κάποιου φαινομένου του κόσμου, έχοντας προσδιορίσει τις έννοιες που σχετίζονται με αυτό. Το ότι είναι ρητή υπονοεί ότι το είδος των εννοιών που χρησιμοποιούνται καθώς και οι περιορισμοί στη χρήση τους έχουν οριστεί ρητά. Η τυποποίηση αναφέρεται στο γεγονός ότι μια οντολογία πρέπει να είναι εκφρασμένη σε μορφή αναγνώσιμη από υπολογιστές. Τέλος, ο όρος κοινή αναφέρεται στο γεγονός ότι μια οντολογία εκφράζει κοινά αποδεκτή γνώση, η οποία δεν είναι γνωστή ή αποδεκτή σε ένα μόνο άτομο, αλλά σε ένα ολόκληρο σύνολο ατόμων.

Η γνώση τυποποιείται σε μια οντολογία χρησιμοποιώντας πέντε διαφορετικά πρότυπα μοντελοποίησης: κλάσεις, σχέσεις, συναρτήσεις, αξιώματα και αντικείμενα. [Gruber, 1993]:

- Οι κλάσεις οργανώνονται σε ταξινομίες. Συχνά ο όρος οντολογία υποβαθμίζεται έτσι ώστε να ταυτίζεται με την ταξινομία [Studer, 1998]. Οι κλάσεις, ή έννοιες, χρησιμοποιούνται με μια ευρεία σημασία. Μια έννοια μπορεί να αντιπροσωπεύει οτιδήποτε σχετικά με το οποίο θέλουμε να διατυπώσουμε μια πρόταση. Έτσι μπορεί να είναι η περιγραφή μιας εργασίας, μια λειτουργία, μια πράξη, κλπ.

- Οι σχέσεις αντιπροσωπεύουν ένα είδος αλληλεπίδρασης ανάμεσα σε έννοιες της γνωστικής περιοχής. Ορίζονται φορμαλιστικά ως το υποσύνολο οποιουδήποτε καρτεσιανού γινομένου n συνόλων, δηλαδή: $R: C_1 \times C_2 \times \dots \times C_n$. Συνήθως τα σύνολα είναι δύο και οι αντίστοιχες σχέσεις διμερείς.
- Οι συναρτήσεις είναι μια ειδική περίπτωση σχέσεων, όπου το n -οστό στοιχείο είναι μοναδικό όταν τα πρώτα $n-1$ είναι γνωστά. Φορμαλιστικά, οι συναρτήσεις ορίζονται ως εξής: $F: C_1 \times C_2 \times \dots \times C_{n-1} \rightarrow C_n$.
- Τα αξιώματα είναι προτάσεις που αληθεύουν πάντα.
- Τα αντικείμενα αντιπροσωπεύουν απτά στοιχεία που ανήκουν σε κλάσεις ή αντιπροσωπεύουν κάποια ή κάποιες έννοιες. Έτσι, μια κλάση ή έννοια ορίζεται πλήρως από το σύνολο των αντικειμένων που προσδιορίζει.

3.2 ΣΧΕΔΙΑΣΤΙΚΕΣ ΑΡΧΕΣ

Στην οντολογική έρευνα έχουν δοκιμαστεί αρκετές διαφορετικές μεθοδολογίες και προσεγγίσεις στο πρόβλημα της σχεδίασης οντολογιών για διάφορες γνωστικές περιοχές. Στην παράγραφο αυτή, συγκεντρώνουμε και συνοψίζουμε μερικούς γενικούς κανόνες σχεδίασης που έχουν προκύψει ως αποτελέσματα των ερευνών, και οι οποίοι αποσκοπούν στην αποτύπωση μιας οντολογίας ευανάγνωστης, κατά το δυνατό εύκολα συντηρήσιμης και επαναχρησιμοποιήσιμης.

- Σαφήνεια και αντικειμενικότητα [Gruber, 1995]. Σύμφωνα με αυτή την αρχή, η οντολογία θα πρέπει να παρέχει το σωστό νόημα των οριζόμενων όρων τόσο σε οντολογικούς όρους, όσο και με περιγραφή σε φυσική γλώσσα που θα πρέπει να συνοδεύει κάθε ορισμό.
- Πληρότητα [Gruber, 1995]. Για να εξασφαλιστεί η πληρότητα, θα πρέπει να προτιμώνται οι πλήρεις ορισμοί, αυτοί δηλαδή που ορίζουν ικανές και

αναγκαίες συνθήκες. Οι μερικοί ορισμοί, αυτοί δηλαδή που περιέχουν μόνο ικανές ή μόνο αναγκαίες συνθήκες, δεν προσδιορίζουν διεξοδικά τους όρους και πρέπει να αποφεύγονται.

- Συνέπεια [Gruber, 1995], δηλαδή η δυνατότητα να πραγματοποιηθούν συλλογισμοί με βάση τους ορισμούς της οντολογίας, οι οποίοι δεν προκαλούν αντιφάσεις.
- Μέγιστη μονοτονική επεκτασιμότητα (maximum monotonic extendibility) [Gruber, 1995]. Αυτό σημαίνει ότι η εισαγωγή νέων όρων που γενικεύουν ή εξειδικεύουν τους υπάρχοντες στην οντολογία πρέπει να είναι δυνατή χωρίς την ανάγκη διόρθωσης των ορισμών των υπάρχοντων όρων.
- Ελάχιστες οντολογικές δεσμεύσεις (minimal ontological commitments) [Gruber, 1995]. Οι οντολογικές δεσμεύσεις είναι παραδοχές σχετικά με τη γνωστική περιοχή που μοντελοποιεί η οντολογία, οι οποίες γίνονται με σκοπό να διασφαλίσουν την σωστή κατανόηση του χρησιμοποιούμενου από την οντολογία λεξιλογίου. Τέτοιες παραδοχές εγγυώνται την συνέπεια της οντολογίας, όχι όμως και την πληρότητα της. Η ελαχιστοποίηση οντολογικών δεσμεύσεων δίνει έτσι ελευθερία επέκτασης της οντολογίας με γενίκευση ή ειδίκευση, ανάλογα με τις ανάγκες.
- Οντολογική διάκριση [Borgo, 1996], που σημαίνει ότι οι κλάσεις σε μια οντολογία πρέπει να είναι μεταξύ τους διαζευγμένες, δηλαδή ένα αντικείμενο που ανήκει σε μια κλάση πρέπει να μην ανήκει σε καμία άλλη που δεν είναι γενίκευση ή ειδίκευση της.
- Μέγιστη δυνατή διαφοροποίηση των ιεραρχιών, ώστε να βελτιστοποιηθεί η αξιοποίηση της πολλαπλής κληρονομικότητας [Arpírez, 1998].
- Αρθρωτή αρχιτεκτονική (modularity), ώστε να ελαχιστοποιηθεί η αλληλεξάρτηση ανάμεσα σε διαφορετικά τμήματα [Bernaras, 1996].

- Χρήση δόκιμων όρων και τυποποίηση της χρησιμοποιούμενης ονοματολογίας, όποτε αυτό είναι δυνατό [Arpírez, 1998].

3.3 ΜΕΘΟΔΟΛΟΓΙΕΣ ΑΝΑΠΤΥΞΗΣ ΟΝΤΟΛΟΓΙΩΝ

3.3.1 Εισαγωγή

Η ειδοποιός ίσως διαφορά μεταξύ ενός τεχνολογικού πεδίου που βρίσκεται σε πρώιμο στάδιο από ένα ώριμο πεδίο είναι η ύπαρξη ευρέως αποδεκτών μεθοδολογιών στο δεύτερο. Κάθε έργο ανάπτυξης προϋποθέτει τον ορισμό και τυποποίηση ενός κύκλου ζωής που καλύπτει όλες τις φάσεις του έργου, από τη συλλογή προδιαγραφών μέχρι την συντήρηση του παραγόμενου προϊόντος. Η πλέον γνωστή και αποδεκτή μεθοδολογία στο χώρο των συστημάτων γνώσης είναι η KADS [Wielinga, 1992] καθώς και η εξέλιξη της, CommonKADS [Schreiber, 1999].

Οι περισσότερες μεθοδολογίες ανάπτυξης οντολογιών είναι προσανατολισμένες στην κατασκευή τους εκ του μηδενός, ή στην επαναχρησιμοποίηση τους χωρίς αλλαγές. Το 1989, οι Lenat και Guha [Guha, 1989] ξεκίνησαν την ανάπτυξη της οντολογίας CYC. Ο στόχος τους ήταν να δημιουργήσουν μια τεράστια οντολογία «κοινής λογικής» που θα περιείχε ένα μεγάλο όγκο θεμελιώδους ανθρώπινης γνώσης. Ήδη μετά από ένα χρόνο δημοσίευσαν τα πρώτα βήματα της διαδικασίας και κάποιες ενδιαφέρουσες παρατηρήσεις σχετικά με την ανάπτυξη της οντολογίας CYC [Lenat, 1990]. Το 1995, οι πρώτες μεθοδολογικές έρευνες δημοσιεύτηκαν στη βάση της εμπειρίας που αποκομίστηκε από την ανάπτυξη της *Enterprise Ontology* [Uschold, 1995] και του έργου TOVE (*TO*ronto *V*irtual *E*nterprise) [Grüninger, 1995], που επικεντρώθηκαν και τα δύο στο χώρο της μοντελοποίησης επιχειρήσεων. Αργότερα οι μεθοδολογικές προσεγγίσεις αυτές βελτιώθηκαν στα [Uschold, 1996] και [Uschold, 1996]. Το 1996, η Bernaras [Bernaras, 1996] παρουσίασε

τη μέθοδο που χρησιμοποίησε για την ανάπτυξη μιας οντολογίας στην περιοχή των ηλεκτρικών δικτύων, στα πλαίσια του έργου KACTUS του προγράμματος Esprit [KACTUS, 1996]. Την ίδια εποχή εμφανίστηκε η METHONTOLOGY [Gomez Perez, 1996], η οποία αναπτύχθηκε με μεταγενέστερες δημοσιεύσεις [Fernandez, 1997], [Gomez Perez, 1998], [Fernandez Lopez, 1999]. Το 1997, προτάθηκε μια μεθοδολογία δόμησης οντολογιών στη βάση του SENSUS [Swartout, 1997]. Αυτές οι μεθοδολογίες περιγράφονται συνοπτικά στα επόμενα κεφάλαια. Τέλος, υπάρχει μια πρόταση για μια μεθοδολογία ανασχεδιασμού οντολογιών [Gomez Perez, 1999], που επίσης παρουσιάζεται στη συνέχεια, της οποίας ο στόχος είναι να ανακτήσει και μετασχηματίσει το εννοιολογικό μοντέλο μιας υπάρχουσας και υλοποιημένης οντολογίας, ώστε να παραχθεί ένα νέο εννοιολογικό μοντέλο το οποίο μπορεί να υλοποιηθεί εκ νέου.

Όπως αναφέρθηκε παραπάνω, ο στόχος της οντολογίας είναι η παροχή μιας κοινής αντίληψης μιας γνωστικής περιοχής, η οποία είναι αποδεκτή ανάμεσα σε ένα μεγάλο σύνολο ατόμων και μπορεί να χρησιμεύσει για την μονοσήμαντη ερμηνεία του λεξιλογίου που αυτό το σύνολο χρησιμοποιεί. Αφού λοιπόν η οντολογία χαρακτηρίζει ένα ολόκληρο σύνολο ατόμων, είναι κοινό φαινόμενο να πρέπει να αναπτυχθεί από κοινού από ένα σύνολο ανθρώπων που μπορεί να είναι γεωγραφικά απομονωμένοι. Έτσι παρουσιάζουμε επίσης κάποιες προσεγγίσεις που αντιμετωπίζουν το πρόβλημα της συλλογικής δόμησης οντολογιών [Benjamins, 1999], [Euzenat, 1996].

Στη συνέχεια παρουσιάζουμε και αναλύουμε την αιχμή του δόρατος στο χώρο των μεθοδολογιών ανάπτυξης οντολογιών εκ του μηδενός, αλλά και επαναχρησιμοποίησής τους. Επίσης, τις συγκρίνουμε με το πρότυπο του IEEE για την ανάπτυξη διαδικασιών κύκλου ζωής λογισμικού (*IEEE Standard for Developing Software Life Cycle Processes*, 1074-1995) [IEEE, 1996]. Παρουσιάζουμε κριτήρια για την ανάλυση των μεθοδολογιών, με βάση τα οποία

τις κατατάσσουμε. Τέλος αναλύουμε τις μεθοδολογίες με βάση τα κριτήρια αυτά και παρουσιάζουμε κάποια γενικά συμπεράσματα.

3.3.2 Το Πρότυπο IEEE 1074-1995

Το πρότυπο 1074-1995 του IEEE περιγράφει τη διαδικασία ανάπτυξης λογισμικού, τις δραστηριότητες που πρέπει να ολοκληρωθούν και τεχνικές που μπορούν να χρησιμοποιηθούν. Οι δραστηριότητες δεν παρουσιάζονται με χρονολογική σειρά, εφ' όσον το πρότυπο προτείνει την ενσωμάτωση τους σε στον κύκλο ζωής του λογισμικού, ο οποίος επιλέγεται και καθορίζεται από τον χρήστη για κάθε συγκεκριμένο έργο. Επίσης, το πρότυπο δεν ορίζει κάποιο συγκεκριμένο κύκλο ζωής. Αυτές οι δραστηριότητες εντάσσονται σε αυτό που ονομάζει διαδικασία λογισμικού, η οποία διαχωρίζεται σε τέσσερις επιμέρους διαδικασίες. Αυτές είναι:

1. **Διαδικασίες μοντελοποίησης κύκλου ζωής:** περιλαμβάνουν τις δραστηριότητες επιλογής και καθορισμού ενός κύκλου ζωής λογισμικού, ο οποίος καθορίζει τη σειρά με την οποία όλες οι υπόλοιπες δραστηριότητες εμφανίζονται.

2. **Διαδικασίες διαχείρισης έργου:** καθορίζουν το πλαίσιο του έργου και διασφαλίζουν το κατάλληλο επίπεδο διαχείρισης σε όλη τη διάρκεια ζωής του έργου. Οι δραστηριότητες της εκκίνησης έργου, ελέγχου και διαχείρισης ποιότητας ανήκουν σ' αυτή την ομάδα διεργασιών.

3. **Διαδικασίες ανάπτυξης λογισμικού:** περιλαμβάνουν την παραγωγή, εγκατάσταση, λειτουργία, συντήρηση και τελικά απόσυρση του λογισμικού. Διαχωρίζονται περαιτέρω στις ακόλουθες ομάδες:

3.1. **Διαδικασίες προετοιμασίας ανάπτυξης:** εκτελούνται πριν την εκκίνηση της διαδικασίας ανάπτυξης. Περιλαμβάνουν δραστηριότητες που σχετίζονται με τη μελέτη του περιβάλλοντος στο οποίο θα λειτουργήσει το λογισμικό, καθώς και τις μελέτες σκοπιμότητας.

3.2. **Διαδικασίες ανάπτυξης:** αποτελούν μέρος της παραγωγής του προϊόντος. Περιλαμβάνουν τις εξής διαδικασίες:

- **Διαδικασία απαιτήσεων:** περιλαμβάνει επαναλαμβανόμενες διαδικασίες που στοχεύουν στην κατάρτιση των απαιτήσεων που πρέπει να πληροί το προϊόν.
- **Διαδικασία σχεδιασμού:** στοχεύει στην ανάπτυξη μιας λεπτομερούς και συνεπούς αναπαράστασης του λογισμικού που να ικανοποιεί την προδιαγραφή απαιτήσεων και να μπορεί να χρησιμοποιηθεί σαν οδηγός για την ανάπτυξη του.
- **Διαδικασία υλοποίησης:** μετατρέπει την αναπαράσταση που παράγει ο σχεδιασμός σε πραγματικό προϊόν λογισμικού, πραγματοποιημένο σε κάποια γλώσσα προγραμματισμού.

3.3. **Διαδικασίες μετά την ανάπτυξη:** περιλαμβάνουν την εγκατάσταση, λειτουργία και υποστήριξη, και απόσυρση του προϊόντος.

4. **Υποστηρικτικές διαδικασίες:** περιλαμβάνουν δραστηριότητες που δεν παράγουν λογισμικό, αλλά είναι απαραίτητες για την ολοκλήρωση της ανάπτυξης του και την εξασφάλιση της ποιότητας του, όπως για παράδειγμα η διαμόρφωση του λογισμικού, η ανάπτυξη εγχειριδίων, η εκπαίδευση.

Σύμφωνα με τον ορισμό του IEEE, το λογισμικό είναι «... προγράμματα υπολογιστών, διαδικασίες και πιθανώς έγγραφα και δεδομένα που σχετίζονται με τη λειτουργία ενός υπολογιστικού συστήματος» [IEEE, 1990]. Σύμφωνα με αυτό τον ορισμό, οι οντολογίες αποτελούν μέρος του πληροφοριακού συστήματος και επομένως η ανάπτυξη τους εμπίπτει στο πλαίσιο των προτύπων ανάπτυξης λογισμικού γενικά, τα οποία βέβαια θα πρέπει να προσαρμοστούν στα ειδικά χαρακτηριστικά τους. Παρακάτω εξετάζουμε το βαθμό στον οποίο οι τυποποιημένες διαδικασίες του IEEE μπορούν να εφαρμοστούν σε μια μεθοδολογία ανάπτυξης οντολογιών.

1. **Διαδικασίες μοντελοποίησης κύκλου ζωής:** Η μεθοδολογία θα πρέπει να προτείνει ένα σύνολο από μοντέλα κύκλου ζωής, από τα οποία ο χρήστης θα πρέπει να επιλέξει το κατάλληλο.

2. **Διαδικασίες διαχείρισης έργου:** Οι διαδικασίες και δραστηριότητες αυτής της ομάδας είναι εφαρμόσιμες σε κάθε έργο λογισμικού, επομένως θα πρέπει να εφαρμόζονται και στην ανάπτυξη οντολογιών.

3. **Διαδικασίες ανάπτυξης λογισμικού:** Γενικά οι διαδικασίες αυτές είναι εφαρμόσιμες στις οντολογίες, με κάποιες διαφοροποιήσεις ανά περίπτωση. Αναλυτικά:

3.1. **Διαδικασίες προετοιμασίας ανάπτυξης:** Η διαδικασία μελέτης σκοπιμότητας (feasibility study) έχει νόημα για κάθε έργο πληροφορικής. Στη μελέτη του περιβάλλοντος του έργου θα πρέπει να δοθεί έμφαση στην χρήση υπάρχουσών οντολογιών και στην ολοκλήρωση τους με συγγενή συστήματα.

3.2. **Διαδικασίες ανάπτυξης:**

- **Διαδικασία απαιτήσεων:** Η τυποποίηση απαιτήσεων για οντολογίες δεν έχει εξελιχθεί επαρκώς. Κάποιες πρώτες ερευνητικές προσπάθειες [Gomez Perez, 1998] δείχνουν ότι είναι δυνατή η μερική τουλάχιστον εξαγωγή απαιτήσεων που πρέπει να πληρεί μία οντολογία.
- **Διαδικασία σχεδιασμού:** Οι οντολογίες υπόκεινται σε διαδικασία σχεδιασμού, αν και διαφορετική αυτής που χρησιμοποιείται συνήθως για λογισμικό. Το πνεύμα του προτύπου, που αποτρέπει την άμεση μετάβαση από τις προδιαγραφές στην υλοποίηση, είναι πάντως εφαρμόσιμο.
- **Διαδικασία υλοποίησης:** Η υλοποίηση είναι αναγκαία, επομένως οι αντίστοιχες διαδικασίες είναι εφαρμόσιμες.

3.3. **Διαδικασίες μετά την ανάπτυξη:** Αυτές είναι κοινές για κάθε είδος λογισμικού.

4. **Υποστηρικτικές διαδικασίες:** Αυτές είναι κοινές για κάθε είδος λογισμικού.

3.3.3 Κριτήρια Ανάλυσης Μεθοδολογιών

Στη συνέχεια αυτής της μελέτης, αξιολογούμε κάθε μεθοδολογία ανάπτυξης οντολογιών με βάση τα παρακάτω κριτήρια.

1. **Εμπειρία από τη μηχανική γνώσης.** Εκφράζει την ενσωμάτωση στη μεθοδολογία των αποτελεσμάτων από τη μελέτη της μηχανικής γνώσης.
2. **Λεπτομέρεια.** Αξιολόγηση του κατά πόσο οι τεχνικές και δραστηριότητες που προτείνει η μεθοδολογία είναι επακριβώς ορισμένες.
3. **Αναπαράσταση γνώσης.** Αξιολόγηση των προτύπων αναπαράστασης γνώσης που υποστηρίζει η μεθοδολογία.
4. **Στρατηγική ανάπτυξης εφαρμογών.** Οι στρατηγικές που ακολουθούνται εμπίπτουν στις ακόλουθες κατηγορίες:
 - I. **Για χρήση από συγκεκριμένη εφαρμογή:** Η οντολογία αναπτύσσεται βασισμένη σε μια βάση γνώσης που αφορά συγκεκριμένη εφαρμογή, χρησιμοποιώντας μια διαδικασία αφαίρεσης.
 - II. **Για συγκεκριμένες χρήσεις:** Η οντολογία δεν αναπτύσσεται για συγκεκριμένη εφαρμογή, αλλά τα πιθανά σενάρια χρήσης της καθορίζονται στη φάση απαιτήσεων και επηρεάζουν την ανάπτυξη της.
 - III. **Ανεξάρτητα από τη χρήση:** Η διαδικασία ανάπτυξης είναι εντελώς ανεξάρτητη από τη χρήση της οντολογίας σε διάφορα συστήματα.
5. **Αναγνώριση εννοιών.** Οι κυριότερες στρατηγικές αναγνώρισης εννοιών είναι από το γενικό προς το ειδικό (top-down), από το ειδικό προς το γενικό

(bottom-up) και από το πιο σχετικό προς τις δύο κατευθύνσεις (middle-out) [Uschold, 1996].

6. **Κύκλος ζωής.** Αφορά το κατά πόσο η μεθοδολογία προτείνει, άμεσα ή έμμεσα, ένα μοντέλο κύκλου ζωής.
7. **Ενσωμάτωση του προτύπου IEEE 1074-1995.** Αξιολογεί το κατά πόσο η μεθοδολογία υποστηρίζει τις διαδικασίες και δραστηριότητες που προβλέπονται από το πρότυπο.
8. **Προτεινόμενες τεχνικές.** Αφορά το κατά πόσο συγκεκριμένες τεχνικές προτείνονται για την πραγματοποίηση κάθε δραστηριότητας που προβλέπεται από τη μεθοδολογία.
9. **Εφαρμογές.** Τυχόν οντολογίες ή συστήματα που έχουν αναπτυχθεί με χρήση της μεθοδολογίας.
10. **Υποστηρικτικά εργαλεία.** Η ύπαρξη εργαλείων που μπορούν να χρησιμοποιηθούν για την υποβοήθηση των δραστηριοτήτων της μεθοδολογίας.

Τα κριτήρια 1-5 αφορούν γενικά χαρακτηριστικά της μεθοδολογίας. Τα υπόλοιπα κριτήρια εξετάζουν την ωριμότητα κάθε μεθοδολογίας, καθώς και την ευκολία πρακτικής εφαρμογής της. Ένα ακόμα ενδιαφέρον κριτήριο θα αποτελούσε η υποστήριξη για ομαδική ανάπτυξη της οντολογίας, πιθανώς από ομάδες γεωγραφικά απομακρυσμένες. Ωστόσο, στη βιβλιογραφία δεν υπάρχουν διαθέσιμα στοιχεία ώστε να ποσοτικοποιήσουμε ένα τέτοιο κριτήριο για κάθε μεθοδολογία.

3.3.4 Cys

Ο κύριος στόχος των δημιουργών της Cys υπήρξε η δημιουργία μιας τεράστιας βάσης γνώσης κοινής λογικής. Η Cys μπορεί να αντιμετωπισθεί ως οντολογία, αφού ο στόχος της είναι να αποτελέσει το υπόβαθρο για την

κατασκευή διαφόρων συστημάτων, καθώς και για την ολοκλήρωση και επικοινωνία τους.

Η ανάπτυξη της Cyc χωρίζεται στις ακόλουθες φάσεις [Lenat, 1990]:

1. Κωδικοποίηση της βασικής, ρητά ή έμμεσα εκφρασμένης γνώσης που μπορεί να αντληθεί από τις σχετικές πηγές χωρίς τη βοήθεια συστημάτων αυτόματης μάθησης ή επεξεργασίας ανθρώπινης γλώσσας.
2. Κωδικοποίηση γνώσης με τη βοήθεια εργαλείων αυτόματης μάθησης ή επεξεργασίας φυσικής γλώσσας, τα οποία βασίζονται στην ήδη αποκτηθείσα γνώση από την πρώτη φάση. Η εκτέλεση αυτής της φάσης δεν ήταν εφικτή κατά την ανάπτυξη της Cyc λόγω της ανωριμότητας των απαιτούμενων τεχνολογιών, αλλά και μέχρι σήμερα δεν έχει εφαρμοστεί στην πράξη.
3. Σχεδόν πλήρης αυτοματοποίηση της συλλογής νέας γνώσης. Ο ανθρώπινος ρόλος θα περιοριστεί στην επιλογή κειμένων για επεξεργασία και στην επεξήγηση των πολυπλοκότερων τμημάτων τους.

Σε κάθε φάση προβλέπεται η εκτέλεση δύο δραστηριοτήτων. Η πρώτη αφορά την ανάπτυξη μιας οντολογίας αναπαράστασης γνώσης, η οποία περιέχει τα στοιχεία που χρησιμοποιούνται για την μοντελοποίηση της γνώσης (κλάσεις, αντικείμενα, συναρτήσεις κλπ). Η δεύτερη αφορά την αναπαράσταση της γνώσης με χρήση αυτών των στοιχείων.

3.3.4.1 Αξιολόγηση

Συγκρίνοντας τη μεθοδολογία Cyc με τα κριτήρια που θέσαμε παραπάνω, έχουμε τα εξής αποτελέσματα:

1. Δεν κληρονομεί χαρακτηριστικά από κάποια μεθοδολογία ανάπτυξης βάσεων γνώσης, αλλά ξεκινά εκ του μηδενός.

2. Η μεθοδολογία δεν παρουσιάζεται λεπτομερώς, αφού στη βιβλιογραφία [Lenat, 1990] εξηγούνται μόνο οι κυριότερες πτυχές του προβλήματος ανάπτυξης.
3. Η γνώση αναπαρίσταται με τη χρήση πλαισίων (frames), στα οποία μπορούν να τεθούν λογικοί περιορισμοί.
4. Η στρατηγική ανάπτυξης εφαρμογών που χρησιμοποιείται είναι η εξαρτώμενη από την εφαρμογή ανάπτυξη της οντολογίας.
5. Η αναγνώριση εννοιών γίνεται με το middle-out πρότυπο.
6. Όσο αφορά τον κύκλο ζωής, περιγράφονται συνοπτικά οι φάσεις ανάπτυξης. Αν αυτό και δεν αναφέρεται ρητά, υπονοείται ότι ο χρησιμοποιούμενος κύκλος ζωής είναι σπειροειδής, με εξελισσόμενα πρωτότυπα συστήματα.
7. Η Cysc έχει σημαντικές αποκλίσεις από το πρότυπο IEEE 1074, με σημαντικότερη ίσως την μη υποστήριξη της φάσης σχεδιασμού. Επίσης δεν ορίζονται διαδικασίες προετοιμασίας και περάτωσης ανάπτυξης, ούτε και υποστηρικτικές διαδικασίες.
8. Η Cysc αναπτύχθηκε με χρήση της γλώσσας CycL, για την οποία παρέχονται κάποιες λεπτομέρειες. Ωστόσο, ολοκληρωμένες τεχνικές για την ανάπτυξη της Cysc δεν υπάρχουν καταγεγραμμένες.
9. Μέχρι σήμερα, αυτή η μεθοδολογία έχει χρησιμοποιηθεί μόνο για την ανάπτυξη της Cysc, η οποία ωστόσο περιέχει διάφορες επί μέρους οντολογίες που αντιμετωπίζουν τη γνώση μιας περιοχής από διαφορετική οπτική γωνία. Ωστόσο δεν είναι διαθέσιμη επαρκής τεκμηρίωση για τις περιοχές που αναλύθηκαν στη Cysc. Στη βάση της Cysc έχουν αναπτυχθεί διάφορα συστήματα που την αξιοποιούν. Οι σημαντικότερες εφαρμογές της είναι στην επεξεργασία φυσικής γλώσσας και στην ολοκλήρωση ετερογενών βάσεων δεδομένων.
10. Τα εργαλεία δόμησης της Cysc δεν είναι διαθέσιμα ελεύθερα.

3.3.5 Enterprise Ontology

Αυτή η μεθοδολογία αναπτύχθηκε από τους Uschold και King με βάση την αποκτηθείσα εμπειρία από την ανάπτυξη της Enterprise Ontology [Uschold, 1995], μιας οντολογίας για τη μοντελοποίηση μεγάλων επιχειρήσεων. Η μεθοδολογία βασίζεται στα ακόλουθα βήματα:

1. Αναγνώριση του στόχου. Περιγράφει το γιατί αναπτύσσεται η οντολογία και ποιες είναι οι προβλεπόμενες χρήσεις της.
2. Ανάπτυξη της οντολογίας. Αναλύεται στις εξής δραστηριότητες:
 - I. Σύλληψη της οντολογίας (ontology capture). Περιλαμβάνει την αναγνώριση των βασικών εννοιών και μεταξύ τους σχέσεων και την μονοσήμαντη λεκτική περιγραφή τους. Ουσιαστικά στοχεύει στον καθορισμό του πλαισίου μέσα στο οποίο πρέπει να κινηθεί η ανάπτυξη της οντολογίας. Στη συνέχεια ακολουθεί η επιλογή όρων για αναφορά σε αυτές τις έννοιες και σχέσεις. Προφανώς η διαδικασία αυτή ακολουθεί την τεχνική middle-out, αφού ξεκινά από τους πλέον σημαντικούς κόμβους της οντολογίας, όχι από την κορυφή ή τη βάση.
 - II. Κωδικοποίηση. Αποτελεί την φορμαλιστική αναπαράσταση της γνώσης που συγκεντρώθηκε στο προηγούμενο βήμα.
 - III. Ολοκλήρωση με υπάρχουσες οντολογίες. Αυτή η δραστηριότητα εκτελείται παράλληλα με τις προηγούμενες δύο και στοχεύει στην επαναχρησιμοποίηση έτοιμης γνώσης.
3. Αξιολόγηση. Η αξιολόγηση αφορά το περιεχόμενο της οντολογίας, τα σχετικά εργαλεία λογισμικού και την γραπτή τεκμηρίωση κάθε είδους, και πραγματοποιείται σε σχέση με κάποιο συγκεκριμένο πλαίσιο αναφοράς, για παράδειγμα τις προδιαγραφές και απαιτήσεις του συστήματος.
4. Τεκμηρίωση. Περιλαμβάνει τη γραπτή περιγραφή της οντολογίας, ανάλογα με τον τύπο και τις προβλεπόμενες χρήσεις της.

3.3.5.1 Αξιολόγηση

1. Η μεθοδολογία αξιοποιεί τις μεθόδους ανάπτυξης συστημάτων βασισμένων στη γνώση, με την έννοια ότι διακρίνει καθαρά τα στάδια της απόκτησης, κωδικοποίησης και αξιολόγησης της γνώσης. Ωστόσο δεν προτείνεται μελέτη σκοπιμότητας, ούτε και δημιουργία πρωτοτύπου του συστήματος.
2. Η μεθοδολογία δεν είναι λεπτομερής, αφού δεν περιγράφει αναλυτικά τις δραστηριότητες και τεχνικές τους.
3. Δεν υπάρχει περιορισμός στο μοντέλο αναπαράστασης γνώσης.
4. Η διαδικασία είναι εντελώς ανεξάρτητη από την χρήση της αναπτυσσόμενης οντολογίας.
5. Όπως ήδη αναφέρθηκε, η οντολογία δομείται με τεχνική middle-out.
6. Δεν υπάρχει πρόταση για τον κύκλο ζωής.
7. Δεν προτείνονται διαδικασίες προετοιμασίας και περάτωσης της ανάπτυξης, αλλά ούτε και διαδικασία σχεδιασμού. Κάποιες δραστηριότητες δεν περιγράφονται στις διαδικασίες που προτείνονται, ιδιαίτερα η μελέτη περιβάλλοντος και η μελέτη σκοπιμότητας.
8. Οι τεχνικές για την ολοκλήρωση των διαφόρων δραστηριοτήτων δεν περιγράφονται αναλυτικά. Για παράδειγμα, η μεθοδολογία προτείνει την αναγνώριση των κυριότερων εννοιών και σχέσεων της περιοχής σαν πρώτο βήμα της απόκτησης γνώσης. Ωστόσο, δεν εξηγείται πώς θα επιτευχθεί αυτό.
9. Η Enterprise Ontology αποτελεί ένα μεγάλο και σημαντικό έργο, που εκτελέσθηκε από το Πανεπιστήμιο του Εδιμβούργου σε συνεργασία με τις εταιρίες IBM, Lloyd's, Logica και Unilever.
10. Η ανάπτυξη της οντολογίας βασίστηκε κατά κανόνα στον Ontolingua Server. Προσφέρεται επίσης μια συλλογή εργαλείων με το όνομα Enterprise Toolkit,

η οποία περιέχει εργαλεία μοντελοποίησης διαδικασιών, ανάπτυξης πρακτόρων (agents), εκτέλεσης εργασιών κλπ.

3.3.6 TOVE

Αυτή η μεθοδολογία αναπτύχθηκε στο έργο TOVE που αφορά την περιοχή των επιχειρηματικών διαδικασιών και δραστηριοτήτων [Grüniger, 1995]. Βασίζεται στη δημιουργία ενός λογικού μοντέλου της γνώσης που πρόκειται να κωδικοποιηθεί στην οντολογία. Το μοντέλο αυτό δεν δημιουργείται απευθείας. Πρώτα παράγεται μια άτυπη περιγραφή των προδιαγραφών που πρέπει να πληρούνται από την οντολογία και στη συνέχεια αυτή η περιγραφή εκφράζεται φORMALISΤΙΚΑ. Αυτό επιτυγχάνεται με την ακόλουθη διαδικασία:

1. Σύλληψη βασικών σεναρίων. Η δημιουργία της οντολογίας θεωρείται ότι προκαλείται από σενάρια που προκύπτουν στην εφαρμογή. Τα σενάρια αυτά είναι παραδείγματα ή προβλήματα που δεν καλύπτονται από τις υπάρχουσες οντολογίες. Ένα τέτοιο σενάριο περιέχει επίσης και πιθανές διαισθητικές λύσεις στα προβλήματα που περιγράφει.
2. Δόμηση άτυπων κριτηρίων πληρότητας. Αυτά βασίζονται στα σενάρια του προηγούμενου βήματος και αποτελούν ουσιαστικά απαιτήσεις πληρότητας που θα πρέπει να ικανοποιήσει η οντολογία, και τα οποία παίρνουν τη μορφή ερωτήσεων. Η οντολογία πρέπει να διαθέτει την απαραίτητη ορολογία ώστε να μπορεί να εκφράσει αυτές τις ερωτήσεις, και να μπορεί να περιγράψει τις απαντήσεις τους με τη μορφή αξιωμάτων ή ορισμών. Αυτές οι ερωτήσεις είναι άτυπα κριτήρια, αφού δεν εκφράζονται με τη βοήθεια της γλώσσας αναπαράστασης γνώσης που χρησιμοποιεί η οντολογία.
3. Καθορισμός της ορολογίας της οντολογίας μέσω της γλώσσας αναπαράστασης γνώσης. Αυτό επιτυγχάνεται με τα ακόλουθα βήματα:

- I. Ορισμός άτυπης ορολογίας. Αυτός προκύπτει από τις απαντήσεις στις ερωτήσεις παραπάνω. Χρησιμοποιούνται σαν βάση για την κατασκευή της φορμαλιστικής ορολογίας.
 - II. Ορισμός τυπικής ορολογίας. Αυτός ο ορισμός προκύπτει εκφράζοντας την άτυπη ορολογία σε όρους της γλώσσας αναπαράστασης γνώσης που χρησιμοποιείται.
4. Δόμηση φορμαλιστικών κριτηρίων πληρότητας. Έχοντας διαθέσιμη την τυπική ορολογία, η έκφραση των ερωτήσεων σε αυτήν είναι πλέον δυνατή.
 5. Δόμηση αξιωμάτων και ορισμών. Οι ορισμοί και τα αξιώματα εκφράζουν τις σχέσεις μεταξύ των όρων της οντολογίας και τους περιορισμούς που τους διέπουν. Ορίζονται σαν λογικές προτάσεις πρώτης τάξεως. Με τον τρόπο αυτό ορίζεται η σημασιολογία των όρων, αφού μια συλλογή όρων δε μπορεί να θεωρηθεί ότι αποτελεί από μόνη της μια οντολογία.
 6. Ο ορισμός αξιωμάτων εκτελείται σε κύκλους. Σε κάθε κύκλο, ελέγχεται η ικανότητα των υπάρχοντων αξιωμάτων να αναπαραστήσουν τις απαντήσεις στα κριτήρια πληρότητας. Αν αυτά είναι ανεπαρκή για την κάλυψη των απαντήσεων, τότε πρέπει να προστεθούν νέα. Αυτός ο κύκλος επαναλαμβάνεται μέχρι να ολοκληρωθεί η κάλυψη των κριτηρίων αποδοτικότητας.
 7. Έκφραση συνθηκών από τις οποίες εξαρτάται η πληρότητα. Έχοντας δομήσει φορμαλιστικά τα κριτήρια πληρότητας, πρέπει επίσης να εκφράσουμε τις υποθέσεις στις οποίες αυτά βασίζονται, και κάτω από τις οποίες η οντολογία είναι πλήρης.

3.3.6.1 Αξιολόγηση

1. Η μέθοδος διαμόρφωσης ερωτήσεων τις οποίες πρέπει ένα σύστημα γνώσης να καλύπτει είναι συνηθισμένη στην ανάπτυξη συστημάτων γνώσης. Η παρούσα μεθοδολογία χρησιμοποιεί αυτή τη μέθοδο, ωστόσο δεν κάνει σαφή

διάκριση των σταδίων ανάπτυξης του συστήματος. Έτσι, μπορούμε να πούμε ότι η αξιοποίηση της υπάρχουσας εμπειρίας από τα συστήματα γνώσης είναι περιορισμένη.

2. Η μεθοδολογία δεν είναι λεπτομερής. Τόσο οι δραστηριότητες όσο και οι τεχνικές για την ολοκλήρωση τους περιγράφονται πολύ συνοπτικά.
3. Για την αναπαράσταση γνώσης πρέπει να χρησιμοποιηθεί κάποιος φορμαλισμός βασισμένος στη λογική.
4. Τα σενάρια χρήσης της οντολογίας καθορίζονται στα πρώτα στάδια ανάπτυξης, επομένως χρησιμοποιείται στρατηγική συγκεκριμένων χρήσεων.
5. Η αναγνώριση εννοιών γίνεται με τη μέθοδο middle-out.
6. Η μεθοδολογία δεν προτείνει συγκεκριμένο μοντέλο κύκλου ζωής, ούτε υπάρχει κάποια σαφής αναφορά σε κάποιο μοντέλο. Ωστόσο, η ακολουθία εκτέλεσης των δραστηριοτήτων περιγράφεται, ενώ υπάρχει και αναφορά στην επέκταση υπάρχουσας οντολογίας, μια διαδικασία η οποία επίσης ξεκινά με σύλληψη σεναρίων. Στην τελευταία περίπτωση, δεν διευκρινίζεται κατά πόσο η επέκταση μπορεί να περιλάβει και αλλαγές στους ήδη υφιστάμενους ορισμούς που περιέχει η οντολογία. Έτσι δεν μπορεί να καθοριστεί με σαφήνεια το κατά πόσο το μοντέλο του κύκλου ζωής βασίζεται σε εξελισσόμενα πρωτότυπα (evolving prototypes) ή είναι απλά ένα αυξητικό (incremental) μοντέλο.
7. Οι διαφορές με το πρότυπο 1074 είναι σημαντικές. Η μεθοδολογία καλύπτει μόνο τις κεντρικές δραστηριότητες (προδιαγραφές, ανάπτυξη, κάποιες υποστηρικτικές διαδικασίες), χωρίς κάλυψη των διαδικασιών που προηγούνται και έπονται της ανάπτυξης, ή των διαδικασιών διαχείρισης.
8. Δεν υπάρχει λεπτομερής ανάλυση χρησιμοποιούμενων τεχνικών.
9. Η μεθοδολογία έχει χρησιμοποιηθεί για την ανάπτυξη περίπλοκων οντολογιών στα πλαίσια του TOVE (Toronto Visual Enterprise), οι οποίες

ωστόσο ανήκουν όλες στην ίδια περιοχή. Αυτές οι οντολογίες έχουν αξιοποιηθεί στα πλαίσια δύο διαφορετικών εφαρμογών, του Enterprise Design Workbench και του Integrated Supply Chain Management. Η πρώτη βασίζεται σε ένα ολοκληρωμένο περιβάλλον ανάλυσης εταιρικών οργανωτικών δομών, ενώ η δεύτερη αποσκοπεί στην οργάνωση της εφοδιαστικής αλυσίδας, η οποία αντιμετωπίζεται ως ένα σύνολο από πράκτορες, κάθε ένας από τους οποίους επιτελεί μία ή περισσότερες λειτουργίες σε συνεργασία με τους υπόλοιπους. Η εφαρμογή αυτή βρίσκεται υπό εξέλιξη (<http://www.ie.utoronto.ca/EIL>).

10. Πληροφορίες για εργαλεία που υποστηρίζουν τη μεθοδολογία TOVE δεν είναι διαθέσιμες στη βιβλιογραφία.

3.3.7 KAKTUS

Η παρούσα μεθοδολογία προέκυψε ως ένα από τα αποτελέσματα του έργου KAKTUS του προγράμματος Esprit [KAKTUS, 1996], ένας από τους στόχους του οποίου ήταν μια μελέτη σκοπιμότητας της επαναχρησιμοποίησης γνώσης στα πλαίσια μεγάλων τεχνολογικών συστημάτων, και η ικανότητα της οντολογίας να την υποστηρίξει. Στο KAKTUS η οντολογία χρησιμοποιείται για να οδηγήσει την ανάπτυξη κάποιας εφαρμογής. Έτσι, όταν πρέπει να σχεδιαστεί κάποια εφαρμογή, αναπτύσσεται η αντίστοιχη οντολογία που κωδικοποιεί την απαραίτητη γνώση. Αυτή μπορεί να επαναχρησιμοποιεί ήδη υπάρχουσες οντολογίες και μπορεί επίσης στο μέλλον να ενσωματωθεί σε οντολογίες για μεταγενέστερες εφαρμογές. Έτσι, η ανάπτυξη της οντολογίας ακολουθεί την παρακάτω διαδικασία:

1. Προδιαγραφή της εφαρμογής. Αυτή είναι απαραίτητη για τον καθορισμό του πλαισίου για την ανάπτυξη της οντολογίας.
2. Προκαταρκτικό σχέδιο των γενικών κατηγοριών της οντολογίας. Αυτό πραγματοποιείται ομαδοποιώντας τους όρους που χρησιμοποιούνται στην

προδιαγραφή εφαρμογής παραπάνω. Στο στάδιο αυτό περιλαμβάνεται και η διερεύνηση υπάρχουσών οντολογιών, για πιθανή επαναχρησιμοποίηση ή επέκταση στα πλαίσια της τρέχουσας εφαρμογής.

3. Αναλυτικό σχέδιο και δόμηση της οντολογίας. Η δόμηση της οντολογίας πραγματοποιείται με στόχο να ελαχιστοποιηθεί η συσχέτιση ανάμεσα σε διαφορετικά τμήματα της και να μεγιστοποιηθεί η ομοιογένεια.

3.3.7.1 Αξιολόγηση

1. Η μεθοδολογία ακολουθεί την παραδοσιακή μέθοδο ανάπτυξης συστημάτων βασισμένων σε γνώση. Η ανάπτυξη της οντολογίας τοποθετείται στο γενικότερο πλαίσιο της ανάπτυξης ενός συστήματος βασισμένο στη γνώση.
2. Η μεθοδολογία δίνεται σε πολύ αδρές γραμμές, χωρίς καθόλου λεπτομέρεια.
3. Δεν υπάρχει πρόταση για τον τρόπο αναπαράστασης γνώσης.
4. Η ανάπτυξη οντολογιών τοποθετείται στο πλαίσιο της ανάπτυξης μιας συγκεκριμένης εφαρμογής. Ωστόσο, η οντολογία δύναται να επαναχρησιμοποιηθεί από άλλες εφαρμογές στη συνέχεια, με αποτέλεσμα τη διεύρυνση και βελτιστοποίηση της. Σταδιακά η οντολογία γενικεύεται έτσι, ωστόσο η ανάπτυξη της παραμένει βασισμένη στην εφαρμογή.
5. Η αναγνώριση εννοιών γίνεται με μέθοδο top-down, από το γενικό προς το ειδικό.
6. Δεν υπάρχει σαφής κύκλος ζωής της οντολογίας, ωστόσο συνάγεται ότι αυτή θα πρέπει να ακολουθεί τον κύκλο ζωής της εφαρμογής.
7. Η μεθοδολογία αποκλίνει από το πρότυπο IEEE 1074, αφού καθορίζει μόνο τις βασικές διαδικασίες ανάπτυξης και μάλιστα με λίγη λεπτομέρεια.
8. Δεν δίνονται συγκεκριμένες τεχνικές.
9. Η μεθοδολογία χρησιμοποιήθηκε για την ανάπτυξη τριών οντολογιών και ισάριθμων εφαρμογών στην περιοχή των ηλεκτρικών δικτύων. Οι εφαρμογές

αφορούν τη διάγνωση βλαβών στο δίκτυο και την αναδιαμόρφωση και έλεγχο του.

10. Πληροφορίες για εργαλεία που υποστηρίζουν τη μεθοδολογία δεν είναι διαθέσιμες στη βιβλιογραφία.

3.3.8 METHONTOLOGY

Η Methontology αναπτύχθηκε στο Πολυτεχνείο της Μαδρίτης [Fernandez, 1997], [Gomez Perez, 1998], [Fernandez Lopez, 1999]. Είναι μια ολοκληρωμένη μεθοδολογία που αποσκοπεί στην κατασκευή οντολογιών και περιλαμβάνει τον καθορισμό της διαδικασίας ανάπτυξης της οντολογίας, ένα μοντέλο κύκλου ζωής βασισμένο σε εξελισσόμενα πρωτότυπα (evolving prototypes) και τεχνικές για την ολοκλήρωση κάθε προβλεπόμενης δραστηριότητας. Στο Πολυτεχνείο της Μαδρίτης έχουν επίσης αναπτυχθεί διάφορα εργαλεία που υποβοηθούν την εφαρμογή αυτής της μεθοδολογίας.

Η διαδικασία ανάπτυξης οντολογίας καθορίζει τις δραστηριότητες που επιτελούνται όταν κατασκευάζεται μια οντολογία. Είναι κρίσιμη η αναγνώριση αυτών των δραστηριοτήτων εκ των προτέρων, ιδιαίτερα όταν η ανάπτυξη πρόκειται να γίνει παράλληλα από διακριτές ομάδες ή άτομα, με μειωμένη δυνατότητα επικοινωνίας (π.χ. λόγω γεωγραφικής απόστασης). Η διαδικασία αυτή περιλαμβάνει τις ακόλουθες δραστηριότητες, χωρισμένες σε τρεις κατηγορίες:

1. Δραστηριότητες διαχείρισης έργου
 - a. Σχεδιασμός έργου
 - b. Έλεγχος
 - c. Διασφάλιση ποιότητας
2. Δραστηριότητες ανάπτυξης
 - a. Προδιαγραφή

- b. Σύλληψη εννοιών
- c. Τυποποίηση
- d. Υλοποίηση
- e. Συντήρηση

3. Υποστηρικτικές δραστηριότητες

- a. Απόκτηση γνώσης
- b. Ολοκλήρωση
- c. Αποτίμηση
- d. Παραμετροποίηση

Οι δραστηριότητες διαχείρισης έργου περιλαμβάνουν το σχεδιασμό και έλεγχο του έργου, και τη διασφάλιση ποιότητας. Ο σχεδιασμός αναφέρεται στην αναγνώριση και οργάνωση των εργασιών που πρέπει να εκτελεστούν, στη δέσμευση πόρων και στον προσδιορισμό του αντίστοιχου χρονοδιαγράμματος. Ο έλεγχος συνίσταται στην παρακολούθηση της πορείας των εργασιών, σύγκριση με το σχέδιο και ενημέρωση του τελευταίου όταν χρειάζεται. Τέλος, η διασφάλιση ποιότητας στοχεύει στην εξασφάλιση των προϋποθέσεων που απαιτούνται για την εξασφάλιση υψηλού επιπέδου ποιότητας σε όλα τα προϊόντα του έργου (οντολογίες, λογισμικό, τεκμηρίωση) και στην οργάνωση των απαραίτητων ελέγχων και διαδικασιών αντιμετώπισης σφαλμάτων για την εξακρίβωση της ποιότητας. Αυτές οι δραστηριότητες είναι γενικές σε όλα τα έργα πληροφορικής, έτσι η μεθοδολογία δεν τις αναλύει διεξοδικά.

Οι δραστηριότητες ανάπτυξης περιλαμβάνουν την προδιαγραφή, τη σύλληψη εννοιών, την τυποποίηση τους και την υλοποίηση της οντολογίας. Η προδιαγραφή καθορίζει το σκοπό που καλείται να εξυπηρετήσει η υπό ανάπτυξη οντολογία, τις προβλεπόμενες χρήσεις της και τους τελικούς χρήστες. Η σύλληψη εννοιών συνίσταται στην οργάνωση της γνώσης της περιοχής

εφαρμογής σε ένα άτυπο μοντέλο. Η τυποποίηση είναι η διαδικασία μετατροπής αυτού του μοντέλου σε ένα αυτοσυνεπές, φορμαλιστικό μοντέλο. Τέλος, η υλοποίηση είναι η διαδικασία έκφρασης αυτού του μοντέλου στην επιλεγμένη για το έργο γλώσσα αναπαράστασης γνώσης. Η συντήρηση είναι η διαδικασία ενημέρωσης της οντολογίας καθώς οι προδιαγραφές εξελίσσονται και αντιμετώπισης τυχόν λαθών που εμφανίζονται. Στη βιβλιογραφία [Fernandez Lopez, 1999] παρέχονται τεχνικές για την ολοκλήρωση της προδιαγραφής, της σύλληψης εννοιών και της υλοποίησης, όχι όμως για την τυποποίηση και την συντήρηση.

Οι υποστηρικτικές δραστηριότητες εξελίσσονται παράλληλα με τις δραστηριότητες ανάπτυξης. Περιλαμβάνουν την πρόσκτηση γνώσης, την αποτίμηση, την ολοκλήρωση και την τεκμηρίωση. Η πρόσκτηση γνώσης είναι η διαδικασία συλλογής γνώσης για την περιοχή ενδιαφέροντος από διάφορες πηγές [Fernandez Lopez, 1999], [Gomez Perez, 1999]. Η αποτίμηση αποτελεί μια τεχνική αξιολόγηση των προϊόντων του έργου (οντολογιών, λογισμικού, τεκμηρίωσης) σε σχέση με το πλαίσιο αναφοράς που έχει καθοριστεί στο σχεδιασμό και την προδιαγραφή του έργου. Πραγματοποιείται σε κάθε φάση και ιδιαίτερα στο τέλος του κύκλου ζωής [Gomez Perez, 1995]. Η ολοκλήρωση είναι μια διαδικασία που εφαρμόζεται όταν η ανάπτυξη μιας οντολογίας βασίζεται σε άλλες, υφιστάμενες. Η τεκμηρίωση είναι μια ομάδα δραστηριοτήτων που παράγουν αναλυτικές γραπτές περιγραφές των προϊόντων και ενδιάμεσων προϊόντων του έργου (οντολογίες, προδιαγραφές, εργαλεία λογισμικού, τεχνικό σχέδιο, εγχειρίδια χρήσης, εκπαίδευσης, παραμετροποίησης και διαχείρισης κλπ). Η παραμετροποίηση και διαμόρφωση περιλαμβάνει την προσαρμογή του παραγόμενου λογισμικού στο περιβάλλον του, καθώς και την παρακολούθηση των αλλαγών στις παραμέτρους και τον κώδικα. Έτσι περιλαμβάνει και την καταγραφή όλων των αλλαγών στον κώδικα, την τεκμηρίωση και τις λοιπές παραμέτρους.

Ο κύκλος ζωής της μεθοδολογίας καθορίζει ένα σύνολο από στάδια, από τα οποία διέρχεται η οντολογία κατά τη διάρκεια της ζωής της. Σε κάθε στάδιο πραγματοποιούνται κάποιες από τις παραπάνω δραστηριότητες. Η Methontology προτείνει ένα κύκλο ζωής βασισμένο σε εξελισσόμενα πρωτότυπα [Fernandez, 1997], όπου η ανάπτυξη κάθε πρωτοτύπου ξεκινά με την προδιαγραφή και την πρόσκτηση γνώσης. Οι δύο αυτές δραστηριότητες πραγματοποιούνται παράλληλα, καθώς δεν έχουν άμεση εξάρτηση μεταξύ τους. Στην συνέχεια ακολουθεί η δημιουργία ενός εννοιολογικού μοντέλου στη φάση σύλληψης εννοιών. Στη φάση αυτή, οι μεμονωμένες ψηφίδες γνώσης που έχουν συλλεχθεί ολοκληρώνονται σε ένα συνεκτικό σύνολο [Fernandez Lopez, 1999]. Η τυποποίηση και η υλοποίηση ακολουθούν.

Οι υποστηρικτικές διαδικασίες εξελίσσονται παράλληλα με την ανάπτυξη και σε όλη τη διάρκεια της. Ωστόσο πρέπει να σημειωθεί ότι τόσο η αξιολόγηση όσο και η ολοκλήρωση παρουσιάζουν τη μεγαλύτερη ένταση εργασίας στη φάση της σύλληψης εννοιών, όπου οι πιθανές αλληλεπιδράσεις με άλλες οντολογίες πρέπει να καθοριστούν και η αξιολόγηση του παραγόμενου εννοιολογικού μοντέλου είναι απαραίτητη, αλλά αρκετά δύσκολη καθώς δε μπορεί να υποστηριχθεί από εργαλεία.

3.3.8.1 Αξιολόγηση

1. Η Methontology βασίζεται στην προϋπάρχουσα, αν και όχι ιδιαίτερα διαδεδομένη, μεθοδολογία ανάπτυξης συστημάτων γνώσης IDEAL [Gomez Perez, 1995].
2. Το μεγαλύτερο μέρος της μεθοδολογίας είναι αναλυτικά τεκμηριωμένο. Υπάρχουν όμως δραστηριότητες και τεχνικές που δεν καλύπτονται αναλυτικά.
3. Για την αναπαράσταση γνώσης, η Methontology δε θέτει περιορισμούς.

4. Η ανάπτυξη της οντολογίας είναι γενικά ανεξάρτητη από τις χρήσεις. Κάποια εξάρτηση εισέρχεται αναπόφευκτα μέσω της διαδικασίας αποτίμησης.
5. Η αναγνώριση εννοιών ακολουθεί το πρότυπο middle-out.
6. Ο κύκλος ζωής που προτείνεται είναι μια μορφή εξελισσόμενων πρωτοτύπων.
7. Η μεθοδολογία καλύπτει σε αρκετά μεγάλο βαθμό το πρότυπο IEEE 1074. Υπάρχουν όμως ομάδες δραστηριοτήτων που δεν καλύπτονται, ή που προβλέπονται αλλά δεν περιγράφονται αναλυτικά. Δεν καλύπτονται οι διαδικασίες προετοιμασίας ανάπτυξης, η εγκατάσταση, υποστήριξη και εκπαίδευση. Επίσης, δεν περιγράφονται οι διαδικασίες διαχείρισης έργου, προετοιμασίας έργου, ελέγχου και διασφάλισης ποιότητας. Τέλος, πρέπει να σημειωθεί ότι τα όρια μεταξύ διαφορετικών δραστηριοτήτων δεν είναι πάντα ξεκάθαρα. Για παράδειγμα, η αποτίμηση αποτελεί μια υποστηρικτική διαδικασία, ωστόσο η σημασία της για τη διασφάλιση ποιότητας είναι προφανής. Επίσης, η τυποποίηση είναι προαιρετικού χαρακτήρα, αφού μπορεί κανείς να μεταβεί κατ' ευθείαν από τη σύλληψη εννοιών στην υλοποίηση, και μάλιστα η απ' ευθείας μετάβαση υποστηρίζεται από τα παρεχόμενα εργαλεία. Επομένως η σκοπιμότητα της τυποποίησης είναι από τη μία πλευρά αμφισβητήσιμη. Από την άλλη πλευρά, δεν είναι σαφές το με ποια δραστηριότητα υλοποιείται η θεμελιώδης για το IEEE διαδικασία του σχεδιασμού, και ποιο προϊόν είναι το σχέδιο στο οποίο βασίζεται η υλοποίηση.
8. Τεχνικές παρέχονται για όλες τις διαδικασίες ανάπτυξης, και αρκετές διαδικασίες υποστήριξης και διαχείρισης.
9. Η μεθοδολογία έχει χρησιμοποιηθεί για την ανάπτυξη ενός σημαντικού αριθμού οντολογιών και εφαρμογών, σε διαφορετικές περιοχές. Σε αυτές περιλαμβάνεται η οντολογία Chemicals [Gomez Perez, 1996], [Fernandez

Lopez, 1999] που περιγράφει χημικά στοιχεία και κρυσταλλικές δομές. Επίσης η Reference-Ontology [Arpírez, 1998], η οποία είναι μια οντολογία για οντολογίες. Συνδέει και συσχετίζει κάτω από ένα ενιαίο μετα-μοντέλο διάφορες υπάρχουσες οντολογίες. Με βάση αυτές έχουν αναπτυχθεί τρεις σημαντικές εφαρμογές: οι (Onto)²Agent και Chemical OntoAgent [Arpírez, 1998] παρέχουν υπηρεσίες διαμεσολάβησης στο WWW. Τέλος, το Ontogeneration [Aguado, 1998] είναι ένα σύστημα που συνδυάζει την οντολογία Chemicals με τη γλωσσολογική οντολογία GUM [Bateman, 1995], για να δημιουργήσει περιγραφές σε φυσική γλώσσα στην περιοχή της χημείας.

10. Η μεθοδολογία υποστηρίζεται από το εργαλείο ODE [Fernandez Lopez, 1999], [Blazquez, 1998]. Πρόκειται για ένα γραφικό εργαλείο που υποστηρίζει τη δημιουργία του εννοιολογικού μοντέλου και μπορεί να παράγει κώδικα σε αρκετές γλώσσες αναπαράστασης γνώσης (Ontolingua, Flogic, Carin).

3.3.9 SENSUS

Η οντολογία SENSUS αναπτύχθηκε στο Information Sciences Institute και προορίζεται για χρήση σε εφαρμογές φυσικής γλώσσας, ιδιαίτερα αυτόματης μετάφρασης. Αποσκοπεί έτσι να παράσχει μια ευρεία εννοιολογική δομή για την κωδικοποίηση πολυγλωσσικής ορολογίας [Knight, 1994] [Knight, 1995]. Το περιεχόμενο της αντλήθηκε φιλτράροντας και συνδυάζοντας γνώση από διάφορες πηγές. Αυτή η διαδικασία ξεκίνησε συνδυάζοντας την οντολογία PENMAN Upper Model [Bateman, 1994] με την ONTOS [Nirenburg, 1991], δύο γλωσσολογικές οντολογίες πολύ υψηλού επιπέδου που αποτέλεσαν την κορυφή της SENSUS. Σε αυτές προστέθηκαν σημασιολογικές κατηγορίες που αντλήθηκαν από λεξικά και κωδικοποιήθηκαν με το χέρι. Στη συνέχεια, πραγματοποιήθηκε ολοκλήρωση του WordNet [Fellbaum, 1998], επίσης με το χέρι. Στο σημείο αυτό αναπτύχθηκε εργαλείο αυτοματοποίησης της διαδικασίας

ενσωμάτωσης, με το οποίο προστέθηκε στο WordNet αγγλική ορολογία με ενσωμάτωση αντίστοιχου λεξικού. Τέλος, προκειμένου να υποστηριχθεί η αυτόματη μετάφραση, στην υπάρχουσα δομή προστέθηκαν ισπανικά και ιαπωνικά λήμματα από τα λεξικά Collins Spanish/English Dictionary και Kenkyusha Japanese/English Dictionary [Swartout, 1997].

Τελικά η οντολογία SENSUS περιέχει πλέον των 50.000 εννοιών, οργανωμένων ιεραρχικά ανάλογα με το επίπεδο αφαιρετικότητας τους. Η οντολογία περιέχει όρους υψηλού και μέσου επιπέδου αφαιρετικότητας, είναι δηλαδή κατά κανόνα αφηρημένη. Σε γενικές γραμμές, δεν περιέχει όρους που αναφέρονται σε συγκεκριμένες περιοχές ή εφαρμογές. Τέτοιοι όροι συνδέονται με τη SENSUS οδηγώντας στη δημιουργία οντολογιών για συγκεκριμένες περιοχές, ενώ οι όροι της SENSUS που δεν αφορούν τη συγκεκριμένη περιοχή απορρίπτονται.

Έτσι, η SENSUS αποτελεί μια κεντρική αφηρημένη οντολογία, με στόχο την επέκταση και συγκεκριμενοποίηση της για τη δημιουργία οντολογιών περιοχής για συγκεκριμένες εφαρμογές. Για τη δημιουργία τέτοιων οντολογιών περιοχής, οι δημιουργοί της SENSUS προτείνουν [Swartout, 1997] συγκεκριμένη μεθοδολογία που αποτελείται από τα ακόλουθα βήματα:

1. Μια σειρά όρων λαμβάνεται ως βάση (seed).
2. Αυτοί οι όροι βάσης συνδέονται με το χέρι με τη SENSUS.
3. Στη συνέχεια συνδέονται οι έννοιες στη διαδρομή από τους όρους βάσης μέχρι την κορυφή της οντολογίας SENSUS.
4. Οι όροι που θα μπορούσαν να είναι σχετικοί με στην περιοχή και δεν έχουν εμφανιστεί ακόμα προστίθενται.
5. Τελικά, για εκείνους τους κόμβους που διασχίζονται από έναν μεγάλο αριθμό τροχιών, είναι δυνατό να προστεθεί ολόκληρο το υποδένδρο κάτω από τον κόμβο, με βάση την ιδέα ότι εάν πολλοί από τους κόμβους του υποδένδρου

έχουν βρεθεί για να είναι σχετικοί, είναι πιθανό να είναι σχετικοί και οι υπόλοιποι. Αυτό το βήμα γίνεται με το χέρι, δεδομένου ότι απαιτείται κάποια κατανόηση της περιοχής για να ληφθεί η απόφαση. Προφανώς, οι κόμβοι πολύ υψηλού επιπέδου στην οντολογία θα έχουν πάντα πολλές τροχιές διαμέσου τους, αλλά βέβαια δεν είναι πάντα σκόπιμο να περιληφθούν ολόκληρα τα subtrees κάτω από αυτούς τους κόμβους.

3.3.9.1 Αξιολόγηση

1. Η μεθοδολογία της SENSUS διαφοροποιείται πλήρως από τις παραδοσιακές μεθόδους, αφού βασίζεται στην προσθήκη όρων σε μία υπάρχουσα οντολογία, η οποία στη συνέχεια περικόπεται με την απόρριψη των άχρηστων όρων.
2. Η μεθοδολογία δεν έχει παρουσιαστεί σε μεγάλη λεπτομέρεια.
3. Για την αναπαράσταση γνώσης, η SENSUS χρησιμοποιεί σημασιολογικά δίκτυα (semantic networks).
4. Κατά την ανάπτυξη της οντολογίας υπάρχει επιρροή των προβλεπόμενων εφαρμογών, αφού οι όροι βάσης επιλέγονται έχοντας υπόψη μία ή περισσότερες εφαρμογές.
5. Η αναγνώριση εννοιών γίνεται ουσιαστικά από κάτω προς τα πάνω. Η διαδικασία αρχίζει από τους όρους βάσης, οι οποίοι βρίσκονται στο κατώτερο άκρο της οντολογίας. Στη συνέχεια η αναγνώριση συνεχίζεται προς τα πάνω, περιλαμβάνοντας τις τροχιές τους. Βέβαια, η διαδικασίες της απόρριψης και της περίληψης υποδένδρων ξεφεύγουν από αυτό το πλαίσιο, ωστόσο η μέθοδος παραμένει bottom-up στον πυρήνα της.
6. Η μεθοδολογία δεν εξετάζει την ανάπτυξη άλλων πρωτοτύπων Η δῶδφδηπέραν του πρώτου, ούτε γίνεται άλλη αναφορά στο μοντέλο κύκλου ζωής. Επομένως αυτό δεν είναι καθορισμένο.

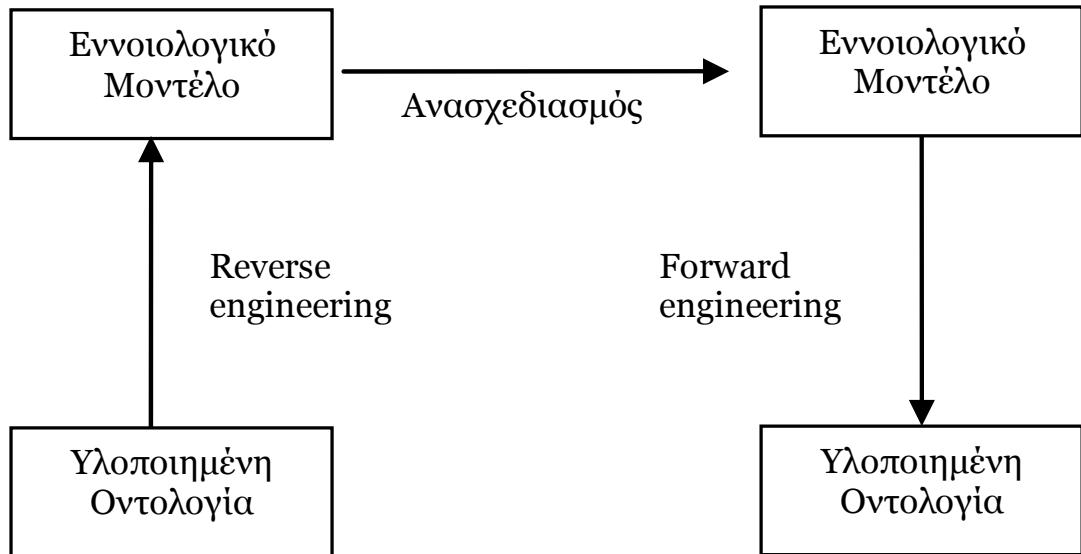
7. Η μεθοδολογία δεν καλύπτει το πρότυπο IEEE 1074-1995. Καθορίζονται μόνο οι διαδικασίες ανάπτυξης και παραλείπονται οι διαδικασίες που έπονται και ακολουθούν, καθώς και οι υποστηρικτικές διαδικασίες. Σημαντικότερη έλλειψη είναι η μη ύπαρξη διαδικασίας σχεδιασμού.
8. Προτεινόμενες τεχνικές για κάθε δραστηριότητα δεν δίνονται.
9. Η μεθοδολογία έχει χρησιμοποιηθεί για την ανάπτυξη μιας οντολογίας στην περιοχή του σχεδιασμού αεροπορικών πολεμικών επιχειρήσεων [Valente, 1999]. Πρόκειται για μια εκτεταμένη οντολογία περιοχής, που περιλαμβάνει επιμέρους οντολογίες οπλισμού, καυσίμων, συστημάτων κλπ. Στη βάση αυτής της οντολογίας αναπτύχθηκαν εφαρμογές σχεδιασμού επιχειρήσεων και υποστήριξης αποφάσεων, με κυριότερη την Strategy Development Assistant, μια εφαρμογή υποστήριξης αποφάσεων και δομημένου σχεδιασμού επιχειρήσεων.
10. Η δημιουργία οντολογιών περιοχής και η σύνδεση τους με τη SENSUS υποστηρίζεται από το εργαλείο Ontosaurus [Swartout, 1997].

3.4 ΜΕΘΟΔΟΛΟΓΙΕΣ ΑΝΑΣΧΕΔΙΑΣΜΟΥ

ΟΝΤΟΛΟΓΙΩΝ

Ο ανασχεδιασμός οντολογιών (ontology re-engineering) είναι μια διαδικασία που αποσκοπεί στην ανασύνθεση του εννοιολογικού μοντέλου μιας ήδη υλοποιημένης οντολογίας, και αντιστοίχιση του σε ένα νέο εννοιολογικό μοντέλο, το οποίο στη συνέχεια υλοποιείται εκ νέου. Μια μεθοδολογία ανασχεδιασμού οντολογιών έχει παρουσιαστεί στο [Gomez Perez, 1999] και αντιμετωπίζει το πρόβλημα αυτό σαν ειδική περίπτωση του προβλήματος ανασχεδιασμού λογισμικού. Στη συνέχεια περιγράφουμε συνοπτικά αυτή τη μεθοδολογία.

Η διαδικασία ανασχεδιασμού οντολογιών αναπαρίσταται συνοπτικά στο Σχήμα 4, η οποία προσαρμόζει τη μεθοδολογία ανασχεδιασμού λογισμικού του Chikofsky [Chikofsky, 1990] στην περιοχή των οντολογιών. Στο σχήμα φαίνονται οι τρεις βασικές διεργασίες που λαμβάνουν χώρα. Αυτές είναι το reverse engineering, ο ανασχεδιασμός, και τέλος το forward engineering και περιγράφονται στη συνέχεια.



Σχήμα 4: Η διαδικασία ανασχεδιασμού οντολογιών

- Reverse engineering: Ο σκοπός αυτής της διεργασίας είναι η παραγωγή ενός εννοιολογικού μοντέλου της οντολογίας, ξεκινώντας από τον κώδικα με τον οποίο είναι υλοποιημένη. Για το σκοπό αυτό, αποτυπώνονται οι ταξινομίες και τυχόν άλλες ιεραρχικές σχέσεις ανάμεσα σε έννοιες, καθώς και οι σχέσεις μεταξύ εννοιών, στην ίδια ή διαφορετική ιεραρχία. Οι συναρτήσεις και αξιώματα της οντολογίας πρέπει επίσης να καταγραφούν.
- Ανασχεδιασμός (ή αναδόμηση): Ο σκοπός του ανασχεδιασμού είναι η τροποποίηση και επαύξηση του εννοιολογικού μοντέλου, έτσι ώστε να ικανοποιήσει τις νέες χρήσεις και εφαρμογές στις οποίες απευθύνεται. Αυτό βέβαια σημαίνει ότι δεν υπάρχει τρόπος να εγγυηθεί κανείς ότι η νέα αναδομημένη οντολογία θα είναι πλήρως συμβατή με την παλαιά. Σε

ορισμένες περιπτώσεις αυτό ενδέχεται να μην είναι καν επιθυμητό, ιδιαίτερα όταν δεν υπάρχουν νέες εφαρμογές αλλά ο σκοπός του ανασχεδιασμού είναι η ενημέρωση, η επικαιροποίηση και η λεπτομερέστερη τεκμηρίωση της οντολογίας. Ωστόσο, σε περιπτώσεις που ο ανασχεδιασμός προορίζεται για να καλύψει νέες ανάγκες, χωρίς να αλλάξει η συμπεριφορά της οντολογίας στις υφιστάμενες εφαρμογές και χρήσεις της, τότε απαιτείται λεπτομερής και διεξοδική σειρά ελέγχων και δοκιμών μετά την ολοκλήρωση του ανασχεδιασμού. Αυτό το πρόβλημα αντιμετωπίζεται, ως ένα βαθμό, με την συμπερίληψη μιας διαδικασίας αξιολόγησης της υφιστάμενης οντολογίας στο στάδιο της αναδόμησης. Στο ίδιο πλαίσιο πραγματοποιείται και ανάλυση της ορθότητας και πληρότητας της οντολογίας. Στη συνέχεια πραγματοποιείται μια διαδικασία σύνθεσης, κατά την οποία πραγματοποιούνται και καταγράφονται οι επιθυμητές αλλαγές στην οντολογία.

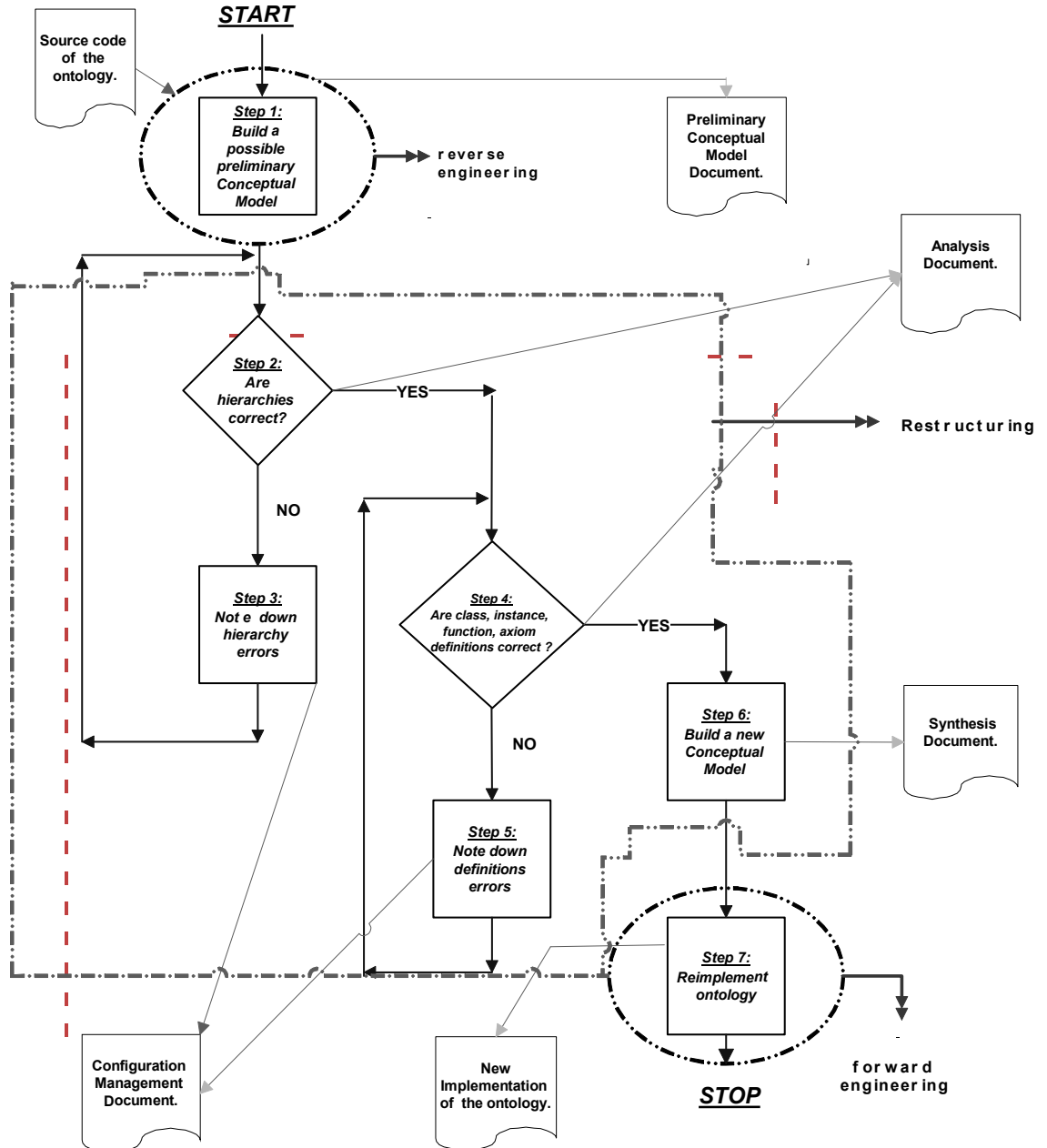
- **Forward Engineering:** Αυτή είναι η διεργασία που παράγει τον υλοποιημένο κώδικα της οντολογίας, με βάση το εννοιολογικό μοντέλο που έχει προκύψει από την αναδόμηση. Η διεργασία αυτή εκτελείται όπως ακριβώς η κανονική διαδικασία υλοποίησης μιας οντολογίας εκ νέου.

Η προτεινόμενη μεθοδολογία είναι μια αρχική προσπάθεια για την περιγραφή των παραπάνω βημάτων, αν και επιδέχεται βελτίωση με λεπτομερέστερη έρευνα βασισμένη σε πιο περίπλοκες οντολογίες. Στο Σχήμα 5 φαίνονται αναλυτικότερα τα βήματα που προβλέπει αυτή η μεθοδολογία. Προκειμένου να ενισχύσει την δυνατότητα επαναχρησιμοποίησης της παραγόμενης οντολογίας, η μεθοδολογία περιλαμβάνει επίσης κάποιες οδηγίες και κριτήρια διασφάλισης αυτής της δυνατότητας, τα οποία όμως δεν είναι πλήρη. Ένα ακόμα ανοικτό πρόβλημα αφορά τη σχέση της οντολογίας υπό ανασχεδιασμό με άλλες οντολογίες υψηλότερου επιπέδου.

Τέλος, για να μπορούν να ελεγχθούν οι πραγματοποιούμενες αλλαγές, διαδικασίες διαχείρισης αλλαγών (change/configuration management) πρέπει να

εφαρμόζονται σε όλη τη διάρκεια του ανασχεδιασμού. Ο στόχος της διαχείρισης αλλαγών πρέπει να είναι η αναλυτική καταγραφή όχι μόνο των αλλαγών καθεαυτών, αλλά και των αιτίων που τις προκαλούν (με αναφορά στις αντίστοιχες προδιαγραφές) και των συνεπειών που αυτές θα επιφέρουν στις τελικές εφαρμογές. Η διαχείριση αλλαγών μπορεί έτσι να βοηθήσει δραστικά στην αποφυγή λανθασμένων αποφάσεων, αφού δίνει τη δυνατότητα σύγκρισης εναλλακτικών τρόπων να επιτευχθεί μια αλλαγή και των συνεπειών που αυτοί θα επιφέρουν. Ωστόσο η βασική της χρησιμότητα είναι να δώσει τη δυνατότητα στους τελικούς χρήστες να επιλέξουν την κατάλληλη έκδοση της οντολογίας για την εφαρμογή τους, συγκρίνοντας τις αλλαγές στη συμπεριφορά της μεταξύ διαφορετικών εκδόσεων.

Η διαδικασία διαχείρισης αλλαγών ξεκινά με μια αίτηση αλλαγής (change request), η οποία περιέχει μια συνοπτική περιγραφή της αλλαγής και την αναλυτική αιτιολόγηση της χρησιμότητας της. Η αίτηση αυτή συνήθως πηγάζει από τους τελικούς χρήστες. Στη συνέχεια, πραγματοποιείται αξιολόγηση της αναγκαιότητας της αλλαγής, εκτίμηση του κόστους της και της επίδρασης που ενδέχεται να έχει στη σταθερότητα των εξαρτώμενων συστημάτων και εφαρμογών. Τα αποτελέσματα αυτής της ανάλυσης εκτιμώνται συνολικά ώστε να εγκριθεί ή να απορριφθεί η αλλαγή. Στην περίπτωση που εγκριθεί, ακολουθεί η πραγματοποίηση της αλλαγής, οι δοκιμές ελέγχου, τυχόν διορθώσεις, και τέλος η τελική αποδοχή της.



Σχήμα 5: Τα στάδια του ανασχεδιασμού οντολογιών [Gomez Perez, 1999]

3.5 ΣΥΝΟΨΗ ΚΑΙ ΣΥΜΠΕΡΑΣΜΑΤΑ

Έχοντας παρουσιάσει και αναλύσει κάθε μεθοδολογία ξεχωριστά, έχει ενδιαφέρον να κάνουμε μια συνολική και συγκριτική θεώρηση τους, για διάφορους λόγους. Ο βασικότερος είναι να μπορέσουμε να δώσουμε κάποια

ποιοτική εκτίμηση του βαθμού ωριμότητας όχι κάποιας συγκεκριμένης μεθοδολογίας, αλλά του ερευνητικού πεδίου της ανάπτυξης οντολογιών γενικότερα. Για το σκοπό αυτό είναι ιδιαίτερα ενδιαφέρουσα η αντιπαράθεση με το πρότυπο ανάπτυξης λογισμικού IEEE 1074-1995. Επίσης, είναι σημαντικό να ανιχνεύσουμε την ύπαρξη κοινών στοιχείων μεταξύ των μεθοδολογιών, καθώς αυτά μπορεί να υποδεικνύουν μια κοινά αποδεκτή κατεύθυνση. Σε αυτό το πλαίσιο, μπορούμε να βγάλουμε τα ακόλουθα συμπεράσματα:

1. Καμία από τις μεθοδολογίες δεν είναι απόλυτα ώριμη σε σύγκριση με το πρότυπο IEEE 1074-1995. Ωστόσο, ως προς την ικανοποίηση του προτύπου μπορούμε να κατατάξουμε τις μεθοδολογίες ως εξής:
 - I. Η METHONTOLOGY είναι η ωριμότερη, αν και λείπουν οι προδιαγραφές των διαδικασιών που προηγούνται και έπονται της ανάπτυξης, ενώ λείπει λεπτομέρεια σε μερικές από τις προδιαγραφές διαδικασιών και τεχνικών. Επιπλέον, η μεθοδολογία αυτή υποστηρίζεται από το Foundation for Intelligent Physical Agents (FIPA).
 - II. Η μεθοδολογία Cys, η οποία έχει τις σημαντικές παραλείψεις στις διαδικασίες, στις δραστηριότητες και στις τεχνικές που περιγράφονται παραπάνω. Ο κύκλος ζωής επίσης δεν περιγράφεται λεπτομερώς.
 - III. Η μεθοδολογία TOVE δεν περιλαμβάνει καμία διαδικασία προετοιμασίας της ανάπτυξης, μετά την ανάπτυξη ή υποστήριξη. Ο κύκλος ζωής δεν εξετάζεται άμεσα, και μόνο έμμεσα μπορεί κανείς να εξαγάγει συμπεράσματα. Επιπλέον, αν και οντολογίες έχουν αναπτυχθεί με αυτήν την μεθοδολογία, και υπάρχουν επίσης εφαρμογές που χρησιμοποιούν αυτές τις οντολογίες, αυτές είναι περιορισμένες στην περιοχή της μοντελοποίησης επιχειρήσεων.
 - IV. Η μεθοδολογία της Enterprise Ontology έχει τις ίδιες παραλείψεις με την προηγούμενη και είναι λιγότερο λεπτομερής.

- V. Η μεθοδολογία SENSUS, η οποία, εκτός από τις ανεπάρκειες των ανωτέρω μεθοδολογιών, δεν εξετάζει καθόλου τον κύκλο ζωής.
- VI. Η μεθοδολογία KAKTUS, η οποία παρουσιάζει όλες τις παραπάνω παραλείψεις και επιπλέον δεν έχει δοκιμαστεί στην πράξη, παρά μόνο για μια μικρή οικογένεια εφαρμογών στον περιορισμένο τομέα των ηλεκτρικών δικτύων.
2. Οι μεθοδολογίες παρουσιάζουν μεγάλες μεταξύ τους αποκλίσεις. Αυτή τη στιγμή κάθε ομάδα εφαρμόζει τη μεθοδολογία της. Αυτό είναι χαρακτηριστικό δείγμα ότι καμία δεν έχει φθάσει στην ωριμότητα, έτσι ώστε να μπορεί να επικρατήσει έναντι των άλλων. Επομένως, απαιτείται σημαντική δουλειά στον τομέα της σύγκλισης και ενοποίησης αυτών των μεθοδολογιών ώστε να προκύψει μια σχετικά σταθερή και ανταγωνιστική μεθοδολογία. Μια πρώτη τέτοια προσπάθεια έχει παρουσιαστεί στο [Uschold, 1996], όπου παρουσιάζονται κάποιες σκέψεις και ιδέες για τη σύντηξη δύο μεθοδολογιών. Πρόκειται όμως για μια πολύ θεωρητική δουλειά, η οποία δεν εξελίχθηκε σε πρακτικά εφαρμόσιμη λύση.
 3. Οι μεθοδολογίες ενσωματώνουν σε διαφορετικό βαθμό προϋπάρχουσα γνώση, με αποκορύφωμα τη SENSUS. Ωστόσο η αξιοποίηση αυτής της γνώσης και η επαναχρησιμοποίηση της γίνεται δυσκολότερη καθώς αυξάνει ο όγκος της.
 4. Υπάρχουν δύο μεθοδολογίες που είναι εντελώς διαφορετικές από τις υπόλοιπες, η Cyc και η SENSUS. Αυτές προσεγγίζουν το πρόβλημα ανάπτυξης οντολογιών μέσω της δημιουργίας μιας τεράστιας οντολογίας υποβάθρου, η οποία στη συνέχεια περικόπτεται και συγκεκριμενοποιείται ελαφρά για κάθε εφαρμογή. Αυτές προσφέρουν το μεγάλο πλεονέκτημα της διαλειτουργικότητας (interoperability) μεταξύ συστημάτων, η οποία γίνεται δυνατή καθώς τα συστήματα βασίζονται στο ίδιο οντολογικό υπόστρωμα.

Τέλος, πρέπει να αναφέρουμε ότι το πρόβλημα του ανασχεδιασμού οντολογιών δεν έχει ερευνηθεί μέχρι στιγμής συστηματικά, παρόλο που η επαναχρησιμοποίηση οντολογιών ως έννοια έχει απασχολήσει έντονα τους σχεδιαστές μεθοδολογιών και γενικότερα τους ερευνητές του πεδίου ως έννοια. Έτσι, αν και οι μεθοδολογίες ανάπτυξης οδηγούνται σε σημαντικό βαθμό από το κριτήριο της επαναχρησιμοποίησης, δεν αντιμετωπίζουν άμεσα το πρόβλημα του ανασχεδιασμού. Καταγράψαμε παραπάνω μια μεμονωμένη προσπάθεια αντιμετώπισης του προβλήματος αυτού, η οποία όμως είναι μόνο ένα πρώτο βήμα και δεν έχει δοκιμαστεί εκτεταμένα στην πράξη.

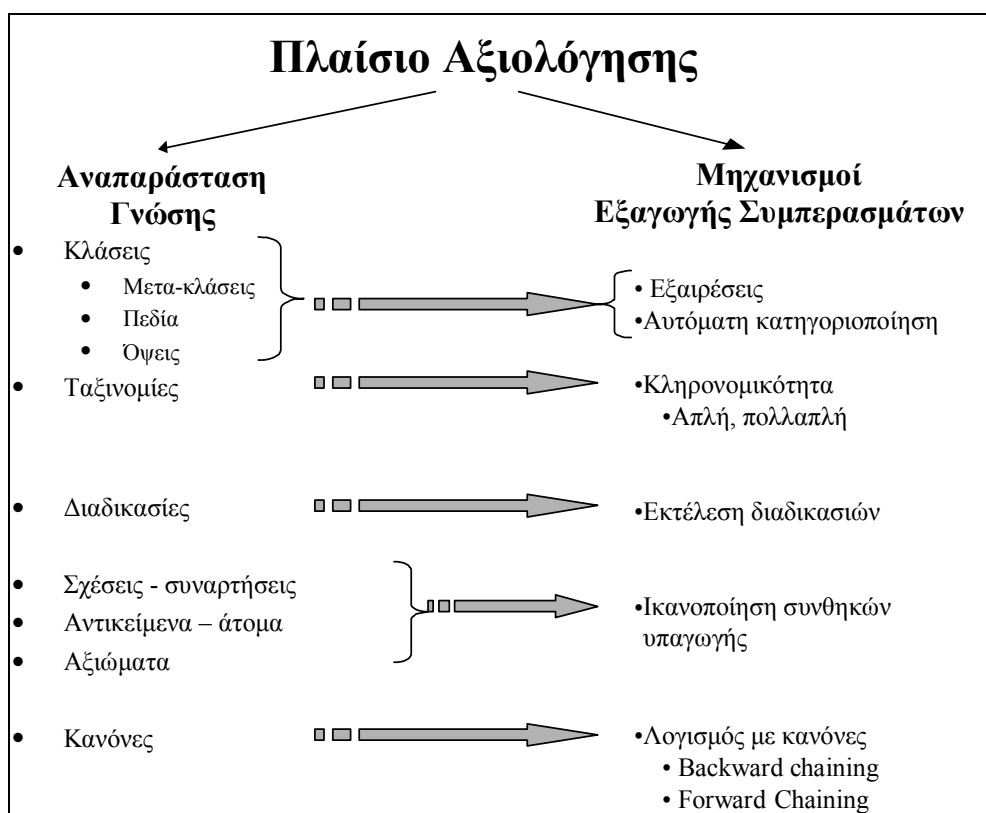
4 ΓΛΩΣΣΕΣ ΚΩΔΙΚΟΠΟΙΗΣΗΣ ΟΝΤΟΛΟΓΙΩΝ

Στο κεφάλαιο αυτό, εξετάζουμε τις πλέον σύγχρονες και καταξιωμένες γλώσσες περιγραφής οντολογιών. Ο σκοπός της εξέτασης είναι να αξιολογηθεί η ωριμότητα των διαφόρων γλωσσών και η δυνατότητα τους να υποστηρίξουν πραγματικές εφαρμογές, η εκφραστικότητα τους και η ευκολία χρήσης. Για να πραγματοποιήσουμε την αξιολόγηση, παρουσιάζουμε πρώτα ένα πλαίσιο κριτηρίων που θα χρησιμοποιηθούν για το σκοπό αυτό. Στη συνέχεια, αναλύουμε κάθε γλώσσα ως προς το πλαίσιο αυτό. Το κεφάλαιο κλείνει με μια συγκριτική αντιπαράθεση των γλωσσών και τα συμπεράσματα που εξάγονται από αυτή.

4.1 ΠΛΑΙΣΙΟ ΑΞΙΟΛΟΓΗΣΗΣ

Στην ανάλυση που ακολουθεί, επιλέγουμε να χρησιμοποιήσουμε το πλαίσιο αξιολόγησης που προτείνει η μεθοδολογία CommonKADS [Schreiber, 1999], το οποίο διαιρεί την αξιολόγηση σε δύο τομείς, αυτόν της αναπαράστασης γνώσης και αυτόν της εξαγωγής συμπερασμάτων. Στο Σχήμα 6 φαίνονται οι κύριες

διαστάσεις του πλαισίου αξιολόγησης και η σχέση μεταξύ των συνιστωσών αναπαράστασης γνώσης και αυτών της εξαγωγής συμπερασμάτων.



Σχήμα 6. Το πλαίσιο αξιολόγησης γλωσσών ανάπτυξης οντολογιών του CommonKADS

Η γνώση της περιοχής εφαρμογής είναι ο κύριος όγκος στατικής – ή αργά μεταβαλλόμενης – πληροφορίας της περιοχής [Schreiber, 1999]. Στην παράγραφο 3.1 παρουσιάσαμε τις βασικές έννοιες μοντελοποίησης που χρησιμοποιούνται για την αναπαράσταση γνώσης σε οντολογίες: κλάσεις (classes), σχέσεις (relations), συναρτήσεις (functions), αξιώματα (axioms) και αντικείμενα (objects, instances). Σε κάποιες γλώσσες χρησιμοποιούνται και άλλες έννοιες, όπως κανόνες και διαδικασίες. Για κάθε ένα από αυτά τα στοιχεία μοντελοποίησης, το πλαίσιο ορίζει συγκεκριμένα χαρακτηριστικά που μπορούν να προσδιοριστούν και να αξιολογηθούν.

4.1.1 Κλάσεις

Ο όρος της κλάσης (class) ή έννοιας (concept) χρησιμοποιείται με ευρεία σημασία. Οι κλάσεις μπορεί να είναι αφηρημένες ή συμπαγείς, αδιαίρετες ή σύνθετες, πραγματικές ή ιδεατές. Στην ουσία, μια κλάση ή έννοια μπορεί είναι οτιδήποτε στο οποίο υπάρχει η ανάγκη να αναφερθούμε. Για να αξιολογήσει κανείς την εκφραστικότητα μιας γλώσσας όσο αφορά τον ορισμό κλάσεων, θα πρέπει να λάβει υπόψη του τα παρακάτω σημεία:

- Υπάρχει η δυνατότητα ορισμού μετα-κλάσεων (κλάσεων που ορίζουν ένα σύνολο από κλάσεις);
- Παρέχονται μηχανισμοί για να οριστούν πεδία (slots, attributes ή fields); Υπάρχουν διάφορα είδη πεδίων:
 - Πεδία κλάσης: Είναι πεδία που χαρακτηρίζουν την κλάση και έχουν την ίδια τιμή για όλα τα αντικείμενα που ανήκουν στην κλάση.
 - Πεδία αντικειμένου: Είναι πεδία που έχουν διαφορετική τιμή για κάθε αντικείμενο που ανήκει στην κλάση.
 - Τοπικά πεδία: Είναι πεδία που ανήκουν σε μια συγκεκριμένη κλάση και έχουν την ίδια σημασιολογία για όλα τα αντικείμενα που ανήκουν στην κλάση.
 - Πολυμορφικά πεδία: Είναι πεδία των οποίων η σημασιολογία μεταλλάσσεται μεταξύ των υποκλάσεων της κλάσης στην οποία είναι ορισμένα. Για παράδειγμα, το πεδίο *συντάκτης* έχει διαφορετική ερμηνεία στα πλαίσια του όρου *άρθρο* (όπου ο συντάκτης είναι δημοσιογράφος) και στα πλαίσια του όρου *πρακτικό* (όπου ο συντάκτης είναι γραμματέας).
- Τι χαρακτηριστικά (facets) υποστηρίζει η γλώσσα για την περιγραφή των πεδίων; Τέτοια χαρακτηριστικά είναι:

- Αρχική τιμή, που χρησιμοποιείται όταν δεν δίνεται ρητά τιμή στο πεδίο.
- Τύπος: Περιορίζει το είδος της τιμής που μπορεί να πάρει το πεδίο.
- Πολλαπλότητα: ο αριθμός τιμών που μπορεί να πάρει ένα χαρακτηριστικό κατ' ελάχιστο, ή κατά μέγιστο.
- Τεκμηρίωση: Περιγραφή σε φυσική γλώσσα του πεδίου.
- Υποστηρίζεται η δημιουργία νέων χαρακτηριστικών;
- Δυναμικός ορισμός: Η δυνατότητα να οριστεί ένα δυναμικό πεδίο, του οποίου η τιμή λαμβάνεται από μια συγκεκριμένη συνάρτηση ή από την εφαρμογή ενός συγκεκριμένου κανόνα.

4.1.2 Ταξινομίες

Οι ταξινομίες χρησιμοποιούνται ευρέως για την οργάνωση της οντολογικής γνώσης με σχέσεις γενίκευσης – εξειδίκευσης, οι οποίες μπορούν να χρησιμοποιηθούν για την υλοποίηση απλής ή πολλαπλής κληρονομικότητας. Υπάρχουν αρκετές σχέσεις που μπορούν να χρησιμοποιηθούν για την δημιουργία μιας ταξινομίας και είναι ενδιαφέρον να δούμε ποιες από αυτές είναι εκ των προτέρων ορισμένες σε κάθε γλώσσα. Στα πλαίσια της Ontolingua [Farquhar, 1996] ορίζονται οι ακόλουθες τέτοιες σχέσεις:

- Subclass-of: Είναι η απλούστερη σχέση εξειδίκευσης, που συνδέει μια γενική κλάση με μια πιο εξειδικευμένη.
- Partitions: Είναι ο ορισμός ενός συνόλου από αμοιβαία αποκλειόμενες κλάσεις.
- Disjoint decompositions: Είναι ο ορισμός μιας διαίρεσης μιας κλάσης σε κλάσεις και υποκλάσεις, οι οποίες είναι αμοιβαία αποκλειόμενες.

- Exhaustive disjoint decompositions: Είναι ο ορισμός ενός συνόλου αμοιβαία αποκλειόμενων υποκλάσεων, η ένωση των οποίων δίνει την μητρική κλάση.

Κάποιες γλώσσες δεν έχουν αυτές τις σχέσεις ορισμένες εκ των προτέρων, αλλά δίνουν την δυνατότητα να τις ορίσει κανείς χρησιμοποιώντας αξιώματα ή κανόνες.

4.1.3 Σχέσεις

Οι σχέσεις [Gruber, 1993] αντιπροσωπεύουν ένα είδος αλληλεπίδρασης μεταξύ εννοιών της γνωστικής περιοχής. Ορίζονται φορμαλιστικά ως ένα υποσύνολο του καρτεσιανού γινομένου n συνόλων, δηλαδή

$$R: C_1 \times C_2 \times C_3 \times \dots \times C_n$$

Οι διμερείς σχέσεις, που είναι οι συνηθέστερες, λαμβάνονται για $n = 2$.

Σε μια οντολογία, ο ορισμός των σχέσεων και η συσχέτιση τους με άλλα τμήματα της οντολογίας μπορεί να ποικίλλει. Έτσι, σε κάποιες περιπτώσεις οι έννοιες μπορούν να θεωρηθούν μονομερείς σχέσεις, ενώ τα πεδία (attributes) των εννοιών ως διμερείς. Ιδιαίτερη σημασία έχουν οι συναρτήσεις (functions). Μια συνάρτηση ορίζεται ως μια μετάβαση από ένα διάνυσμα σε ένα συγκεκριμένο στοιχείο [Motta, 1999], δηλαδή φορμαλιστικά:

$$F: C_1 \times C_2 \times \dots \times C_{n-1} \rightarrow C_n$$

Το ίδιο αποτέλεσμα μπορεί να επιτευχθεί ορίζοντας την ακόλουθη σχέση:

$$R_F: C_1 \times C_2 \times C_3 \times \dots \times C_{n-1} \times C_n$$

Αυτή η ισοδυναμία υλοποιείται σε κάποιες γλώσσες, αναγνωρίζοντας τις συναρτήσεις ως ένα είδος σχέσεως.

Είναι επίσης σημαντική η υποστήριξη που παρέχει μια γλώσσα ως προς τις παραμέτρους. Συγκεκριμένα, πρέπει να ελεγχθούν τα ακόλουθα σημεία:

- Τι πλήθος παραμέτρων επιτρέπεται να έχει μια σχέση;
- Υπάρχει η δυνατότητα περιορισμού του τύπου των παραμέτρων;
- Υπάρχει η δυνατότητα περιορισμού των τιμών των παραμέτρων; Αυτό συνήθως επιτυγχάνεται μέσω περιορισμών αριότητας (integrity constraints) ή λειτουργικών ορισμών (operational definitions), δηλαδή συναρτήσεων ή κανόνων που μπορούν να χρησιμοποιηθούν για να υπολογιστεί η τιμή μιας παραμέτρου.

4.1.4 Αξιώματα

Τα αξιώματα [Farquhar, 1996] είναι προτάσεις που είναι πάντα αληθείς. Χρησιμοποιούνται κυρίως για να περιορίσουν την πληροφορία που περιέχει μια οντολογία ή να δώσουν την δυνατότητα επαλήθευσης της ορθότητας της. Επίσης, μπορούν να χρησιμοποιηθούν για παραγωγικούς συλλογισμούς.

Η υποστήριξη των γλωσσών κωδικοποίησης οντολογιών ως προς την υποστήριξη αξιωμάτων διαφοροποιείται ως προς την τάξη της λογικής που χρησιμοποιείται (πρώτης ή δεύτερης τάξεως αξιώματα) και τη δυνατότητα ορισμού αξιωμάτων ως ανεξάρτητων συστατικών της οντολογίας (named axioms).

4.1.5 Αντικείμενα

Ο όρος αντικείμενα περιγράφει πραγματικά, υπαρκτά στοιχεία που ενσαρκώνουν μία ή περισσότερες έννοιες. Στη διεθνή βιβλιογραφία οι χρησιμοποιούμενοι όροι είναι facts, individuals, objects, instances ή και claims. Ειδικότερα, ο όρος fact χρησιμοποιείται κυρίως για να δηλώσει μια σχέση που ισχύει ανάμεσα σε αντικείμενα [Motta, 1999]. Ο όρος individual χρησιμοποιείται στην Ontolingua και το OKBC [Farquhar, 1996] για να δηλώσει οτιδήποτε δεν είναι κλάση, δηλαδή όλα τα αντικείμενα, ακόμα και τις σχέσεις. Ο όρος claim (ισχυρισμός) [Luke, 2000] είναι πολύ πιο εξειδικευμένος: αναφέρεται σε μια

πρόταση που υποστηρίζεται ως αληθής από ένα συγκεκριμένο αντικείμενο της γνωστικής περιοχής. Η πρόταση μπορεί να μην ισχύει γενικά στην περιοχή, ή να είναι σε αντίθεση με άλλους ισχυρισμούς που εκφράζουν άλλα αντικείμενα. Η έννοια του ισχυρισμού είναι ιδιαίτερα χρήσιμη σε ανοικτά συστήματα πρακτόρων, όπου είναι αναγκαίος ο διαχωρισμός μεταξύ προτάσεων γενικά παραδεκτών και διασταυρωμένων, και προτάσεων που υποστηρίζονται από ένα ή μία ομάδα πρακτόρων.

4.1.6 Παραγωγικοί Κανόνες

Οι παραγωγικοί κανόνες [MacGregor, 1991] είναι προτάσεις συνεπαγωγής που συνδέουν μια υπόθεση με ένα συμπέρασμα. Τέτοιοι κανόνες υποστηρίζονται ευρέως σε όλες τις γλώσσες, ωστόσο υπάρχει διαφοροποίηση τους στα ακόλουθα σημαντικά στοιχεία:

Υποστήριξη διαζευκτικών και συζευκτικών προτάσεων.

Υποστήριξη μηχανισμού σύνδεσης των κανόνων σε σειρά.

Υποστήριξη εκτέλεσης ενεργειών όταν επαληθεύεται η υπόθεση.

Υποστήριξη για μεταβολή της βάσης γνώσης, με πρόσθεση ή αφαίρεση προτάσεων ή ισχυρισμών.

4.1.7 Μηχανισμοί Εξαγωγής Συμπερασμάτων

Πέρα από τη δυνατότητα στατικής περιγραφής γνώσης, υπάρχει η ανάγκη αξιοποίησης αυτής της γνώσης στα πλαίσια μιας διαδικασίας λογικών συνειρμών για την εξαγωγή συμπερασμάτων [Schreiber, 1999]. Υπάρχει έντονη συσχέτιση ανάμεσα στο σύστημα αποθήκευσης της γνώσης και της λογικής μηχανής, αφού και τα δύο χρησιμοποιούν τις ίδιες δομές δεδομένων. Έτσι, ο κανόνας είναι οι γλώσσες περιγραφής οντολογιών να έχουν τη δική τους μηχανή λογικής. Τα σημεία που θα εξετάσουμε στον τομέα αυτό είναι τα ακόλουθα:

- Υπάρχουν μηχανές λογικής που να υποστηρίζουν τη γλώσσα που εξετάζεται;

- Έχει η μηχανή τη δυνατότητα αυτόματης ταξινόμησης;
- Έχει η μηχανή τη δυνατότητα διαχείρισης πολυμορφισμού ή εξαιρέσεων;
- Υπάρχει υποστήριξη κληρονομικότητας; Υποστηρίζεται απλή ή πολλαπλή κληρονομικότητα;
- Υπάρχει η δυνατότητα εκτέλεσης διαδικασιών;
- Υπάρχει η δυνατότητα διαχείρισης αξιωμάτων;
- Στην εφαρμογή κανόνων, η μηχανή έχει δυνατότητα διασύνδεσης προς τα εμπρός ή προς τα πίσω (forward/backward chaining);

4.1.8 Επιλογή Γλωσσών για Αξιολόγηση

Στα προηγούμενα χρόνια έχει προταθεί ένας σημαντικός αριθμός γλωσσών που μπορούν να υποστηρίξουν την κωδικοποίηση οντολογιών. Η πλέον αποδεκτή από την κοινότητα τεχνητής νοημοσύνης είναι η Ontolingua [Farquhar, 1996], ενώ ανάμεσα στις σημαντικότερες θα πρέπει να περιλάβουμε την LOOM [MacGregor, 1991], την CARIN [Levy, 1998], την OCML [Motta, 1999] και την FLogic [Kifer, 1995]. Η OKBC [Chaudhri, 1997] [Chaudhri, 1998] μπορεί να θεωρηθεί τόσο γλώσσα περιγραφής οντολογιών όσο και προγραμματιστική διαπροσωπεία (API) για την πρόσβαση σε οντολογίες, και θα περιληφθεί επίσης στη μελέτη αυτή.

Επίσης, η σύγχρονη τάση είναι πλέον προς τις γλώσσες που βασίζονται σε XML και RDF [Lassila, 1999], οι οποίες είναι ακόμα υπό εξέλιξη. Τέτοιες γλώσσες είναι η SHOE [Luke, 2000], η XOL [Karp, 1999], η OIL [Horrocks, 00], όπως επίσης και οι RDF Schema [Brickley, 1999] και XML Schema [Thompson, 1999]. Ο ρόλος αυτών των γλωσσών είναι διπλός. Από τη μία, μπορούν να χρησιμοποιηθούν για την κωδικοποίηση της σημασιολογίας ιστοσελίδων ή άλλων στοιχείων πληροφορίας που διατίθενται στο Internet ή σε κάποιο άλλο δίκτυο, ενώ από την άλλη μπορούν να χρησιμοποιηθούν για την ανταλλαγή

οντολογιών διαμέσου του Internet. Στο [Harmelen, 1999] δίνεται μια εμπειριστατωμένη ανάλυση του ρόλου τέτοιων γλωσσών στη σημασιολογία του WWW και της αλληλεπίδρασης τους με τις HTML, XML και RDF. Αυτές οι γλώσσες αποτελούν την αιχμή του δόρατος της έρευνας αυτή τη στιγμή, και θα περιληφθούν επίσης στην παρούσα μελέτη.

4.2 ΚΛΑΣΣΙΚΕΣ ΓΛΩΣΣΕΣ ΠΕΡΙΓΡΑΦΗΣ

ΟΝΤΟΛΟΓΙΩΝ

Στη συνέχεια εξετάζουμε κάθε μια γλώσσα ξεχωριστά, σε σχέση με το πλαίσιο αξιολόγησης που παρουσιάστηκε παραπάνω.

4.2.1 *Ontolingua*

Η *Ontolingua* [Farquhar, 1996] είναι μια γλώσσα βασισμένη στην KIF [Genesereth, 1992] και την οντολογία Frame Ontology [Gruber, 1993]. Η γλώσσα έχει αναπτυχθεί για χρήση με τον Ontolingua Server, που συνδυάζει μια μηχανή λογικής με ένα σύστημα διαχείρισης βάσης γνώσεως. Αναπτύχθηκε στο Knowledge Systems Laboratory του Stanford University.

Η KIF (Knowledge Interchange Format), στην οποία βασίζεται, είναι λιγότερο μια γλώσσα και περισσότερο ένα πρωτόκολλο που αναπτύχθηκε με στόχο να αντιμετωπίσει την ετερογένεια των γλωσσών αναπαράστασης γνώσης, επιτρέποντας την ανταλλαγή γνώσης μεταξύ ανομοιογενών πληροφοριακών συστημάτων. Η KIF υποστηρίζει την περιγραφή αντικειμένων, σχέσεων και συναρτήσεων. Έχει δηλωτική σημασιολογία (declarative semantics), δηλαδή είναι δυνατό να κατανοήσει κανείς εύκολα το νόημα μιας πρότασης χωρίς τη χρήση μεταγλωττιστή, και βασίζεται σε κατηγορηματικό λογισμό πρώτης τάξεως. Παρέχει τη δυνατότητα περιγραφής μετα-γνώσης, δηλαδή γνώσης που αφορά τη δομή και σημασιολογία της γνώσης περιοχής, και επιτρέπει την αναπαράσταση μη μονοτονικών παραγωγικών κανόνων. Οι προτάσεις της KIF

αποτελούνται από λίστες, κάθε μια από τις οποίες ξεκινά με ένα σταθερό όρο που αναπαριστά μια σχέση και συνεχίζεται με έναν αριθμό όρων, σταθερών ή μη, που συμμετέχουν στη σχέση. Προτάσεις μπορούν να συνδυαστούν μεταξύ τους με τη χρήση λογικών τελεστών.

Επειδή η KIF έχει κατασκευαστεί με τη χρήση της ανταλλαγής γνώσης κι όχι της απευθείας κωδικοποίησης κατά νου, είναι πολύ κουραστική η χρήση της ως γλώσσας περιγραφής οντολογιών. Πρέπει όμως να διασαφηνιστεί ότι η γλώσσα έχει την ικανότητα να χρησιμοποιηθεί για το σκοπό αυτό. Με σκοπό να διευκολυνθεί η χρήση της KIF για την περιγραφή οντολογιών, αναπτύχθηκε σε KIF η οντολογία Frame Ontology [Gruber, 1993]. Η οντολογία αυτή παίζει ρόλο αντίστοιχο με αυτό των βιβλιοθηκών στις συμβατικές γλώσσες προγραμματισμού. Έτσι, παρέχει δομές υψηλού επιπέδου και μεθόδους διαχείρισης τους, απαλλάσσοντας τον προγραμματιστή από την ανάγκη να δουλέψει σε χαμηλό επίπεδο και μειώνοντας τον όγκο εργασίας του, αυξάνοντας παράλληλα την αναγνωσιμότητα του παραγόμενου κώδικα. Η Frame Ontology δίνει τη δυνατότητα περιγραφής οντολογιών βασισμένων στο μοντέλο των πλαισίων (frames). Το πετυχαίνει αυτό περιγράφοντας τις δομικές μονάδες γνώσης που περιλαμβάνονται στο μοντέλο των πλαισίων, καθώς και τις μεταξύ τους σχέσεις. Έτσι ορίζει έννοιες όπως class, instance, subclass-of, instance-of και άλλες.

Εξαιτίας του γεγονότος ότι η Frame Ontology ακολουθεί ένα πολύ συγκεκριμένο μοντέλο γνώσης, αυτό των πλαισίων, είναι λιγότερο εκφραστική από την KIF. Έτσι, δεν είναι δυνατό να εκφράσει κανείς σε όρους Frame Ontology όλα τα στοιχεία γνώσης που μπορούν να εκφραστούν σε KIF. Για το λόγο αυτό, και προκειμένου να μην υποστεί την αντίστοιχη μείωση εκφραστικότητας, η Ontolingua επιτρέπει την εισαγωγή εκφράσεων σε KIF μέσα σε ορισμούς βασισμένους στη Frame Ontology. Επομένως, η Ontolingua δίνει τη δυνατότητα περιγραφής οντολογιών με τρεις διαφορετικούς τρόπους:

1. Χρησιμοποιώντας αποκλειστικά το λεξιλόγιο που παρέχει η Frame Ontology, που περιορίζεται στο μοντέλο των πλαισίων και δεν μπορεί να αναπαραστήσει αξιώματα.
2. Χρησιμοποιώντας αποκλειστικά την KIF.
3. Χρησιμοποιώντας τόσο την KIF όσο και την Frame Ontology, ανάλογα με τις ανάγκες έκφρασης της συγκεκριμένης οντολογίας. Σε αυτό το μοντέλο η περιγραφή της οντολογίας γίνεται κατά βάση σε Frame Ontology, καταφεύγοντας σε KIF όποτε η εκφραστικότητα της Frame Ontology δεν είναι αρκετή.

Οι ορισμοί της Ontolingua αποτελούνται από μια επικεφαλίδα (heading), μια άτυπη περιγραφή σε φυσική γλώσσα και τον ορισμό σε KIF ή σε Frame Ontology. Οι οντολογίες εισάγονται στον Ontolingua Server με τη χρήση του εργαλείου Ontology Editor, μιας εφαρμογής σύνταξης οντολογιών. Επίσης υπάρχει η δυνατότητα εισαγωγής οντολογιών που έχουν αναπτυχθεί εξωτερικά και είναι αποθηκευμένες σε αρχεία απλού κειμένου.

4.2.1.1 Αναπαράσταση γνώσης

Η περιγραφή της γνώσης σε μια οντολογία βασίζεται στο μοντέλο πλαισίων που παρέχει η Frame Ontology. Τα δομικά στοιχεία που μπορούν να χρησιμοποιηθούν είναι τα παρακάτω:

- **Κλάσεις.** Οι κλάσεις είναι ένα σύνολο από αντικείμενα. Φορμαλιστικά, μια κλάση ορίζεται ως μια μονομερής σχέση και επομένως μπορεί να χρησιμοποιηθεί σαν κατηγορηματικό λογισμό πρώτης τάξεως. Ο ορισμός μιας κλάσης μπορεί να συμπληρωθεί με τα πεδία που αυτή περιέχει καθώς και χαρακτηριστικά τους όπως τιμές, πολλαπλότητα τιμών και άλλα. Τα πεδία αυτά πρέπει να οριστούν ξεχωριστά ως διμερείς σχέσεις. Επίσης, αναγκαίες και ικανές συνθήκες μπορούν να οριστούν, τις οποίες πρέπει να πληρούν τα αντικείμενα που

ανήκουν σε μία κλάση. Αυτές οι συνθήκες δεν χρησιμοποιούνται για αυτόματη ταξινόμηση, αλλά μόνο για έλεγχο περιορισμών. Επιπλέον, μπορούν να οριστούν άλλες προτάσεις δευτέρας τάξεως (προτάσεις που αναφέρονται στην ίδια την κλάση).

- **Ταξινομίες.** Η Ontolingua υποστηρίζει διάφορα δομικά στοιχεία για την κατασκευή ταξινομιών, όπως subclass-of, class-partition, subclass-partition και exhaustive-subclass-partition. Έτσι επιτυγχάνεται μεγάλη ευελιξία και εκφραστικότητα στον ορισμό ταξινομιών. Πρέπει πάντως να σημειωθεί ότι δεν υπάρχει δομικό στοιχείο που να εκφράζει την έννοια not-subclass-of, ωστόσο αυτή η σχέση μπορεί να οριστεί ως η άρνηση της σχέσης subclass-of.
- **Σχέσεις.** Οι σχέσεις στην Ontolingua είναι πολυμερείς και ορίζονται ως ένα σύνολο n στοιχείων που ικανοποιούν τη σχέση. Η Ontolingua δίνει τη δυνατότητα ορισμού των περιορισμών στους οποίους πρέπει να υπακούουν τα στοιχεία αυτού του συνόλου, προκειμένου να συμμετέχουν στη σχέση. Επίσης, τα πεδία ορίζονται ως διμερείς σχέσεις μεταξύ της κλάσης και του τύπου του πεδίου. Τέλος, στη Frame Ontology ορίζονται κάποιες κατηγορίες σχέσεων όπως αυτοπαθείς, συμμετρικές και μεταβατικές, έτσι ώστε οι σχέσεις που ορίζει κανείς να μπορούν να κληρονομήσουν τα χαρακτηριστικά της κατηγορίας στην οποία ανήκουν.
- **Συναρτήσεις.** Οι συναρτήσεις αντιμετωπίζονται στην Ontolingua ως ειδικές περιπτώσεις σχέσεων, όπου η τιμή του n -οστού στοιχείου είναι μοναδική, για δεδομένες τιμές των πρώτων $n-1$ στοιχείων. Περιορισμοί μπορούν να οριστούν, όπως επίσης αναγκαίες και ικανές συνθήκες που πρέπει να πληρούν οι τιμές προκειμένου να έχει νόημα η συνάρτηση.
- **Αντικείμενα.** Τα αντικείμενα (objects, instances) ή άτομα (individuals) είναι μέλη μιας κλάσης. Ορίζονται δηλωτικά, δηλαδή πρέπει να εκφραστεί ευθέως η σχέση μεταξύ ενός αντικειμένου και της κλάσης στην οποία

ανήκει. Έτσι, ο ορισμός ενός αντικειμένου περιλαμβάνει το όνομα του, το όνομα της κλάσης στην οποία ανήκει και πιθανώς ένα σύνολο τιμών για τα πεδία που έχουν οριστεί στην κλάση.

- **Αξιώματα.** Ο ορισμός αξιωμάτων στην Ontolingua είναι δυνατός μόνο με την απευθείας χρήση προτάσεων σε KIF.
- **Κανόνες.** Ο ορισμός κανόνων είναι δυνατός, αν και αυτοί δεν θεωρούνται βασικό συστατικό δόμησης οντολογιών. Στην ουσία η έννοια του κανόνα δεν υπάρχει στην Ontolingua, αλλά ο ορισμός τους είναι δυνατός με τη χρήση του τελεστή *:sufficient* κατά τον ορισμό μιας κλάσης ή σχέσης. Πρέπει όμως να σημειωθεί ότι ο Ontolingua Server δεν εκτελεί αυτόματα τέτοιους κανόνες.

4.2.1.2 Μηχανή λογικής

Η υποστήριξη λογικών συνειρμών είναι το βασικό μειονέκτημα της Ontolingua. Εξαιτίας της υψηλής εκφραστικότητας της, η ανάπτυξη μιας μηχανής λογικής που να επιτρέπει την εκτέλεση λογικών συνειρμών βάσει της γνώσης που κωδικοποιείται σε Ontolingua έχει αποδειχθεί εξαιρετικά δύσκολη. Είναι σαφής η ύπαρξη μιας ισορροπίας ανάμεσα στην εκφραστικότητα μιας οντολογίας και την ευκολία χρήσης της για συλλογισμούς. Η Ontolingua βρίσκεται στην πλευρά της εκφραστικότητας, επιλέγοντας έτσι να θυσιάσει την ευκολία πραγματοποίησης συλλογισμών. Έτσι, τα μόνα χαρακτηριστικά που υποστηρίζονται από τη μηχανή λογικής της είναι ο έλεγχος περιορισμών (constraint checking) και η κληρονομικότητα. Μια λύση για την παράκαμψη αυτού του προβλήματος είναι η μετάφραση της οντολογικής γνώσης από Ontolingua σε LOOM, η οποία έχει μια πλήρη μηχανή λογικής. Ωστόσο σε αυτή την περίπτωση υπάρχει το πρόβλημα της ανακριβούς μετάφρασης, αφού αρκετά στοιχεία της γνώσης που κωδικοποιείται σε Ontolingua δεν είναι μεταφράσιμα σε LOOM.

4.2.2 OKBC

Το ακρωνύμιο OKBC [Chaudhri, 1998] προέρχεται από την ονομασία Open Knowledge Base Connectivity (Ανοικτή Διασύνδεση Βάσεων Γνώσης) και αποτελεί εξέλιξη του Generic Frame Protocol. Αποτελεί αποτέλεσμα συνεργασίας μεταξύ του Κέντρου Τεχνητής Νοημοσύνης του SRI και του Εργαστηρίου Συστημάτων Γνώσης του Πανεπιστημίου Stanford. Το OKBC είναι ένα πρωτόκολλο για την πρόσβαση σε γνώση που είναι αποθηκευμένη σε βάσεις γνώσης που χρησιμοποιούν το μοντέλο των πλαισίων για την κωδικοποίηση τους. Η πρόσβαση σε βάσεις γνώσης που δεν χρησιμοποιούν αυτό το μοντέλο είναι δυνατή μόνο με την προσαρμογή της γνώσης στο μοντέλο των πλαισίων. Το OKBC αποτελεί το απαραίτητο συμπλήρωμα του ορισμού γλωσσών για την ανταλλαγή γνώσης.

Το OKBC βασίζεται στο μοντέλο αναπαράστασης γνώσης του Generic Frame Protocol (GFP) [Karp, 1995] [Chaudhri, 1997]. Πρόκειται για ένα μοντέλο βασισμένο σε πλαίσια και, ως τέτοιο, χρησιμοποιεί τα τυπικά δομικά στοιχεία που συναντώνται σε συστήματα αναπαράστασης γνώσης με πρότυπα, όπως σταθερές (constants), πλαίσια (frames), πεδία (slots), όψεις (facets), κλάσεις (classes) και αντικείμενα (instances).

Αξίζει να αναφερθεί η κατασκευή της OKBC Ontology for Ontolingua, η οποία περιλαμβάνεται στη Frame Ontology και είναι επομένως διαθέσιμη σε όλες τις οντολογίες που αναπτύσσονται σε Ontolingua. Η περίληψη της OKBC Ontology σαν υποσύνολο της Frame Ontology επιβεβαιώνει την μεγαλύτερη εκφραστικότητα της Ontolingua.

Τελικά το OKBC συνδυάζει μια προγραμματιστική διαπροσωπεία με ένα μοντέλο αναπαράστασης γνώσης και ένα πρωτόκολλο πρόσβασης σε γνώση. Έτσι, μπορεί κανείς να ερμηνεύσει το OKBC με τρεις διαφορετικούς τρόπους:

- Ως ένα πρωτόκολλο για την πρόσβαση σε συστήματα γνώσης βασισμένα σε πλαίσια, το οποίο εμπεριέχει τις ελάχιστες δυνατές σημασιολογικές συμβάσεις.
- Ως ένα πρωτόκολλο για την πρόσβαση σε συστήματα γνώσης βασισμένα σε πλαίσια, το οποίο επιτρέπει τη μέγιστη δυνατή εκφραστικότητα.
- Ως μια οντολογία, την OKBC Ontology, που ορίζει το μοντέλο γνώσης που χρησιμοποιείται από το πρωτόκολλο.

4.2.2.1 Αναπαράσταση γνώσης

Το OKBC βασίζεται σε μοντέλο πλαισίων, επομένως το πιο σημαντικό δομικό στοιχείο είναι προφανώς το πλαίσιο. Τα πλαίσια μπορούν να αναπαραστήσουν τόσο κλάσεις όσο και αντικείμενα. Σε σχέση με τα δομικά στοιχεία γνώσης που εξετάστηκαν στο πλαίσιο αξιολόγησης παραπάνω, μπορούμε να πούμε τα εξής:

- **Κλάσεις.** Οι κλάσεις αντιμετωπίζονται ως σύνολα οντοτήτων. Είναι δυνατός ο ορισμός πεδίων τοπικά, δηλαδή για μία συγκεκριμένη κλάση. Επίσης, υπάρχει η δυνατότητα ορισμού πεδίων της κλάσης (own slots), τα οποία έχουν την ίδια τιμή για όλα τα αντικείμενα της κλάσης, και πεδίων αντικειμένου (template slots), τα οποία παίρνουν ξεχωριστή τιμή για κάθε instance της κλάσης. Περιορισμοί στον τύπο και την πολλαπλότητα τιμών των πεδίων μπορούν επίσης να οριστούν.

Πρέπει να σημειωθεί εδώ η διαφοροποίηση μεταξύ στοιχειωδών (primitive) και μη στοιχειωδών (non-primitive) κλάσεων. Αυτή η διαφοροποίηση έχει υιοθετηθεί από γλώσσες βασισμένες σε description logic, όπως η LOOM, οι οποίες μπορούν να πραγματοποιήσουν αυτόματη ταξινόμηση. Έτσι, στοιχειώδεις κλάσεις είναι αυτές για τις οποίες δεν μπορεί να πραγματοποιηθεί αυτόματη ταξινόμηση, και πρέπει κάθε αντικείμενο να συσχετισθεί με αυτές δηλωτικά. Οι μη στοιχειώδεις κλάσεις είναι αυτές για

τις οποίες είναι δυνατή η αυτόματη ταξινόμηση, δηλαδή με την εισαγωγή ενός αντικειμένου στη βάση γνώσης είναι δυνατό αυτό να συσχετισθεί αυτόματα με την μη στοιχειώδη κλάση στην οποία ανήκει και να ελεγχθεί κατά πόσο ικανοποιεί τους περιορισμούς των πεδίων της κλάσης αυτής.

- **Ταξινομίες.** Στο OKBC μπορούν να οριστούν απλές ταξινομίες, με τη χρήση του τελεστή `direct-superclasses`, που είναι σημασιολογικά ισοδύναμο του συνηθέστερου `subclass-of`. Πρέπει να σημειωθεί ότι ο ορισμός κατατμήσεων (`partitions`) δεν είναι δυνατός, ούτε άμεσα (δεν υπάρχει σχετικό όρισμα) ούτε έμμεσα (δεν υπάρχει η δυνατότητα ορισμού αξιωμάτων).
- **Σχέσεις.** Οι σχέσεις δεν υφίστανται ως ανεξάρτητα δομικά στοιχεία οντολογιών στο OKBC. Ωστόσο, πολυμερείς σχέσεις μπορούν να οριστούν σαν κλάσεις, που συνήθως ονομάζονται `associative classes`, χρησιμοποιώντας πεδία που δέχονται ως τιμή τα αντικείμενα που μετέχουν στη σχέση.
- **Συναρτήσεις.** Οι συναρτήσεις επίσης δεν υφίστανται ως ανεξάρτητα δομικά στοιχεία οντολογιών. Όπως και με τις σχέσεις, είναι δυνατό να ορίσει κανείς κλάσεις που να πληρούν τη σημασιολογία των συναρτήσεων. Ωστόσο, δεν είναι δυνατό να οριστούν εκτελέσιμες διαδικασίες για τον υπολογισμό της τιμής της συνάρτησης. Επίσης, δεν είναι δυνατό να εκφραστεί κάποιος περιορισμός που να εξασφαλίζει τη μοναδικότητα του αποτελέσματος της συνάρτησης για ένα δεδομένο σύνολο παραμέτρων.
- **Διαδικασίες.** Το OKBC επιτρέπει τον ορισμό εκτελέσιμων διαδικασιών αντιστοιχών με αυτών της Lisp. Αυτές μπορούν να αξιοποιηθούν για την εκτέλεση εργασιών στη βάση γνώσης.

- **Αντικείμενα.** Τα αντικείμενα ορίζονται ως πλαίσια, ακριβώς όπως και οι κλάσεις, όπου το όρισμα `direct-types` χρησιμοποιείται για να δηλώσει την ή τις κλάσεις στις οποίες ανήκει το αντικείμενο. Η ίδια ακριβώς σημασιολογία χρησιμοποιείται για να ορίσει κανείς μια υποκλάση. Έτσι, το OKBC ταυτίζει ένα αντικείμενο με την υποκλάση που περιέχει μόνο το συγκεκριμένο στοιχείο. Αφού οι κλάσεις ορίζονται ως σύνολα αντικειμένων, σε όρους θεωρίας συνόλων αυτό σημαίνει ότι ένα αντικείμενο αναπαρίσταται με το σύνολο που περιέχει μόνο αυτό το αντικείμενο.
- **Αξιώματα.** Δεν υπάρχει εγγενής υποστήριξη για τα αξιώματα στο OKBC. Ένα σύστημα που υλοποιεί το OKBC μπορεί να υποστηρίξει αξιώματα μέσω του τελεστή `tell`, το οποίο επιτρέπει την εκτέλεση διαδικασιών στη βάση γνώσης. Βέβαια αυτή η λύση δε μπορεί να γενικευθεί και εξαρτάται από την συγκεκριμένη βάση γνώσης.
- **Κανόνες.** Οι παραγωγικοί κανόνες δεν υποστηρίζονται εγγενώς από το OKBC. Δεν υπάρχει απλή μέθοδος για την υλοποίησή τους.

4.2.2.2 Μηχανή λογικής

Το OKBC είναι κατασκευασμένο με σκοπό την ανταλλαγή και όχι την επεξεργασία γνώσης, έτσι υποστηρίζει μόνο πολύ βασικές λογικές διεργασίες. Αυτές είναι:

- Έλεγχος περιορισμών
- Αρχικοποίηση παραμέτρων
- Κληρονομικότητα

Για τις διεργασίες αυτές προσφέρεται η δυνατότητα παραμετροποίησης της συμπεριφοράς τους, για παράδειγμα μπορεί να καθοριστεί το πότε

πραγματοποιείται ο έλεγχος περιορισμών, ενώ είναι δυνατή και η απενεργοποίηση τους.

4.2.3 OCML

Η Operational Conceptual Modeling Language (Γλώσσα Λειτουργικής Εννοιολογικής Μοντελοποίησης) [Motta, 1999] αναπτύχθηκε στο Knowledge Media Institute της Αγγλίας. Ο αρχικός στόχος της ανάπτυξης της ήταν η περιγραφή λειτουργικών μοντέλων για το έργο VITAL. Η OCML είναι έτσι μια γλώσσα που αναπτύχθηκε για πρακτικούς σκοπούς. Στην ανάπτυξη της έχουν ληφθεί υπόψη αρκετοί πρακτικοί παράγοντες. Ένας από αυτούς είναι η συμβατότητα με αποδεκτά πρότυπα. Έτσι η OCML έχει βασιστεί στην Ontolingua και μπορεί κανείς να την περιγράψει ως μια λειτουργική παραλλαγή της, που υποστηρίζει την απόδειξη θεωρημάτων και τον υπολογισμό συναρτήσεων.

Όντας βασισμένη στην Ontolingua, η OCML παρέχει υποστήριξη για σχέσεις, συναρτήσεις, κλάσεις, αντικείμενα και αξιώματα. Επιπλέον, προσθέτει ορίσματα για την υποστήριξη κανόνων με διασύνδεση προς τα εμπρός και προς τα πίσω (forward/backward chaining). Για να βελτιώσει την επίδοση της γλώσσας στην εκτέλεση, προσθέτει ακόμα κάποιους μηχανισμούς βελτιστοποίησης, όπως για παράδειγμα procedural attachments. Τέλος υποστηρίζει ένα γενικό μοντέλο ερωταπαντήσεων (tell & ask interface) που μπορεί να χρησιμοποιηθεί για να προσθέσει κανείς προτάσεις στην βάση γνώσης, ή να εξετάσει τα περιεχόμενα της.

Για να υποβοηθήσει την ανάπτυξη οντολογιών, η OCML παρέχει μια βιβλιοθήκη από δώδεκα βασικές οντολογίες που διευκολύνουν τη διαχείριση βασικών τύπων δεδομένων όπως λίστες, αριθμοί, σύνολα και αλφαριθμητικά (strings), και ορίζουν τα βασικά στοιχεία μιας οντολογίας σε αντιστοιχία με τη Frame Ontology, όπως κλάσεις, αντικείμενα, συναρτήσεις και λοιπά.

4.2.3.1 Αναπαράσταση γνώσης

Η OCML μπορεί να περιγράψει γνώση με χρήση των ακόλουθων δομικών συστατικών.

- **Κλάσεις.** Ο ορισμός των κλάσεων είναι σχεδόν πανομοιότυπος με αυτόν που χρησιμοποιεί η Ontolingua. Τα πεδία, που ονομάζονται υποδοχές (slots), ορίζονται σαν μέρος της δήλωσης της κλάσης μαζί με τις βασικότερες ιδιότητες τους (επιτρεπτές τιμές, αρχικές τιμές, περιορισμοί, τύπος, πολλαπλότητα). Είναι δυνατός και ο καθορισμός της συμπεριφοράς των αρχικών τιμών κατά την κληρονομικότητα. Αναγκαίες ή ικανές συνθήκες για την υπαγωγή σε μια κλάση μπορούν να δοθούν μέσα στον ορισμό της κλάσης, καθώς επίσης και περιορισμοί ή άλλες προτάσεις που σχετίζονται με τη λειτουργική συμπεριφορά της κλάσης, δηλαδή στο πώς μπορεί να αποδειχθεί κατά την εκτέλεση ότι ένα αντικείμενο ανήκει στην κλάση, ή στο πώς μπορεί να εξαχθεί μια λίστα με όλα τα αντικείμενα που ανήκουν στην κλάση.
- **Ταξινομίες.** Οι ταξινομίες ορίζονται ως ιεραρχίες κλάσεων. Στον ορισμό κάθε κλάσης περιλαμβάνονται οι υπερκλάσεις και υποκλάσεις της. Κατατμήσεις δεν υποστηρίζονται άμεσα, αλλά μπορούν να οριστούν έμμεσα με τη χρήση αξιωμάτων.
- **Σχέσεις.** Η OCML υποστηρίζει πολυμερείς σχέσεις.
- **Συναρτήσεις.** Οι συναρτήσεις δέχονται μια σειρά από παραμέτρους, στις οποίες αντιστοιχίζουν μια συγκεκριμένη τιμή. Αν και ορίζονται μέσω συναρτήσεων της Ontolingua, υπάρχει μεγάλη διαφορά μεταξύ των συναρτήσεων της Ontolingua και αυτών της OCML. Η διαφορά επικεντρώνεται σε θέματα εκτέλεσης, καθώς η OCML επιτρέπει τον ορισμό εκφράσεων για τον έλεγχο της ορθότητας των παραμέτρων και της επιστρεφόμενης τιμής. Σημαντικότερα ακόμα, δίνει τη δυνατότητα

ορισμού εκφράσεων που μπορούν να εκτελεστούν για τον υπολογισμό τις επιστρεφόμενης τιμής για δεδομένες τιμές των παραμέτρων.

- **Διαδικασίες.** Στα πλαίσια του λειτουργικού χαρακτήρα της, η OCML υποστηρίζει διαδικασίες οι οποίες δεν μπορούν να οριστούν ως συναρτήσεις, αφού δεν μεταβάλλουν μόνο μια επιστρεφόμενη τιμή αλλά μπορούν να προκαλέσουν γενικότερες μεταβολές στη βάση γνώσης. Για τον ορισμό διαδικασιών χρησιμοποιείται η γλώσσα Lisp.
- **Αντικείμενα.** Ανήκουν σε μία και μόνο κλάση και ορίζονται ως ένα σύνολο πεδίων και τιμών. Τα πεδία πρέπει να είναι αυτά που έχουν οριστεί για την κλάση στην οποία ανήκει το αντικείμενο, ενώ οι τιμές πρέπει να ικανοποιούν τους τυχόν περιορισμούς που έχουν δηλωθεί για αυτές στον ορισμό της κλάσης.
- **Αξιώματα.** Στην OCML υπάρχει δυνατότητα ορισμού αξιωμάτων. Τα αξιώματα ορίζονται ως λογικές προτάσεις.
- **Κανόνες.** Στην OCML υπάρχουν δύο είδη κανόνων, οι ορθοί και οι ανάστροφοι κανόνες (forward/backward rules), που διαχωρίζονται σε σχέση με τις απαιτήσεις αναπαράστασης γνώσης κάθε οντολογίας. Οι ανάστροφοι κανόνες μπορούν να θεωρηθούν συμπερασματικοί κανόνες. Λειτουργούν με τη λογική της Prolog, ορίζοντας κάποιες συνθήκες και το συμπέρασμα που ισχύει όταν αυτές οι συνθήκες επαληθεύονται. Στον ορισμό κλάσεων ή σχέσεων, μπορούν να συμπεριληφθούν με τη χρήση του τελεστή `prove-by`, που δείχνει ακριβώς ότι η προσδοκώμενη χρήση τους είναι για αιτιολόγηση. Επομένως χρησιμοποιούνται με διασύνδεση προς τα πίσω, γεγονός στο οποίο οφείλουν την ονομασία τους.

Οι ορθοί κανόνες, αντίθετα, είναι παραγωγικοί κανόνες που εκτελούνται όταν οι συνθήκες τους ικανοποιούνται. Οι ορθοί κανόνες δεν ορίζουν μόνο ένα λογικό συμπέρασμα που ισχύει όταν ισχύουν οι συνθήκες, αλλά

ακόμα και ενέργειες που θα εκτελεστούν στη βάση γνώσης μόλις αυτό συμβεί, όπως περίληψη ή εξαίρεση τμημάτων της, αλλαγή τιμών κλπ. Οι ορθοί κανόνες της OCML διαφέρουν σημαντικά από τους συνήθεις παραγωγικούς κανόνες στο γεγονός ότι όποιες ενέργειες καθορίζονται στο δεξιό μέρος τους εκτελούνται σειριακά, μέχρι μία τους να αποτύχει. Τότε η εκτέλεση ενεργειών σταματά.

4.2.3.2 Μηχανή λογικής

Η μηχανή λογικής της OCML αποτελείται από δύο διακριτά τμήματα. Το πρώτο είναι ένα υποσύστημα που αναλαμβάνει την επαλήθευση των περιορισμών και συνθηκών αρτιότητας που ορίζονται στην οντολογία. Το δεύτερο υποσύστημα εκτελεί τις ενέργειες που ορίζονται σε διάφορα σημεία της οντολογίας για παράδειγμα στους κανόνες, όποτε αυτό χρειάζεται.

Ένα σημαντικό χαρακτηριστικό της OCML είναι η δυνατότητα καθορισμού του μοντέλου διασύνδεσης κανόνων (chaining mode) που θα χρησιμοποιηθεί στην οντολογία. Αυτό δίνει τη δυνατότητα στον προγραμματιστή να ελέγξει άμεσα τους λογικούς μηχανισμούς της γλώσσας και να επιδιώξει μια λειτουργικά αποδοτική λύση. Η μηχανή λογικής είναι μονοτονική και πραγματοποιεί μη μονοτονική αιτιολόγηση μόνο όταν διαχειρίζεται τις αρχικές τιμές πεδίων. Τέλος, υποστηρίζει πολλαπλή κληρονομικότητα στις κλάσεις της OCML.

4.2.4 FLogic

Η FLogic [Kifer, 1995], της οποίας το όνομα προκύπτει ως ακρωνύμιο του Frame Logic, είναι μια γλώσσα που συνδυάζει χαρακτηριστικά γλωσσών βασισμένων σε πλαίσια, με κατηγορηματικό λογισμό πρώτης τάξεως (first-order predicate calculus). Είναι μια σαφής γλώσσα, που αντιμετωπίζει με περιγραφικό και ευανάγνωστο τρόπο τις περισσότερες δομικές έννοιες του αντικειμενοστραφούς και του βασισμένου σε πλαίσια μοντέλου. Υποστηρίζει κληρονομικότητα, πολυμορφισμό, σύνθετα αντικείμενα, απομόνωση

(encapsulation) και άλλα χαρακτηριστικά. Χρησιμοποιεί αντικειμενοστραφή ορολογία (objects, attributes κλπ) αντί της ορολογίας πλαισίων (frames, slots κλπ), ωστόσο στην ανάλυση της θα χρησιμοποιήσουμε την κοινή ορολογία που δίνεται στο πλαίσιο αξιολόγησης.

Από μια άποψη, η FLogic έχει με το αντικειμενοστραφές μοντέλο την ίδια σχέση που ο κατηγορηματικός λογισμός έχει με το σχεσιακό προγραμματισμό. Έτσι η FLogic ακολουθεί τη θεωρητική σημασιολογία του μοντέλου και περιέχει μια πλήρη και ορθή θεωρία αιτιολόγησης. Οι θεμελιώδεις έννοιες του αντικειμενοστραφούς μοντέλου αντικατοπτρίζονται άμεσα στην FLogic, ενώ οι δευτερεύουσες έννοιες μπορούν να αναπαρασταθούν αρκετά εύκολα.

Η FLogic δεν αναπτύχθηκε ως γλώσσα περιγραφής οντολογιών, αλλά ως μια αντικειμενοστραφής προσέγγιση στον κατηγορηματικό λογισμό πρώτης τάξεως. Η προσδοκώμενη χρήση ήταν σε συμπερασματικές (deductive) και αντικειμενοστραφείς βάσεις δεδομένων. Ωστόσο, αργότερα προσαρμόστηκε για χρήση στην περιγραφή οντολογιών, στην οποία αξιοποιήθηκε και εκτιμήθηκε σε μεγάλο βαθμό. Έχει χρησιμοποιηθεί σε μεγάλα έργα οντολογίας, όπως το Ontobroker [Decker, 1999] και το (KA)² [Benjamins, 1998], ενώ παράλληλα έχει αποτελέσει τη βάση ανάπτυξης ενός συστήματος για την παρουσίαση στον ιστό, γνώσης που περιέχεται σε οντολογίες [Erdmann, 1999].

4.2.4.1.1 Αναπαράσταση γνώσης

Η FLogic παρέχει τα ακόλουθα συστατικά στοιχεία περιγραφής γνώσης:

- **Κλάσεις.** Η FLogic επιτρέπει τον ορισμό κλάσεων στο πρότυπο των πλαισίων. Τα πεδία πρέπει να συμπεριληφθούν εξολοκλήρου στον ορισμό της κλάσης. Για κάθε πεδίο πρέπει να οριστούν περιορισμοί στον τύπο και τη μέγιστη πολλαπλότητα του. Αυτό περιορίζει την πολυπλοκότητα στους περιορισμούς πολλαπλότητας που μπορεί να αξιοποιήσει κανείς. Επίσης, δεν υπάρχει υποστήριξη για την τεκμηρίωση των πεδίων. Έτσι,

από τη μία πλευρά ο ορισμός των πεδίων μέσα στις κλάσεις και η έλλειψη πολύπλοκων χαρακτηριστικών στον ορισμό τους, κάνει την προσέγγιση της FLogic στον ορισμό κλάσεων συμπαγή και ευανάγνωστη.

- **Ταξινομίες.** Υπάρχει μόνο ένα προκαθορισμένο όρισμα για τον ορισμό ταξινομιών στην FLogic, το σύνηθες subclass-of. Στην Flogic αναπαρίσταται με το σύμβολο ::, για παράδειγμα η έκφραση A::B υποδηλώνει ότι η έννοια (κλάση) A είναι μια εξειδίκευση (υποκλάση) της έννοιας (κλάσης) B. Πολλαπλή κληρονομικότητα υποστηρίζεται στον ορισμό ταξινομιών, ωστόσο δεν υπάρχει άμεση υποστήριξη για τον ορισμό κατατμήσεων (partitions). Είναι δυνατό να οριστούν κατατμήσεις έμμεσα, χρησιμοποιώντας αξιώματα για να καθορίσει κανείς την ασυμβατότητα μεταξύ υποκλάσεων.
- **Σχέσεις.** Καθώς η FLogic βασίζεται κατά το ήμισυ σε λογική πλαισίων, οι σχέσεις δεν υποστηρίζονται ως δομικό συστατικό οντολογιών. Για την αναπαράσταση τους μπορεί να ακολουθηθεί η ίδια διαδικασία που προτάθηκε και στο OKBC. Αυτή συνίσταται σε χρήση κλάσεων (πλαισίων) που αντιπροσωπεύουν τη σχέση και συνδέουν άλλες κλάσεις μεταξύ τους. Με τον τρόπο αυτό, είναι δυνατή η αναπαράσταση πολυμερών σχέσεων. Θα πρέπει εδώ να επισημανθεί η έντονη διαφορά μεταξύ γλωσσών όπως η Ontolingua ή η OCML, που υπάρχουν πολλές εναλλακτικές επιλογές για τον καθορισμό χαρακτηριστικών σχέσεων ή κλάσεων, και όπως η FLogic, που δίνει μόνο βασικές δυνατότητες.
- **Συναρτήσεις.** Η FLogic τις ονομάζει μεθόδους (methods). Οι μέθοδοι είναι μια ειδική περίπτωση συναρτήσεων, οι οποίες ορίζονται στο εσωτερικό μιας έννοιας (κλάσης) και επομένως ανήκουν σε αυτή. Είναι προφανής η συγγένεια τους με τις μεθόδους του αντικειμενοστραφούς προγραμματισμού. Οι απαραίτητοι υπολογισμοί προκειμένου να εξαχθεί η

τιμή της μεθόδου για ένα συγκεκριμένο σύνολο παραμέτρων, μπορούν να περιγραφούν με τη χρήση κανόνα (rule).

- **Διαδικασίες.** Η FLogic δεν επιτρέπει τον ορισμό διαδικασιών.
- **Αντικείμενα.** Ορίζονται δίνοντας την κλάση στην οποία ανήκουν, και τις τιμές για τα πεδία που περιέχει η κλάση αυτή. Δεν είναι δυνατός ο ορισμός αντικειμένων που να ανήκουν σε περισσότερες από μία κλάσεις.
- **Αξιώματα.** Λογικά αξιώματα πρώτης τάξεως μπορούν να οριστούν, χρησιμοποιώντας σύνταξη επηρεασμένη από τον κατηγορηματικό λογισμό. Η FLogic προφανώς οφείλει την επιλογή αυτής της σύνταξης στην επιρροή από την μέθοδο των πλαισίων, ωστόσο είναι αρκετά ευανάγνωστη και εύκολη στη χρήση.
- **Κανόνες.** Υπάρχει η δυνατότητα υποστήριξης συμπερασματικών κανόνων (deductive rules). Αποτελούνται από το συμπέρασμα, που είναι ένα μοναδικό κατηγορήμα, και την υπόθεση η οποία δίνεται ως μια λογική πρόταση. Η βασικότερη προσδοκώμενη χρήση τους είναι για τον υπολογισμό τιμών των μεθόδων.

4.2.4.2 Μηχανή λογικής

Η μηχανή λογικής της FLogic βασίζεται σε ανάλυση (resolution) και σύνθεση (unification) και είναι πλήρης και ορθή (sound and complete). Μπορεί να υποστηρίξει μονοτονική και μη μονοτονική, απλή και πολλαπλή κληρονομικότητα, όπως περιγράφηκε παραπάνω. Πραγματοποιεί έλεγχο συνθηκών και περιορισμών, ενώ είναι σε θέση να υποστηρίξει συλλογισμούς με διασύνδεση προς τα εμπρός ή προς τα πίσω (forward/backward chaining).

4.2.5 LOOM

Η LOOM [MacGregor, 1991] είναι μια γλώσσα προγραμματισμού υψηλού επιπέδου, κατασκευασμένη για χρήση σε έμπειρα συστήματα (expert systems)

και άλλες εφαρμογές τεχνητής νοημοσύνης. Αναπτύχθηκε στο Ινστιτούτο Επιστήμης Πληροφορίας του Πανεπιστημίου της Νότιας Καλιφόρνια (Information Science Institute, Southern California University). Αποτελεί μια εξέλιξη της οικογένειας γλωσσών KL-ONE, και θεωρείται μια από τις πιο σημαντικές γλώσσες περιγραφικής λογικής (description logics).

Οι γλώσσες περιγραφικής λογικής διακρίνονται κυρίως για τα αποδοτικά συστήματα αυτόματης κατηγοριοποίησης. Η LOOM επιτυγχάνει επιπλέον να συγκεράσει το μοντέλο πλαισίων με το κλασσικό μοντέλο λογικών κανόνων. Έτσι συνδυάζει μια γλώσσα περιγραφής, με την οποία μπορούν να περιγραφούν αντικείμενα, σχέσεις και κλάσεις, με μια γλώσσα δήλωσης λογικών προτάσεων, με την οποία μπορούν να οριστούν συνθήκες και περιορισμοί, να καθοριστούν χαρακτηριστικά αντικειμένων, και να διατυπωθούν λογικά ερωτήματα. Το προγραμματιστικό μοντέλο που χρησιμοποιεί βασίζεται στο δομημένο προγραμματισμό (procedural programming), ενώ η μηχανή λογικής της LOOM εστιάζει στην υποστήριξη παραγωγικών κανόνων και αυτόματης κατηγοριοποίησης.

4.2.5.1.1 Αναπαράσταση γνώσης

Η LOOM προσφέρει τα ακόλουθα συστατικά δόμησης γνώσης:

- **Κλάσεις.** Η ορολογία της LOOM τις χαρακτηρίζει έννοιες (concepts) και όχι κλάσεις (classes), αλλά φυσικά αναφερόμαστε στο ίδιο δομικό συστατικό. Αντιμετωπίζονται ως μια ειδική περίπτωση σχέσης, δηλαδή μια μονομερής σχέση. Ορίζονται μέσω ικανών και αναγκαίων συνθηκών που πρέπει να ικανοποιούν αντικείμενα που ανήκουν στην κλάση. Στα πεδία μιας κλάσης μπορούν να οριστούν πολύπλοκοι περιορισμοί που αναφέρονται στον τύπο ή την πολλαπλότητα των τιμών τους, τα αποδεκτά πεδία τιμών, ακόμα και αριθμητικοί ή λογικοί περιορισμοί. Τα πεδία πρέπει να οριστούν ως σχέσεις σε άλλο σημείο

της οντολογίας και όχι μέσα στον ορισμό της κλάσης. Τέλος, η LOOM πραγματοποιεί τη διαφοροποίηση μεταξύ στοιχειωδών (primitive) και μη στοιχειωδών (non-primitive) κλάσεων, η οποία αφορά την αυτόματη κατηγοριοποίηση και έχει ήδη εξηγηθεί παραπάνω.

- **Ταξινομίες.** Οι ταξινομίες στη LOOM υλοποιούνται με τη χρήση του τελεστή `:is-primitive`, το οποίο είναι ισοδύναμο από πλευράς λειτουργικότητας με το γνωστό `subclass-of`. Είναι δυνατός ο ορισμός πολλαπλών υπερκλάσεων για μια κλάση. Υπάρχουν ειδικά ορίσματα για τη δήλωση κατατμήσεων, τόσο αμοιβαία αποκλειόμενων όσο και εξαντλητικών. Επίσης, υπάρχει το όρισμα `not subclass of` το οποίο χρησιμοποιείται για να δηλώσει τη μη υπαγωγή μιας κλάσης σε κάποια άλλη. Πρέπει τέλος να σημειωθεί ότι μια ταξινόμια μπορεί να εξαχθεί αυτόματα από τη LOOM μέσω της δυνατότητας αυτόματης κατηγοριοποίησης, ανάλογα με τους περιορισμούς που έχουν διατυπωθεί στον ορισμό κάθε κλάσης.
- **Σχέσεις.** Η LOOM προσφέρει ειδικά ορίσματα για ορισμό διμερών σχέσεων, ωστόσο ο ορισμός πολυμερών σχέσεων είναι άμεσα δυνατός. Για κάθε σχέση μπορεί να οριστεί ένα σύνολο χαρακτηριστικών, το οποίο περιλαμβάνει τον τύπο των παραμέτρων της, ικανές και αναγκαίες συνθήκες που πρέπει να ικανοποιηθούν για να ισχύει η σχέση, ανάστροφες σχέσεις και άλλα.
- **Συναρτήσεις.** Όπως οι κλάσεις, έτσι και οι συναρτήσεις υλοποιούνται ως ειδική περίπτωση σχέσεων. Οι περιορισμοί στις παραμέτρους είναι ίδιοι με αυτούς που ισχύουν για τις σχέσεις. Επιπρόσθετα, στην περίπτωση των συναρτήσεων μπορεί να χρησιμοποιηθεί ο τελεστής `:function`, με το οποίο ορίζεται η έκφραση που πρέπει να υπολογιστεί για να προσδιοριστεί το αποτέλεσμα της συνάρτησης.

- **Διαδικασίες.** Η LOOM επιτρέπει τον ορισμό διαδικασιών σε γλώσσα Lisp.
- **Αντικείμενα.** Υπάρχουν δύο τρόποι να εισαχθεί ένα αντικείμενο σε μια οντολογία σε LOOM. Ο πρώτος είναι ο συμβατικός, σύμφωνα με τον οποίο καθορίζεται η κλάση στην οποία ανήκει το αντικείμενο και οι τιμές κάποιων ή όλων των πεδίων του. Αυτός ο τρόπος χρησιμοποιείται σε όλες τις γλώσσες που δεν υποστηρίζουν αυτόματη κατηγοριοποίηση. Ο δεύτερος τρόπος, που υποστηρίζεται μόνο σε γλώσσες που προσφέρουν αυτόματη κατηγοριοποίηση, συνίσταται στη δήλωση ενός συνόλου προτάσεων που ισχύουν και αφορούν το νέο αντικείμενο. Αυτές οι προτάσεις μπορούν να περιλαμβάνουν και τιμές των πεδίων του. Η δήλωση των προτάσεων πραγματοποιείται μέσω του τελεστή tell. Με βάση τις ισχύουσες προτάσεις, η μηχανή αυτόματης κατηγοριοποίησης της LOOM επιχειρεί την κατάταξη του αντικειμένου σε μια από τις κλάσεις της ταξινόμιας.
- **Αξιώματα.** Η δήλωση αξιωμάτων ως αυτόνομων οντοτήτων στη LOOM δεν είναι δυνατή. Είναι όμως δυνατό να ορίσει κανείς αξιώματα στα πλαίσια του ορισμού κλάσεων και σχέσεων. Τα αξιώματα χρησιμοποιούν λογική πρώτης τάξεως για τον ορισμό χαρακτηριστικών και περιορισμών των πεδίων της κλάσης ή παραμέτρων της σχέσης, καθώς και των ίδιων των κλάσεων ή σχέσεων.
- **Κανόνες.** Η LOOM υποστηρίζει δύο είδη κανόνων, τους παραγωγικούς (productive) και τους συμπερασματικούς (deductive). Οι συμπερασματικοί κανόνες περιέχονται στον ορισμό των κλάσεων και των σχέσεων με τη χρήση των ορισμάτων is ή is-primitive. Οι παραγωγικοί κανόνες ορίζονται ανεξάρτητα και δίνουν τη δυνατότητα

να οριστεί ένα σύνολο από ενέργειες που εκτελούνται όταν ικανοποιείται η υπόθεση τους.

4.2.5.2 Μηχανή λογικής

Όπως αναφέρθηκε και στην εισαγωγή, το ειδοποιό χαρακτηριστικό των γλωσσών περιγραφικής λογικής (description logics) είναι η υποστήριξη τους για αυτόματη κατηγοριοποίηση. Η LOOM παρέχει μια τέτοια λειτουργικότητα, που δίνει τη δυνατότητα αυτόματης κατηγοριοποίησης αντικειμένων με βάση τους περιορισμούς στα πεδία των κλάσεων που έχουν οριστεί. Η ίδια λειτουργικότητα δίνει και τη δυνατότητα προσδιορισμού άγνωστων τιμών κάποιων πεδίων του αντικειμένου. Η μηχανή λογικής της LOOM υποστηρίζει και τις πιο συμβατικές διεργασίες, όπως η κληρονομικότητα πεδίων κατά μήκος της ταξινόμιας, ο έλεγχος συνθηκών και περιορισμών, ο εντοπισμός ασυνεπειών στους ορισμούς τους κλπ.

4.2.6 CARIN

Η γλώσσα CARIN [Levy, 1998] είναι ένας συνδυασμός μιας γλώσσας με κανόνες Datalog (μη επαναλαμβανόμενοι κανόνες Horn χωρίς συναρτήσεις) με την γλώσσα περιγραφικής λογικής ALN. Η ALN είναι μια απλή γλώσσα περιγραφικής λογικής, που υποστηρίζει τελεστές σύνδεσης εννοιών, περιορισμού τύπου (forall), ποσοτικούς (atleast, atmost), καθώς και την άρνηση.

4.2.6.1 Αναπαράσταση γνώσης

Κάθε οντολογία σε CARIN απαρτίζεται από δύο συστατικά:

- Ένα ορολογικό τμήμα, στο οποίο συγκεντρώνονται οι ορισμοί εννοιών, και πιθανώς ένα σύνολο από συνθήκες υπαγωγής (που έχουν μορφή αξιωμάτων και αφορούν ατομικές έννοιες μόνο) ή αμοιβαίου αποκλεισμού μεταξύ ατομικών εννοιών.

- Ένα τμήμα κανόνων, που απαρτίζεται από ένα σύνολο κανόνων στους οποίους οι έννοιες μπορούν να εμφανίζονται στην υπόθεση, αλλά όχι στο συμπέρασμα.

Έτσι, οι οντολογίες της CARIN χρησιμοποιούν δομικά συστατικά ως εξής:

- **Κλάσεις.** Μια κλάση, ή έννοια (concept) στην ορολογία της CARIN, ορίζεται από το όνομα της και μια περιγραφή, η οποία περιέχει συνθήκες υπαγωγής (που καθορίζουν αναγκαίες και ικανές συνθήκες για να ανήκει ένα αντικείμενο στην κλάση), αξιώματα που ισχύουν για τα αντικείμενα που ανήκουν στην κλάση καθώς και ένα σύνολο από περιορισμούς στον τύπο και στην πολλαπλότητα των πεδίων της κλάσης. Τα πεδία είναι πάντα πεδία των αντικειμένων (instance attributes), δηλαδή η τιμή τους καθορίζεται για κάθε αντικείμενο ανεξάρτητα. Ο ορισμός πεδίων κλάσης (class attributes), των οποίων η τιμή είναι ίδια για όλα τα αντικείμενα της κλάσης, δεν υποστηρίζεται.
- **Ταξινομίες.** Η CARIN προσφέρει επαρκή υποστήριξη για ταξινομίες, επιτρέποντας πολλαπλή κληρονομικότητα και ορισμό αμοιβαία αποκλειόμενων κατατμήσεων. Επίσης, υπάρχει το όρισμα not subclass of που δηλώνει τη μη υπαγωγή σε κλάση, και μπορεί να εφαρμοστεί σε ατομικές κλάσεις.
- **Σχέσεις.** Πολυμερείς σχέσεις ορίζονται με απλή σύνταξη, ωστόσο δεν είναι δυνατή η περιγραφή συνθηκών και περιορισμών που τις διέπουν.
- **Συναρτήσεις.** Οι συναρτήσεις δεν υποστηρίζονται.
- **Διαδικασίες.** Ομοίως, η CARIN δεν δίνει δυνατότητα περιγραφής διαδικασιών.
- **Αντικείμενα.** Κάθε αντικείμενο ορίζεται στην CARIN με ένα κατηγορημα που έχει το όνομα της κλάσης και μια παράμετρο που

αντιπροσωπεύει το όνομα του αντικειμένου. Αντίστοιχα αναπαριστώνται και οι σχέσεις, με ένα κατηγορημα που δίνει το όνομα της σχέσης και μια λίστα με παραμέτρους που αντιπροσωπεύουν τα μέρη της.

- **Αξιώματα.** Δεν υπάρχει δυνατότητα καθορισμού αυτόνομων αξιωμάτων.
- **Κανόνες.** Η CARIN υποστηρίζει, όπως αναφέρθηκε στην εισαγωγή, κανόνες τύπου Horn οι οποίοι περιέχουν κλάσεις, αντικείμενα ή πεδία στην υπόθεση, και κατηγορήματα στο συμπέρασμα. Ουσιαστικά πρόκειται για παραγωγικούς κανόνες, που δίνουν τη δυνατότητα περίληψης νέων κατηγορημάτων στη βάση γνώσης.

4.2.6.2 Μηχανή λογικής

Το πρόβλημα πραγματοποίησης λογικών συνειρμών σε CARIN έχει μελετηθεί εκτεταμένα. Η CARIN είναι εξοπλισμένη με πλήρεις και ορθές διαδικασίες αυτόματης κατηγοριοποίησης. Η μηχανή της μπορεί να πραγματοποιήσει ακόμα διασύνδεση προς τα εμπρός και προς τα πίσω, και να διαχειριστεί σωστά τις εξαιρέσεις. Ωστόσο, δεν υπάρχει δυνατότητα διαχείρισης μη μονοτονικής κληρονομικότητας.

4.3 ΓΛΩΣΣΕΣ ΟΝΤΟΛΟΓΙΩΝ ΒΑΣΙΣΜΕΝΕΣ ΣΕ XML

Η εκρηκτική ανάπτυξη του παγκόσμιου ιστού (World Wide Web) τον μετέτρεψε γρήγορα από μια απλή πλατφόρμα ροής πληροφορίας σε ένα μέσο ανταλλαγής γνώσης. Στο πλαίσιο αυτό οι οντολογίες μπορούν να παίξουν θεμελιώδη ρόλο, αφού έχουν τη δυνατότητα να κωδικοποιήσουν τη σημασιολογία της πληροφορίας, επιτρέποντας έτσι την ανταλλαγή γνώσης ανάμεσα σε ετερογενή συστήματα.

Έτσι ήρθε στο προσκήνιο η ανάγκη για τη δημιουργία γλωσσών αναπαράστασης γνώσης και περιγραφής οντολογιών, οι οποίες να είναι συμβατές με τα πρότυπα του Internet και του παγκόσμιου ιστού (WWW). Στη συνέχεια εξετάζουμε τις βασικότερες από αυτές τις γλώσσες (XML, RDF, XOL, SHOE and OIL), αρχίζοντας με την XML που αποτελεί και ιστορικά την πρώτη, στην οποία οι υπόλοιπες βασίστηκαν.

Η XML [Bray, 1998] (eXtensible Markup Language) προέρχεται από την SGML (Standard General Markup Language). Η ανάπτυξη της πραγματοποιήθηκε από την ομάδα εργασίας XML (XML Working Group) του World Wide Web Consortium (W3C), με γνώμονα την ευκολία στη χρήση και τη διαλειτουργικότητα με την SGML και την HTML.

Πρόκειται για μια γλώσσα σήμανσης (markup) που στοχεύει στην ανταλλαγή δεδομένων μεταξύ εφαρμογών μέσω του Internet. Τα βασικότερα πλεονεκτήματα της στο πλαίσιο του παγκόσμιου ιστού είναι η καλά ορισμένη και συμπαγής σύνταξη, η ευκολία στην ανάλυση (parsing), η καλή αναγνωσιμότητα της και η εύκολη επεκτασιμότητα. Η XML δίνει τη δυνατότητα στους χρήστες να ορίσουν τις δικές τους ετικέτες (tags) και να δημιουργήσουν δομές δεδομένων εμφωλιάζοντας τις. Είναι δυνατός ο φορμαλιστικός ορισμός της δομής ενός εγγράφου XML με χρήση διαφόρων προτύπων (XSD, DTD) και ο έλεγχος της δομικής ορθότητας του στη συνέχεια.

Αν και σίγουρα αυτή δεν ήταν η προσδοκώμενη χρήση κατά τη δημιουργία της, η XML μπορεί να χρησιμοποιηθεί για τον ορισμό οντολογιών. Με τον τρόπο αυτό, επιτυγχάνεται εξαιρετικά μεγάλη αναγνωσιμότητα, αφού ο χρήστης χρησιμοποιεί ένα υπάρχον, διαδομένο πρότυπο και δεν χρειάζεται να μάθει μια νέα γλώσσα. Τα δομικά υλικά μιας οντολογίας μπορούν να οριστούν με χρήση DTD ή XSD, ενώ η περιγραφή της οντολογίας μπορεί να μεταδοθεί σε XML μέσω του παγκόσμιου ιστού χωρίς προβλήματα κωδικοποίησης, ασφαλείας κλπ.

Τα μειονεκτήματα στη χρήση της XML για περιγραφή οντολογιών περιλαμβάνουν τις χαλαρές συντακτικές απαιτήσεις της XML, που επιτρέπουν την περίληψη αδόμητης πληροφορίας ανάμεσα σε ετικέτες και καθιστούν έτσι δύσκολη την αναζήτηση οντολογικών όρων στο εσωτερικό ενός εγγράφου, και κυρίως την έλλειψη εργαλείων που να υποστηρίζουν λογικούς συνειρμούς σε XML.

Συμπερασματικά, η XML μπορεί κάλλιστα να χρησιμοποιηθεί για την κωδικοποίηση και ανταλλαγή οντολογιών, ωστόσο η εύκολη χρήση της απαιτεί επεκτάσεις και ειδικά εργαλεία. Έτσι, έχει αποτελέσει τη βάση για όλες τις γλώσσες που παρουσιάζονται στη συνέχεια και αποσκοπούν στη συμβατότητα με τον παγκόσμιο ιστό, με μοναδική εξαίρεση τις πρώτες εκδόσεις της SHOE.

4.3.1 RDF(S)

Το ακρώνυμο RDF [Lassila, 1999] αντιπροσωπεύει την ονομασία Resource Description Framework, ή πλαίσιο περιγραφής πόρων. Πρόκειται για ένα πρότυπο που αναπτύσσεται από το W3C με στόχο τη δημιουργία μιας γλώσσας βασισμένης σε XML για την περιγραφή μετα-δεδομένων που περιγράφουν πληροφορία και άλλους πόρους του παγκόσμιου ιστού. Υπάρχει μια συμπληρωματική σχέση μεταξύ XML και RDF. Ένας από τους βασικούς στόχους της δεύτερης είναι ο ορισμός της σημασιολογίας δεδομένων που ανταλλάσσονται σε XML με τυποποιημένο τρόπο, χρησιμοποιώντας την ίδια την XML.

Ο γενικότερος στόχος της RDF είναι η δημιουργία ενός μηχανισμού περιγραφής πόρων που δεν επιβάλλει υποθέσεις σχετικά με την περιοχή εφαρμογής ή τη δομή ενός εγγράφου που περιέχει πληροφορία. Το δομικό μοντέλο της RDF ορίζει τρεις τύπους αντικειμένων:

- Τους πόρους (resources) ή υποκείμενα (subjects), δηλαδή οποιαδήποτε οντότητα μπορεί να προσδιοριστεί με τη χρήση μιας

διεύθυνσης του παγκοσμίου ιστού. Οι διευθύνσεις πόρων είναι γνωστές ως URIs (Uniform Resource Identifiers).

- Ιδιότητες (properties) ή κατηγορήματα (predicates), που ορίζουν χαρακτηριστικά, ιδιότητες ή σχέσεις που αφορούν κάποιον πόρο.
- Δηλώσεις (statements) ή αντικείμενα (objects), που αποδίδουν μια συγκεκριμένη τιμή σε μια ιδιότητα κάποιου πόρου.

Η RDF δανείζεται αυτό το φορμαλισμό από τη θεωρία των σημασιολογικών δικτύων (semantic nets) [Staab, 00].

Η RDF δεν παρέχει από μόνη της κάποιο τρόπο για να οριστούν οι σχέσεις ανάμεσα σε ιδιότητες και πόρους. Αυτός είναι ο ρόλος της RDFS (RDF Schema specification language) [Brickley, 1999]. Πρόκειται για μια δομημένη γλώσσα που χρησιμοποιείται για να περιγράψει δομές σε RDF (RDF schemas), συσχετίζοντας τις ιδιότητες με τη σημασιολογία τους. Η RDFS προσφέρει στοιχεία μοντελοποίησης γνώσης που έχουν μεγάλη συγγένεια με αυτά που χρησιμοποιούνται στις βασισμένες σε πλαίσια γλώσσες.

Ο συνδυασμός της RDF με την RDFS αναφέρεται συνήθως ως RDF(S) και αποτελεί μια γλώσσα που μπορεί να περιγράψει γνώση και οντολογίες [Amann, 1999]. Αν και είναι βασισμένη σε κάποιες αρχές αναπαράστασης γνώσης (σημασιολογικά δίκτυα, πλαίσια και κατηγορηματικό λογισμό), είναι πολύ απλούστερη στην υλοποίηση από τις κατεξοχήν γλώσσες κατηγορηματικού λογισμού, όπως η KIF ή η CycL. Τα βασικά στοιχεία που χρησιμοποιεί για την μοντελοποίηση γνώσης είναι κλάση (class), ο πόρος (resource) και η ιδιότητα (property). Ιεραρχίες και βασικοί περιορισμοί μπορούν να οριστούν με τη βοήθεια διαφόρων ορισμάτων (subclassOf, subPropertyOf, seeAlso, isDefinedBy).

4.3.1.1 Αναπαράσταση γνώσης

Η RDF(S) προσπαθεί να επιτύχει μια ισορροπία ανάμεσα στην εκφραστικότητα και την απλότητα, συνέπεια της οποίας είναι ότι παρέχει μόνο τα πλέον στοιχειώδη δομικά συστατικά για την περιγραφή γνώσης. Η πρόθεση είναι τα στοιχεία αυτά να αποτελέσουν ένα βασικό πυρήνα που να μπορεί να καλύψει πλήρως την περιγραφή πόρων στο Internet αλλά και να είναι εύκολα επεκτάσιμα ώστε να μπορούν να αξιοποιηθούν και σε άλλες χρήσεις. Τα στοιχεία αυτά είναι τα εξής:

- **Κλάσεις.** Ο ορισμός μιας κλάσης στην RDF(S) γίνεται με την απόδοση ενός ονόματος και τη συσχέτιση του με ένα συγκεκριμένο URI (Uniform Resource Identifier). Μια περιγραφή της κλάσης σε φυσική γλώσσα μπορεί να περιληφθεί στον ορισμό. Επίσης, μπορεί να καθορίσει κανείς τις υπερκλάσεις της και τους τυχόν πόρους με τους οποίους σχετίζεται. Τα πεδία της κλάσης πρέπει να οριστούν ξεχωριστά ως ιδιότητες. Η θέση περιορισμών πολλαπλότητας στις τιμές των πεδίων δεν είναι δυνατή.
- **Ταξινομίες.** Η δημιουργία ταξινομιών υποστηρίζεται ως μέρος του ορισμού κλάσεων, με τη χρήση του τελεστή `subclassOf`. Για κάθε κλάση υπάρχει η δυνατότητα ορισμού πολλών υπερκλάσεων, δίνοντας έτσι τη δυνατότητα χρήσης πολλαπλής κληρονομικότητας. Η RDF(S) δεν υποστηρίζει κατατμήσεις (partitions), αν και έχουν γίνει κάποιες προσπάθειες για την επέκτασή της ώστε να τις συμπεριλάβει [Staab, 00].
- **Σχέσεις.** Οι σχέσεις δεν υποστηρίζονται εγγενώς από την RDF(S). Υπάρχει η δυνατότητα ορισμού διμερών σχέσεων με χρήση ιδιοτήτων. Ωστόσο, ο ορισμός πολυμερών σχέσεων παρουσιάζει κάποιες δυσκολίες και είναι δυνατός μόνο με ανάλυση τους σε διμερείς σχέσεις. Το μοντέλο αυτό είναι ιδιαίτερα δύσχρηστο στην πράξη. Η

υλοποίηση των σχέσεων με ιδιότητες δίνει την δυνατότητα ορισμού ιεραρχιών σχέσεων, αφού μια ιδιότητα μπορεί να οριστεί ως ειδική περίπτωση (subproperty) κάποιας άλλης.

- **Συναρτήσεις.** Η RDF(S) δεν υποστηρίζει συναρτήσεις. Ωστόσο είναι δυνατό να οριστούν ως ειδικές περιπτώσεις σχέσεων (που, με τη σειρά τους, ορίζονται ως ειδικές περιπτώσεις ιδιοτήτων). Ακόμα και τότε όμως δεν είναι δυνατή η θέση περιορισμών που να εξασφαλίζουν τη μοναδικότητα της τιμής της συνάρτησης για ένα δεδομένο διάνυσμα εισόδων, ούτε και ο υπολογισμός της τιμής αυτής.
- **Διαδικασίες.** Δεν υποστηρίζονται από την RDF(S).
- **Αντικείμενα.** Υπάρχουν διάφοροι τρόποι για τον ορισμό αντικειμένων, οι διαφορές μεταξύ τους όμως είναι καθαρά συντακτικές. Η κλάση στην οποία ανήκει το αντικείμενο, η οποία μπορεί να είναι μόνο μία, πρέπει σε κάθε περίπτωση να οριστεί μαζί με ένα σύνολο πεδίων και των τιμών τους.
- **Αξιώματα.** Τα αξιώματα δεν περιλαμβάνονται στο δομικό μοντέλο της RDF(S), ωστόσο κάποια μελέτη έχει πραγματοποιηθεί για τη δυνατότητα περίληψης τους [Staab, 00]. Η έρευνα αυτή προτείνει την περίληψη προτάσεων σε FLogic, οι οποίες θα αποτελέσουν το σώμα των αξιωμάτων. Με τον τρόπο αυτό, μια μηχανή λογικής για FLogic θα μπορούσε να αντλήσει πληροφορία από ένα έγγραφο RDF(S) και να πραγματοποιήσει κάποιους λογικούς συνειρμούς πάνω σε αυτή. Η παραπάνω προσέγγιση έχει το μειονέκτημα της εξάρτησης της RDF(S) από την FLogic, η οποία είναι πολύ λιγότερο δημοφιλής και διαδεδομένη από την πρώτη. Έτσι, δεν έχει βρει πρόσφορο έδαφος.
- **Κανόνες.** Δεν υπάρχει η δυνατότητα ορισμού κανόνων στην RDF(S).

4.3.1.2 Μηχανή λογικής

Η RDF(S) δεν υποστηρίζει κανενός είδους λογικούς συνειρμούς, δηλαδή δεν υπάρχει κάποια μηχανή λογικής η οποία να διαβάσει έγγραφα RDF(S) και να πραγματοποιεί λογικές διεργασίες με την πληροφορία που περιέχουν. Ωστόσο, χάρη στην εύγλωπτη σύνταξη της RDF, η οποία είναι φτιαγμένη εξάλλου για το σκοπό αυτό, είναι δυνατή η μετατροπή της γνώσης RDF(S) σε κάποια άλλη γλώσσα, για παράδειγμα FLogic, και η χρήση της λογικής μηχανής της γλώσσας αυτής.

Έχουν πραγματοποιηθεί κάποιες προσπάθειες για την υλοποίηση μιας μηχανής που να δουλεύει απευθείας πάνω σε RDF(S) [Decker, 1999]. Και αυτή στηρίζεται στην FLogic προκειμένου να πραγματοποιήσει λογικούς συνειρμούς με οντολογίες RDF(S).

4.3.2 XOL

Η XOL (XML-Based Ontology Exchange Language) [Karp, 1999] σχεδιάστηκε σαν ένα μέσο για την ανταλλαγή οντολογιών ανάμεσα σε ετερογενή συστήματα. Έτσι, δεν είναι κατασκευασμένη για την ανάπτυξη οντολογιών, αλλά για χρήση σαν ενδιάμεση γλώσσα μεταφοράς οντολογιών ανάμεσα σε διαφορετικές εφαρμογές, εργαλεία ανάπτυξης οντολογιών ή βάσεις δεδομένων.

Η XOL δημιουργήθηκε στα πλαίσια της εξέλιξης της οντολογίας στο χώρο της βιοπληροφορικής [McEntire, 1999]. Προέκυψε από τη μελέτη των αναγκών αναπαράστασης γνώσης στο χώρο αυτό, και την εξέταση υπό αυτό το πρίσμα διαφόρων γλωσσών περιγραφής οντολογιών. Η μελέτη αρχικά κατέδειξε την Ontolingua [Farquhar, 1996] και την OML [Kent, 1998] ως κατάλληλες γλώσσες για το σκοπό αυτό, ωστόσο και οι δύο παρουσίαζαν μειονεκτήματα. Η μεν Ontolingua υστερούσε σε αναγνωσιμότητα και ευχρηστία, λόγω της βασισμένης στη Lisp σύνταξη της, ενώ η δε OML, που είναι βασισμένη στην XML, υστερούσε

σε εκφραστικότητα. Έτσι προέκυψε η ανάγκη δημιουργίας της XOL ως συμβιβασμό μεταξύ των δύο, ο οποίος να συνδυάζει πλεονεκτήματα και από τις δύο πλευρές – την υψηλή εκφραστικότητα της Ontolingua με την ευχρηστία και τη συμβατότητα με XML της OML. Στα πλαίσια του συμβιβασμού αυτών των αντικρουόμενων στόχων, υιοθέτησε το μοντέλο γνώσης της OKBC [Chaudhri, 1998], που αποτελεί στην ουσία μια απλουστευμένη μορφή του μοντέλου της Ontolingua, περιορίζοντας έτσι την εκφραστικότητα της γλώσσας ως ένα σημείο.

Τελικά, η XOL επιτρέπει την αναφορά σε ένα μοντέλο γνώσης που είναι υποσύνολο του OKBC και ονομάζεται OKBC-Lite. Καθώς το OKBC είναι κατασκευασμένο για την ανταλλαγή πληροφορίας ανάμεσα σε συστήματα βασισμένα σε πλαίσια, η XOL είναι ιδιαίτερα κατάλληλη για την ανταλλαγή πληροφορίας μέσω του WWW, συνδυάζοντας έτσι τα πλεονεκτήματα και της XML και του OKBC. Το κυριότερο μειονέκτημα της είναι ίσως η όχι καλή υποστήριξη των πλαισίων. Τα πλαίσια, στα οποία βασίζεται το μοντέλο του OKBC, δεν ορίζονται απευθείας στην XOL, η οποία υποστηρίζει μόνο κλάσεις και ιεραρχίες. Ακόμη, δεν υπάρχουν ολοκληρωμένα περιβάλλοντα ή εργαλεία για την ανάπτυξη οντολογιών σε XOL. Ωστόσο οποιαδήποτε εφαρμογή επεξεργασίας κειμένου ή σύνταξης XML μπορεί να χρησιμοποιηθεί.

4.3.2.1 Αναπαράσταση γνώσης

Η XOL παρέχει τα ακόλουθα δομικά συστατικά για την κατασκευή οντολογιών:

- **Κλάσεις.** Μια κλάση ορίζεται από ένα μοναδικό όνομα. Ο ορισμός της κλάσης μπορεί να περιέχει πεδία, αυτά όμως μπορούν και σε άλλο σημείο της οντολογίας. Υπάρχει η δυνατότητα ορισμού μετα-κλάσεων, δηλαδή κλάσεων που περιέχουν άλλες κλάσεις. Τα πεδία μπορούν να διακριθούν σε πεδία της κλάσεως και πεδία αντικειμένων. Ο ορισμός τους μπορεί επίσης να περιέχει μια περιγραφή σε φυσική γλώσσα,

περιορισμούς στον τύπο και την πολλαπλότητα των τιμών καθώς και στην ελάχιστη και μέγιστη τιμή για αριθμητικά πεδία. Είναι επίσης δυνατό να αποδοθούν τιμές στα πεδία.

- **Ταξινομίες.** Η XOL παρέχει τη βασική υποστήριξη για ταξινομίες, οι οποίες περιγράφονται με τη βοήθεια του τελεστή subclass of, που περιλαμβάνεται στον ορισμό κάθε κλάσης. Η πολλαπλή κληρονομικότητα επιτρέπεται.
- **Σχέσεις.** Δεν είναι δυνατός ο κατευθείαν καθορισμός πολυμερών σχέσεων ανάμεσα σε κλάσεις, αφού ο ορισμός σχέσεων δεν υποστηρίζεται άμεσα. Διμερείς σχέσεις μπορούν να οριστούν μέσω πεδίων (slots). Έτσι, πολυμερείς σχέσεις μπορούν να οριστούν μόνο μέσω κλάσεων κατασκευασμένων για αυτό το σκοπό (associative classes) ή αναλύοντας τις σε διμερείς σχέσεις.
- **Συναρτήσεις.** Δεν υπάρχει υποστήριξη συναρτήσεων.
- **Διαδικασίες.** Οι διαδικασίες επίσης δεν υποστηρίζονται στην XOL.
- **Αντικείμενα.** Τα αντικείμενα ορίζονται από ένα όνομα, προαιρετική περιγραφή σε φυσική γλώσσα, την κλάση στην οποία ανήκουν και ένα σύνολο από ονόματα πεδίων και τις αντίστοιχες τιμές τους.

4.3.2.2 Μηχανή λογικής

Δεν υπάρχει μηχανή λογικής για την XOL. Αυτό οφείλεται εν μέρει στο γεγονός ότι η XOL έχει κατασκευαστεί με κύριο στόχο την μεταφορά οντολογιών, και όχι την ανάπτυξη τους.

4.3.3 SHOE

Η SHOE [Luke, 2000] (Simple HTML Ontology Extension) είναι μια γλώσσα βασισμένη σε HTML που αναπτύχθηκε στο Πανεπιστήμιο του Maryland και έχει αποτελέσει της βάση για την ανάπτυξη της OML/CKML [Kent, 1998]. Ο σκοπός

της ανάπτυξης της ήταν να αποτελέσει ένα απλό μέσο προσθήκης σημασιολογικής πληροφορίας στις σελίδες HTML του παγκόσμιου ιστού, η οποία να είναι αναγνώσιμη από υπολογιστή. Η κύρια προσδοκώμενη χρήση της είναι να δώσει τη δυνατότητα σε ευφυείς πράκτορες (intelligent agents) να αντλήσουν πληροφορία από σελίδες στον ιστό με βάση τη σημασιολογία τους, βελτιώνοντας έτσι δραστικά την ακρίβεια των μεθόδων αναζήτησης και συγκέντρωσης γνώσης. Αυτό επιτυγχάνεται με τα ακόλουθα τρία βήματα:

1. Κατ' αρχήν δημιουργείται μια οντολογία η οποία περιγράφει την κατηγοριοποίηση των αντικειμένων υπό εξέταση, καθώς και τις μεταξύ τους σχέσεις.
2. Στη συνέχεια, οι σελίδες HTML επαυξάνονται με την προσθήκη σημασιολογικών σημειώσεων (annotations).
3. Τέλος, ένας πράκτορας ή μια μηχανή αναζήτησης συγκρίνει την σημασιολογική πληροφορία της σημείωσης με το αντικείμενο της αναζήτησης του, προκειμένου να συμπεράνει σε ποιο βαθμό ανταποκρίνεται σε αυτή.

4.3.3.1 Αναπαράσταση γνώσης

Στη SHOE, μια οντολογία είναι μια απλή ιεραρχία από κλάσεις, που ονομάζονται κατηγορίες (categories), σε συνδυασμό με ένα σύνολο ατομικών σχέσεων μεταξύ τους και ένα σύνολο από κανόνες σε μορφή απλοποιημένων προτάσεων Horn. Έτσι, οι δυνατότητες μοντελοποίησης είναι οι ακόλουθες:

- **Κλάσεις.** Ορίζονται σαν κατηγορίες οι οποίες χρησιμοποιούνται για την κατηγοριοποίηση αντικειμένων που βρίσκονται στις διάφορες σελίδες του ιστού. Εκτός από το όνομα και τις υπερκλάσεις της, ο ορισμός μιας κλάσης μπορεί να περιλάβει και μια μικρή περιγραφή, που μπορεί στη συνέχεια να αξιοποιηθεί από πράκτορες. Η πολλαπλή κληρονομικότητα υποστηρίζεται, αλλά δεν υπάρχει δυνατότητα

ορισμού μετα-κλάσεων ή κατατημήσεων. Τα πεδία δεν ορίζονται μέσα στις κλάσεις αλλά ξεχωριστά, σαν διμερείς σχέσεις. Πεδία κλάσης, περιορισμοί πολλαπλότητας και αρχικές τιμές δεν μπορούν να οριστούν στη SHOΕ.

- **Ταξινομίες.** Οι ταξινομίες ορίζονται με τη χρήση του τελεστή is-a, που είναι ισοδύναμος του συνηθέστερου subclass of. Η πολλαπλή κληρονομικότητα υποστηρίζεται.
- **Σχέσεις.** Η SHOΕ δίνει τη δυνατότητα ορισμού πολυμερών σχέσεων. Για κάθε παράμετρο της σχέσης, μπορεί να οριστεί η κλάση ή ο τύπος του.
- **Συναρτήσεις.** Οι συναρτήσεις δεν υποστηρίζονται άμεσα από το μοντέλο της SHOΕ. Μπορούν να οριστούν ως ειδικές περιπτώσεις σχέσεων, ωστόσο αυτός ο τρόπος έχει το σύνηθες μειονέκτημα της αδυναμίας εξασφάλισης της μοναδικότητας της τιμής της συνάρτησης για συγκεκριμένο διάνυσμα παραμέτρων.
- **Διαδικασίες.** Η SHOΕ δεν δίνει δυνατότητα καθορισμού διαδικασιών.
- **Αντικείμενα.** Όπως ήδη αναφέρθηκε παραπάνω, τα αντικείμενα που περιγράφει μια οντολογία SHOΕ βρίσκονται συνήθως διεσπαρμένα σε διάφορες σελίδες του παγκόσμιου ιστού. Τα αντικείμενα περιγράφονται με βάση οντολογικά στοιχεία (κατηγορίες, σχέσεις και κανόνες) τα οποία βρίσκονται σε κάποιο κεντρικό εξυπηρετητή οντολογιών (ontology server). Ένα αντικείμενο ορίζεται καθορίζοντας την κατηγορία στην οποία ανήκει και τις τιμές των πεδίων της κατηγορίας αυτής. Τα αντικείμενα μπορούν να ανήκουν σε κλάσεις (class instances) ή σε σχέσεις (relation instances ή facts). Επίσης, υπάρχει η δυνατότητα εμφωλιασμού αντικειμένων (nesting). Αυτό είναι ένα σημαντικό χαρακτηριστικό της SHOΕ, το οποίο της επιτρέπει

τη διαχείριση αντικειμένων που ορίζονται σε διεσπαρμένα στον ιστό σημεία και δεν υπόκεινται σε κεντρικό έλεγχο. Έτσι, η SHOE αντιμετωπίζει την πληροφορία που παρέχει κάθε αντικείμενο ως ισχυρισμό (claim), ο οποίος δεν είναι απαραίτητα καθολικά αποδεκτός. Η δυνατότητα εμφωλιασμού επιτρέπει την έκφραση αντιφατικών ισχυρισμών, χωρίς αυτό να οδηγεί σε ασυνέπειες.

- **Αξιώματα.** Η SHOE δεν παρέχει δυνατότητες ορισμού αξιωμάτων.
- **Κανόνες.** Η SHOE παρέχει τη δυνατότητα ορισμού συμπερασματικών κανόνων που ορίζονται σαν προτάσεις Horn, με κάποιους επιπλέον περιορισμούς (δεν πρέπει να αναφέρονται ελεύθερες μεταβλητές στο δεξιό μέρος, και οι επιμέρους προτάσεις του δεξιού μέρους πρέπει να συνδέονται με σύζευξη (conjunction) και όχι με διάζευξη (disjunction)).

4.3.3.2 Μηχανή λογικής

Δεν υπάρχει μηχανή λογικής για τη SHOE, ωστόσο έχει πραγματοποιηθεί κάποια μελέτη στον τομέα της προδιαγραφής της, όπως η κληρονομικότητα και ο μηχανισμός διασύνδεσης προς τα πίσω.

4.3.4 OIL

Η OIL [Horrocks; 00] (Ontology Interchange Language) είναι μια προσέγγιση στη δημιουργία ενός διπλού προτύπου, τόσο για τη δημιουργία όσο και για την μεταφορά οντολογιών. Έχει σχεδιαστεί ώστε να καλύπτει τα περισσότερα χαρακτηριστικά τόσο των γλωσσών πλαισίων, όσο και των γλωσσών περιγραφικής λογικής (*description logic*).

Συγκεκριμένα, η OIL δανείζεται από τις προσεγγίσεις περιγραφικής λογικής τη φορμαλιστική σημασιολογία και τις δυνατότητες αποδοτικών λογικών συνειρμών. Στις γλώσσες περιγραφικής λογικής, η γνώση περιγράφεται με

έννοιες και περιορισμούς (ή ρόλους), που χρησιμοποιούνται για τη δημιουργία ταξινομιών, όπως για παράδειγμα στην περίπτωση της *LOOM*. Χρησιμοποιούν ένα υποσύνολο της λογικής πρώτης τάξεως, το οποίο διατηρεί αρκετά υψηλή εκφραστικότητα, ενώ ταυτόχρονα επιτρέπει την κατασκευή αποδοτικών μηχανών λογικής.

Από τις γλώσσες πλαισίων η *OIL* δανείζεται τις βασικές αρχές μοντελοποίησης. Ορίζονται ιεραρχίες κλάσεων που περιέχουν κλάσεις και υποκλάσεις, ενώ οι σχέσεις μπορούν να οριστούν ως ανεξάρτητες οντότητες, όχι απλώς σαν χαρακτηριστικά (πεδία) των κλάσεων. Η θεμελιώδης διαφορά ανάμεσα σε αυτή την προσέγγιση και αυτή της περιγραφικής λογικής, βρίσκεται στον ορισμό των πεδίων. Στην *OIL* τα πεδία ορίζονται μέσα στις κλάσεις, μια προσέγγιση που χρησιμοποιείται στα συστήματα πλαισίων. Αντίθετα, στις γλώσσες περιγραφικής λογικής τα πεδία ορίζονται εκτός των κλάσεων, σε συνολικό επίπεδο.

Προκειμένου να διευκολυνθεί η προσπάθεια προτυποποίησης της, η *OIL* έχει υιοθετήσει υπάρχουσες προτάσεις σύνταξης και σημασιολογίας, ώστε η χρήση της να είναι εύκολη από ανθρώπους που έχουν εμπειρία με άλλες γλώσσες. Έτσι, η σύνταξη της *OIL* έχει κληρονομήσει στοιχεία από την *OKBC*, την *XOL* και την *RDF* και η σημασιολογία της είναι απλή και καθαρή, με υποστήριξη για λογικούς συνειρμούς. Η γλώσσα είναι ακόμα υπό εξέλιξη.

4.3.4.1 Αναπαράσταση γνώσης

Στην *OIL*, μια οντολογία είναι μια δομή που περιγράφεται από διάφορα συστατικά, οργανωμένα σε τρία επίπεδα: το επίπεδο αντικειμένων, το πρώτο μετα-επίπεδο, που είναι το επίπεδο ορισμού της οντολογίας, και το δεύτερο μετα-επίπεδο που περιέχει πληροφορία για την οντολογία, όπως ο κατασκευαστής της. Για την περιγραφή πληροφορίας που περιέχεται στα τρία αυτά επίπεδα, η *OIL* παρέχει τα ακόλουθα δομικά στοιχεία:

- **Κλάσεις.** Ορίζονται μέσω περιορισμών που πρέπει να ικανοποιούν τα αντικείμενα που ανήκουν στην κλάση. Αυτοί οι περιορισμοί (πολλαπλότητα, τύπος, τιμές) είναι αναγκαίες συνθήκες και όχι απαραίτητα ικανές. Υπάρχουν δύο είδη κλάσεων, οι στοιχειώδεις (primitive) και οι μη στοιχειώδεις (non-primitive, ή defined). Η διαφορά τους είναι αυτή που έχει αναφερθεί παραπάνω για άλλες γλώσσες περιγραφικής λογικής, δηλαδή οι στοιχειώδεις κλάσεις πρέπει να κατηγοριοποιηθούν ρητά, ενώ οι μη στοιχειώδεις μπορούν να κατηγοριοποιηθούν αυτόματα, με εκμετάλλευση των δυνατοτήτων αυτόματης κατηγοριοποίησης που προσφέρουν οι γλώσσες περιγραφικής λογικής.
- **Ταξινομίες.** Μπορούν να δημιουργηθούν με χρήση απλής και πολλαπλής κληρονομικότητας, ιδιότητες που ορίζονται στο εσωτερικό κάθε κλάσης. Ο τελεστής `not subclass of` είναι διαθέσιμος επίσης, και χρησιμοποιείται για να δηλώσει αμοιβαία αποκλειόμενες κατατμήσεις (disjoint partitions).
- **Σχέσεις.** Η OIL υποστηρίζει εγγενώς μόνο διμερείς σχέσεις, χρησιμοποιώντας ορολογία πλαισίων (δηλαδή, τις ονομάζει slots). Παρέχει κάποια προκαθορισμένα χαρακτηριστικά για αυτές, όπως η μεταβατικότητα (transitivity) και η συμμετρικότητα (symmetry), τα οποία μπορούν να χρησιμοποιηθούν για να χαρακτηρίσουν κάποια σχέση. Επίσης, είναι δυνατή η δημιουργία σχέσεων γενίκευσης και ειδίκευσης μεταξύ σχέσεων.
- Για τις πολυμερείς σχέσεις υπάρχουν μόνο οι συνήθεις επιλογές της ανάλυσης τους σε πολλές διμερείς σχέσεις ή της χρήσης συσχετιστικών (associative) κλάσεων, με τα μειονεκτήματα που αυτές επισύρουν.
- **Συναρτήσεις.** Δεν υπάρχει υποστήριξη για συναρτήσεις.

- **Διαδικασίες.** Δεν υποστηρίζονται.
- **Αντικείμενα.** Για τα αντικείμενα έχει ήδη οριστεί στην OIL το επίπεδο αντικειμένων (object level), ωστόσο οι λεπτομέρειες του ορισμού τους δεν έχουν ακόμη διευκρινιστεί.
- **Αξιώματα.** Μέχρι στιγμής η OIL δεν υποστηρίζει αξιώματα, ωστόσο η συμπερίληψη αξιωμάτων στη γλώσσα είναι από τις βασικές ερευνητικές κατευθύνσεις στην ανάπτυξη της γλώσσας.
- **Κανόνες.** Όπως και με τα αξιώματα, αυτό το χαρακτηριστικό δεν υποστηρίζεται από την OIL, αλλά αποτελεί υψηλή προτεραιότητα για την περαιτέρω ανάπτυξη της.

4.3.4.2 Μηχανή λογικής

Η OIL διαθέτει μια μηχανή αυτόματης κατηγοριοποίησης που ονομάζεται FaCT (Fast Classification of Terminologies). Χρησιμοποιεί πλήρεις και ορθούς αλγορίθμους κατηγοριοποίησης, βασισμένους σε λογικές περιγραφές. Οι κύριες λειτουργίες του είναι ο έλεγχος της συνέπειας των ορισμών κλάσεων που περιέχει μια οντολογία, και ο εντοπισμός σχέσεων γενίκευσης – ειδίκευσης μεταξύ των κλάσεων, οι οποίες δεν έχουν ρητά δοθεί αλλά προκύπτουν από τους ορισμούς των κλάσεων.

4.4 ΣΥΓΚΡΙΣΗ – ΣΥΜΠΕΡΑΣΜΑΤΑ

Στη συνέχεια παρουσιάζονται τα αποτελέσματα της σύγκρισης των γλωσσών που εξετάστηκαν, υπό το πρίσμα του πλαισίου αξιολόγησης που αναλύθηκε στην αρχή του κεφαλαίου. Για να διατηρήσουμε όσο το δυνατό αντικειμενικότερες συνθήκες σύγκρισης, χρησιμοποιήθηκε ένα ενιαίο πλαίσιο αξιολόγησης παρά τα διαφορετικά μοντέλα αναπαράστασης γνώσης που χρησιμοποιούν οι γλώσσες που εξετάστηκαν (πλαίσια, λογική περιγραφής, αντικειμενοστραφή). Στις παραγράφους που ακολουθούν, συνοψίζουμε τα χαρακτηριστικά των γλωσσών

που εξετάστηκαν σε πίνακες. Η σημειολογία που ακολουθείται στους πίνακες είναι η εξής:

- N: Το χαρακτηριστικό υποστηρίζεται από τη γλώσσα.
- O: Το χαρακτηριστικό δεν υποστηρίζεται από τη γλώσσα.
- OE: Το χαρακτηριστικό δεν υποστηρίζεται εγγενώς, αλλά είναι δυνατή η ανάπτυξη του από τον προγραμματιστή, μέσω άλλων χαρακτηριστικών της γλώσσας.
- O?: Η γλώσσα δεν απορρίπτει το χαρακτηριστικό, αλλά αυτό δεν έχει οριστεί. Η υποστήριξη του είναι δυνατή σε μελλοντική επέκταση της γλώσσας.
- ?: Δεν είναι γνωστό αν το συγκεκριμένο χαρακτηριστικό υποστηρίζεται ή όχι.

4.4.1 Αναπαράσταση Γνώσης

Στο ακόλουθο σχήμα συνοψίζονται τα βασικά χαρακτηριστικά αναπαράστασης γνώσης των γλωσσών περιγραφής οντολογιών που εξετάστηκαν. Παρατηρούμε ότι κλάσεις, σχέσεις και αντικείμενα μπορούν να οριστούν σχεδόν σε όλες τις γλώσσες. Στην OKBC και την FLogic, οι οποίες είναι βασισμένες σε μοντέλο πλαισίων, οι σχέσεις μπορούν να αναπαρασταθούν σαν πλαίσια αλλά δεν υποστηρίζονται εγγενώς από τις γλώσσες. Στο OKBC, τα αξιώματα υποστηρίζονται μόνο στο τμήμα ερωταποκρίσεων της προγραμματιστικής διεπαφής (API), όμως η αποθήκευσή τους δεν είναι εγγυημένη για κάθε υλοποίηση του OKBC.

	Ontolingua	OKBC	OCML	LOOM	FLogic	CARIN	XOL	SHOE	RDF(S)	OIL
Κλάσεις	N	N	N	N	N	N	N	N	N	N
Σχέσεις	N	OE	N	N	OE	N	OE	N	N	N
Συναρτήσεις	N	OE	N	N	OE	O	O	O	O	N
Διαδικασίες	N	N	N	N	O	O	O	O	O	O
Αντικείμενα	N	N	N	N	N	N	N	N	N	O
Αξιώματα	N	OE	N	N	N	N	O	O	O	N
Κανόνες	O	O	N	N	O	N	O	O	O	OE

Σχήμα 7: Σύνοψη των χαρακτηριστικών μοντελοποίησης γνώσης

Συναρτήσεις, διαδικασίες και αξιώματα δεν μπορούν να οριστούν με γλώσσες βασισμένες στην XML, με την εξαίρεση κάποιων περιορισμένων μορφών αξιωμάτων, όπως για παράδειγμα συμπερασματικοί κανόνες, που υποστηρίζονται από τη SHOE. Τέλος, αξίζει να αναφερθεί ότι εκτελέσιμες διαδικασίες υποστηρίζονται μόνο από γλώσσες που βασίζονται στη Lisp, ενώ παραγωγικοί κανόνες βρίσκονται μόνο στην OCML, την LOOM και την CARIN.

4.4.1.1 Κλάσεις

	Ontolingua	OKBC	OCML	LOOM	FLogic	CARIN	XOL	SHOE	RDF(S)	OIL
Μετα-κλάσεις	N	N	N	N	N	O	N	O	N	O
Πεδία										
Πεδία αντικειμένου	N	N	N	N	N	N	N	N	N	N
Πεδία κλάσης	N	N	N	N	N	O	N	O	N	OE
Πολυμορφισμός	N	N	N	N	N	N	O	O	O	N
Ιδιότητες πεδίων										
Αρχική τιμή	O	N	N	N	N	O	N	O	O	O
Περιορισμός τύπου	N	N	N	N	N	N	N	N	N	N
Περιορισμός πολλαπλότητας τιμών	N	N	N	N	OE	N	N	O	O	N
Περιγραφή πεδίου	N	N	N	N	O	O	N	N	O	N
Λειτουργική γνώση	O	O	N	N	O	O	O	O	O	O
Ορισμός νέων ιδιοτήτων	N	N	O	N	O	O	O	O	O	O

Σχήμα 8: Υποστήριξη χαρακτηριστικών κλάσεων

Στο σχήμα συνοψίζονται οι δυνατότητες των γλωσσών που εξετάστηκαν ως προς τον ορισμό κλάσεων. Με την εξαίρεση της δυνατότητας υποστήριξης μετα-κλάσεων, τα υπόλοιπα χαρακτηριστικά διακρίνονται σε δύο κατηγορίες: χαρακτηριστικά που αφορούν τον ορισμό των πεδίων, και χαρακτηριστικά που αφορούν τις ιδιότητες των πεδίων (facets).

Η δυνατότητα ορισμού μετα-κλάσεων, παρά τη σημαντικότητα της στην δόμηση γνώσης, δεν υποστηρίζεται από όλες τις γλώσσες. Η μη υποστήριξη αυτής της δυνατότητας είναι ένας σημαντικός περιορισμός στην εκφραστικότητα μιας γλώσσας. Η υποστήριξη πεδίων κλάσης και πεδίων αντικειμένου αναφέρεται στη δυνατότητα ορισμού πεδίων των οποίων η τιμή είναι διαφορετική για κάθε αντικείμενο (πεδία αντικειμένου) ή είναι ίδια για όλα τα αντικείμενα της κλάσης (πεδία κλάσης) και επομένως αντιπροσωπεύουν μια

ιδιότητα της ίδιας της κλάσης κι όχι ενός αντικειμένου της. Τα πεδία αντικειμένου είναι κοινό χαρακτηριστικό όλων των υπό εξέταση γλωσσών, αλλά και τα πεδία κλάσης (αντίστοιχα των static μεταβλητών στις αντικειμενοστραφείς γλώσσες) υποστηρίζονται από την πλειοψηφία.

Από τα υπόλοιπα χαρακτηριστικά του πίνακα, μόνο οι περιορισμοί τύπου είναι κοινό χαρακτηριστικό όλων των γλωσσών, ενώ στην υποστήριξη των υπόλοιπων δυνατοτήτων οι γλώσσες διαφοροποιούνται σημαντικά. Ωστόσο, μπορεί να παρατηρήσει κανείς ότι γενικά οι κλασσικές γλώσσες ανάπτυξης οντολογιών προσφέρουν μεγαλύτερη εκφραστικότητα από αυτές που είναι βασισμένες σε XML και προορίζονται για χρήση στον παγκόσμιο ιστό. Το τυπικότερο τέτοιο παράδειγμα είναι η υποστήριξη περιορισμών πολλαπλότητας τιμών, που υφίσταται σε όλες τις κλασσικές γλώσσες (αν και η FLogic δίνει τη δυνατότητα καθορισμού της μέγιστης πολλαπλότητας, ενώ η ελάχιστη είναι πάντα μηδέν), λείπει όμως στις δύο από τις τέσσερις XML γλώσσες. Πρέπει εδώ να αποσαφηνίσουμε ότι η μικρότερη εκφραστικότητα δεν οφείλεται σε κάποιον εγγενή περιορισμό των XML γλωσσών, αλλά μάλλον θα πρέπει να αποδοθεί στην ανωριμότητα τους.

Τέλος, θα πρέπει να σημειώσουμε ότι η OCML και η LOOM προσφέρουν το ξεχωριστό χαρακτηριστικό της δυνατότητας ενσωμάτωσης λειτουργικής γνώσης στον ορισμό πεδίων. Με τον όρο αυτό εννοούμε πληροφορία η οποία χρησιμοποιείται κατά το χρόνο λειτουργίας του οντολογικού συστήματος για την πραγματοποίηση συνειρμών και την δυναμική τροποποίηση της βάσης γνώσης. Αυτή η δυνατότητα είναι εναρμονισμένη με τον εμφατικά λειτουργικό χαρακτήρα των δύο αυτών γλωσσών. Έτσι, η OCML παρέχει τελεστές όπως :prove-by και :lisp-fun, οι οποίοι παραπέμπουν σε εκτελέσιμες λειτουργίες που σχετίζονται με ένα πεδίο και εκτελούνται για να υπολογιστεί η τιμή του, ενώ η LOOM προσφέρει αντίστοιχη δυνατότητα με τους τελεστές :sufficient, :is-primitive και :implies.

4.4.1.2 Ταξινομίες

	Ontolingua	OKBC	OCML	LOOM	Flogic	CARIN	XOL	SHOE	RDF(S)	OIL
Υποκλάσεις	N	N	N	N	N	N	N	N	N	N
Κατατμήσεις	N	O	O E	N	O E	N	O	O	O	O
-- Πλήρεις	N	O	O E	N	O E	O	O	O	O	O
-- Αμοιβαία αποκλειόμενες	N	O	O E	N	O E	N	O	O	O	O E
Not subclass of	O E	O	O	O E	O	O E	O	O	O	N

Σχήμα 9: Δυνατότητες ορισμού ταξινομιών

Όλες οι γλώσσες προσφέρουν τη βασική υποστήριξη ταξινομιών μέσω του τελεστή subclass-of ή κάποιου ισοδύναμου. Οι Ontolingua, LOOM, CARIN και OIL προσφέρουν επιπλέον τελεστές. Τέτοιοι μπορούν να οριστούν και στις υπόλοιπες γλώσσες σαν σχέσεις, αλλά έτσι χάνεται η δυνατότητα ειδικού χειρισμού τους που δίνει η εγγενής υποστήριξη τους. Στην FLogic πρέπει να οριστούν αξιώματα για να εκφράσουν τη σημασιολογία τους. Η OIL επιτρέπει τη χρήση του τελεστή not subclass of και, μέσω αυτού, τον ορισμό αμοιβαία αποκλειόμενων κατατμήσεων. Και εδώ, οι κλασσικές γλώσσες υπερέχουν σε δυνατότητες.

4.4.1.3 Relations and functions

	Ontolingua	OKBC	OCML	LOOM	FLogic	CARIN	XOL	SHOE	RDF(S)	OIL
Συναρτήσεις	N	N	O	N	N	O	O	O	O	N
Κλάσεις ως μονομερείς σχέσεις	N	N	N	N	O	N	O	N	O	N
Πεδία ως διμερείς σχέσεις	N	N	N	N	O	N	O	N	N	N
Πολυμερείς σχέσεις – συναρτήσεις	N	O E	N	N	O E	N	O	N	N	O E
Περιορισμοί τύπου	N	N	N	N	N	N	O	N	N	N
Περιορισμοί αρτιότητας	N	N	N	N	N	N	O	O	O	O
Λειτουργικοί ορισμοί	O	O	N	N	N	O	O	O	O	O

Σχήμα 10: Υποστήριξη σχέσεων και συναρτήσεων

Οι σχέσεις είναι θεμελιώδες στοιχείο της θεωρητικής οντολογίας και, επακόλουθα, υποστηρίζονται εγγενώς από τις περισσότερες γλώσσες. Έτσι, μόνο η XOL δεν εμπεριέχει καθόλου την έννοια της σχέσης, με αποτέλεσμα ακόμα και διμερείς σχέσεις να μπορούν να οριστούν έμμεσα, υπό τη μορφή πεδίων. Ωστόσο δεν υποστηρίζονται όλα τα επιθυμητά χαρακτηριστικά των σχέσεων από όλες τις γλώσσες, ενώ πολλές δεν διαθέτουν υποστήριξη για συναρτήσεις. Κάποιες γλώσσες αναπαριστούν τις κλάσεις με μονομερείς σχέσεις, έτσι ώστε μπορούν να χρησιμοποιηθούν στην οντολογία με ομοιόμορφο τρόπο, σε σχέση με όλες τις άλλες σχέσεις. Στις υπόλοιπες οι κλάσεις και οι σχέσεις είναι ανομοιογενή στοιχεία που δεν μπορεί να τα διαχειριστεί κανείς με ομοιόμορφο τρόπο. Τα πεδία θεωρούνται συνήθως διμερείς σχέσεις. Στην FLogic αξίζει να σημειωθεί ότι αντιμετωπίζονται ως τριμερείς σχέσεις. Επίσης, πολυμερείς σχέσεις δεν υποστηρίζονται από όλες τις γλώσσες. Φυσικά υπάρχει πάντα η λύση του

έμμεσου ορισμού τους με συσχετιστικές κλάσεις, ή με μια σειρά διμερών σχέσεων.

Η ανάλυση του ρόλου των συναρτήσεων αποκαλύπτει μεγάλες σημασιολογικές διαφορές μεταξύ των γλωσσών. Κάποιες γλώσσες, όπως η KIF και η Ontolingua, αντιμετωπίζουν τις συναρτήσεις ως ειδική περίπτωση πολυμερών σχέσεων, όπου το τελευταίο στοιχείο της σχέσης είναι μοναδικό, όταν όλα τα προηγούμενα είναι δεδομένα. Στην LOOM οι συναρτήσεις αντιμετωπίζονται ως σχέσεις, όπου το αποτέλεσμα μπορεί να υπολογιστεί όταν οι παράμετροι είναι δεδομένες. Η FLogic προχωράει ένα βήμα ακόμα: οι συναρτήσεις ορίζονται ως μέθοδοι στο εσωτερικό μιας κλάσης, και η τιμή τους υπολογίζεται από ένα κανόνα που δίνεται ως τμήμα της μεθόδου. Στην OCML, οι συναρτήσεις είναι διακριτό δομικό στοιχείο που επιτελεί διαφορετικό ρόλο από αυτόν των σχέσεων.

Ο περιορισμός του τύπου των στοιχείων που μετέχουν σε μια σχέση είναι δυνατός σε όλες τις γλώσσες. Η κύρια διαφορά μεταξύ των κλασικών γλωσσών και των XML γλωσσών εστιάζεται στους περιορισμούς αρτιότητας. Έτσι, όλες οι κλασικές γλώσσες επιτρέπουν τέτοιους περιορισμούς, ενώ από τις XML γλώσσες δεν τους υποστηρίζει καμία.

Το τελευταίο χαρακτηριστικό αφορά τη λειτουργικότητα, και όχι τη δυνατότητα μοντελοποίησης των γλωσσών. Η OCML, η LOOM και η FLogic παρέχουν τη δυνατότητα περιγραφής της λογικής υπολογισμού της τιμής μιας σχέσης, αν και υπάρχει διαφορά μεταξύ τους. Η LOOM παρέχει λειτουργικές περιγραφές που χρησιμοποιούνται μόνο για υπολογιστικούς σκοπούς, ενώ η OCML παρέχει και μη λειτουργικές περιγραφές που μπορούν να χρησιμοποιηθούν για αναπαράσταση και μοντελοποίηση. Στην FLogic, οι λειτουργικές περιγραφές ορίζονται ξεχωριστά με αξιώματα. Τέλος, η Ontolingua δεν επιτρέπει τη χρήση εκφράσεων Lisp για τον ορισμό των τιμών σχέσεων, ωστόσο περιέχει κάποιες σχέσεις που έχουν λειτουργικό χαρακτήρα, ο οποίος

ενεργοποιείται μέσω της διεπαφής ερωταπαντήσεων (tell&ask interface). Για παράδειγμα η ερώτηση (N 3 2 ?x) επιστρέφει την τιμή 5 για το x.

4.4.1.4 Αντικείμενα

	Ontolingua	OKBC	OCML	LOOM	FLogic	CARIN	XOL	SHOE	RDF(S)	OIL
Class instances	N	N	N	N	N	N	N	N	N	O
Facts	N	N	N	N	N	N	N	N	N	O
Claims	O	O	O	O	O	O	O	N	N	O

Table 1. Definition of instances.

Η αναπαράσταση αντικειμένων είναι δυνατή σε όλες τις γλώσσες εκτός της OIL. Τα αντικείμενα διακρίνονται σε instances κλάσεων και σχέσεων (facts). Και οι δύο κατηγορίες μπορούν να αναπαρασταθούν σε όλες τις γλώσσες. Ωστόσο, οι ισχυρισμοί (claims) υποστηρίζονται μόνο από δύο από τις XML γλώσσες. Αυτό είναι το μοναδικό σημείο που οι XML γλώσσες υπερτερούν των κλασικών σε εκφραστικότητα, και πηγάζει από το γεγονός ότι αυτές είναι κατασκευασμένες για ανταλλαγή και κοινή χρήση οντολογιών σε μεγάλη κλίμακα.

4.4.1.5 Αξιώματα

	Ontolingua	OKBC	OCML	LOOM	FLogic	CARIN	XOL	SHOE	RDF(S)	OIL
Λογική 1^{ης} τάξεως	N	OE	N	N	N	N	O	OE	OE	O E
Λογική 2^{ης} τάξεως	N	OE	O	O	O	O	O	O	O	O
Ανεξάρτητα αξιώματα (named axioms)	N	N	N	O	O	O	O	O	O	O

Σχήμα 11: Χαρακτηριστικά αξιωμάτων

Τα αξιώματα αποτελούν από μόνα τους ένα καλό μέτρο της εκφραστικότητας μιας γλώσσας περιγραφής οντολογιών. Όσο πιο πολύπλοκα αξιώματα μπορεί να χειριστεί μια γλώσσα, τόσο πιο εκφραστική είναι. Αυτό συμβαίνει με την Ontolingua, η οποία μπορεί να χειριστεί ανεξάρτητα αξιώματα πρώτης και δεύτερης τάξης. Η OCML και η FLogic επίσης υποστηρίζουν ανεξάρτητα αξιώματα, αλλά μόνο πρώτου βαθμού.

Η LOOM υποστηρίζει αξιώματα πρώτου βαθμού στα πλαίσια σχέσεων ή κλάσεων, δηλαδή ο ορισμός του αξιώματος εμπεριέχεται στον ορισμό της κλάσης ή της σχέσης. Το ίδιο ισχύει και για την CARIN, η οποία παρέχει επιπλέον τη δυνατότητα ορισμού κανόνων Horn για την εξαγωγή συμπερασμάτων από τη βάση γνώσης. Οι υπόλοιπες γλώσσες, με την εξαίρεση της XOL που δεν υποστηρίζει αξιώματα καθόλου, έχουν περιορισμένες δυνατότητες ορισμού αξιωμάτων. Η OKBC επιτρέπει το υποσύνολο των αξιωμάτων που μπορούν να αναπαρασταθούν με την KIF, τα οποία εισάγονται είτε σε πλαίσιο (frame), είτε χρησιμοποιώντας τη διεπαφή ερωταπαντήσεων (tell&ask interface). Η SHOE επιτρέπει μόνο τον ορισμό συμπερασματικών (deductive) κανόνων. Τέλος, στην OIL η σύνταξη για αξιώματα δεν έχει οριστεί ενώ στην RDF(S) υπάρχουν σε εξέλιξη έρευνες για τον καθορισμό σύνταξης και σημασιολογίας των συνηθέστερων ειδών αξιωμάτων.

4.4.1.6 Παραγωγικοί Κανόνες

	Ontolingua	OKBC	OCML	LOOM	FLogic	CARIN	XOL	SHOE	RDF(S)	OIL
Συζευκτικοί κανόνες (conjunctive)	O	O	N	N	O	O	O	O	O	O?
Διαζευκτικοί κανόνες (disjunctive)	O	O	N	N	O	O	O	O	O	O?
Εκτέλεση διαδικασιών	O	O	OE	N	O	O	O	O	O	O?

Τροποποίηση βάσης γνώσης	Ο	Ο	N	N	Ο	Ο	Ο	Ο	Ο	Ο?
-------------------------------------	---	---	---	---	---	---	---	---	---	----

Σχήμα 12: Υποστήριξη παραγωγικών κανόνων

Οι παραγωγικοί κανόνες συμπεριλαμβάνονται ως στοιχείο μοντελοποίησης οντολογιών στην LOOM, την OCML και την OIL. Επίσης, η SHOE, η LOOM, η OCML και η CARIN δίνουν τη δυνατότητα ορισμού συμπερασματικών κανόνων, για την εξαγωγή νέας πληροφορίας από την γνώση που ήδη περιέχεται στη βάση. Η LOOM διακρίνει τους συμπερασματικούς κανόνες από τους παραγωγικούς. Οι πρώτοι χρησιμοποιούνται για την επαλήθευση ή διάψευση προτάσεων, ενώ οι δεύτεροι χρησιμοποιούνται για την εκκίνηση διαδικασιών όταν ικανοποιηθεί η πρόταση του αριστερού μέρους τους. Αντίστοιχα, η OCML διαχωρίζει μεταξύ ανάστροφων (backward) και ορθών (forward) κανόνων. Οι δύο αυτές γλώσσες επιτρέπουν την επιλογή προς τα εμπρός ή προς τα πίσω διασύνδεσης (forward/backward chaining) κατά την εκτέλεση, καθορίζοντας έτσι το είδος κανόνων που χρησιμοποιείται. Τέλος, στην OIL οι κανόνες είναι μια απλουστευμένη μορφή αξιωμάτων συμπερίληψης γνώσης.

4.4.2 Μηχανή Λογικής

	Ontolingua	OKBC	OCML	LOOM	FLogic	CARIN	XOL	SHOE	RDF(S)	OIL
Ορθή	O	O	N	N	N	N	O	O	O	N
Πλήρης	O	O	O	O	N	N	O	O	O	N
Αυτόματη κατηγοριοποίηση	O	O	O	N	O	N	O	O	O	N
Διαχείριση εξαιρέσεων	O	O	O	O	N	N	O	O	O	O
Μονοτονική κληρονομικότητα	N	N	N	N	N	N	O?	N	O?	N
Μη μονοτονική κληρονομικότητα	O E	N	O E	N	N	O	O?	O	O?	O
Απλή κληρονομικότητα	N	N	N	N	N	N	O?	N	N	N
Πολλαπλή κληρονομικότητα	N	N	N	N	N	N	O?	N	N	N
Εκτέλεση διαδικασιών	N	N	N	N	O	O	O	O	O	O
Έλεγχος περιορισμών	N	N	N	N	N	N	O	O	O	O
Διασύνδεση προς τα εμπρός	O	O	N	N	N	N	O	O?	O	O
Διασύνδεση προς τα πίσω	O	O	N	N	N	N	O	O?	O	O

Σχήμα 13: Δυνατότητες μηχανής λογικής

Σε όλες τις γλώσσες, με τη μοναδική εξαίρεση της OCML, διαχωρίζουν σαφώς την αναπαράσταση γνώσης από τις λογικές διεργασίες. Για παράδειγμα, η Ontolingua είναι η πιο εκφραστική από τις γλώσσες που καλύπτονται στην παρούσα σύγκριση, αλλά δεν υπάρχει διαθέσιμη μηχανή λογικής που να την υποστηρίζει. Η OCML επιτρέπει τον ορισμό κάποιων χαρακτηριστικών που αφορούν λογικές διεργασίες (όπως για παράδειγμα τον τρόπο διασύνδεσης των κανόνων) μέσα σε στοιχεία αναπαράστασης.

Από πλευράς πληρότητας και ορθότητας, μόνο οι FLogic, CARIN και OIL παρέχουν και τα δύο χαρακτηριστικά. Το χαρακτηριστικό αυτό είναι πολύ σημαντικό, αλλά έρχεται σε αντίθεση με την εκφραστική δυνατότητα της γλώσσας, η οποία επιβαρύνει σημαντικά τη δυνατότητα υλοποίησης μιας

μηχανής λογικής. Η αυτόματη κατηγοριοποίηση, ως επιμέρους χαρακτηριστικό της μηχανής λογικής, παρέχεται από τις γλώσσες λογικής περιγραφής (CARIN, LOOM, OIL). Επίσης, η διαχείριση εξαιρέσεων (exception handling) δεν είναι ιδιαίτερα διαδεδομένη, αφού υποστηρίζεται μόνο από την FLogic και την CARIN.

Η απλή αλλά και η πολλαπλή κληρονομικότητα υποστηρίζονται σχεδόν καθολικά (με την εξαίρεση της XOL), ωστόσο δεν υπάρχουν σε καμία γλώσσα μηχανισμοί διαχείρισης συγκρούσεων στην πολλαπλή κληρονομικότητα. Όλες οι γλώσσες είναι μονοτονικές, αν και συνήθως διαθέτουν κάποιες μη μονοτονικές ιδιότητες. Για παράδειγμα, η Ontolingua και οι OCML προσφέρουν κάποιες μη μονοτονικές δυνατότητες στη διαχείριση αρχικών τιμών πεδίων.

Έλεγχο περιορισμών πραγματοποιούν οι κλασσικές γλώσσες. Στην OKBC, ο έλεγχος περιορισμών ορίζεται ως υποχρεωτικός στην προδιαγραφή της γλώσσας, έτσι ώστε όλες οι υλοποιήσεις πρέπει να τον υποστηρίζουν υποχρεωτικά. Ωστόσο, είναι δυνατή η παραμετροποίηση ή και πλήρης απενεργοποίηση του. Στη SHOE δεν επιχειρείται έλεγχος περιορισμών, με το σκεπτικό ότι διαφορές και συγκρούσεις θα είναι συχνές σε διεσπαρμένα περιβάλλοντα παράλληλης ανάπτυξης οντολογιών, και η αυτόματη επίλυση τους προβληματική. Έτσι ο μόνος έλεγχος περιορισμών που πραγματοποιείται αφορά τον τύπο παραμέτρων.

Η αντιμετώπιση των μεθόδων διασύνδεσης (chaining) διαφέρει σημαντικά ανάμεσα στις γλώσσες, με τρόπο που δεν μπορεί να παρασταθεί εύκολα στον πίνακα παραπάνω. Η SHOE δεν ορίζει το μοντέλο διασύνδεσης, έτσι ώστε κάθε υλοποίηση της μπορεί να επιλέξει ανάμεσα στα δύο ελεύθερα. Η OCML υποστηρίζει και τα δύο μοντέλα, και επιτρέπει τον καθορισμό του μοντέλου διασύνδεσης κατά τον ορισμό των κανόνων. Επίσης, η LOOM πραγματοποιεί διασύνδεση και με τους δύο τρόπους, ενώ η FLogic χρησιμοποιεί ένα υβριδικό σχήμα που κάνει χρήση διασύνδεσης τόσο προς τα εμπρός όσο και προς τα πίσω.

5 ΟΛΟΚΛΗΡΩΣΗ ΠΗΓΩΝ ΠΛΗΡΟΦΟΡΙΑΣ ΜΕ ΟΝΤΟΛΟΓΙΕΣ

Στη λογοτεχνία ολοκλήρωσης βάσεων δεδομένων υπάρχουν τρεις βασικές προσεγγίσεις στο πρόβλημα ολοκλήρωσης. Η ονομαζόμενη προσέγγιση του καθολικού σχήματος (global schema approach) [Batini, 1986], [Czejdo, 1987], [Dayal & Hwang, 1984], [Elmasri & Navathe, 1984], [Koh & Chen, 1993] λύνει το πρόβλημα ολοκλήρωσης βάσεων δεδομένων με την αναγωγή σε ένα ενοποιημένο καθολικό σχήμα και ειδικές χαρτογραφήσεις μεταξύ του κεντρικού σχήματος και των επιμέρους σχημάτων. Η προσέγγιση γλωσσών πολλαπλών βάσεων δεδομένων (multi-database languages) [Pitoura, 1997], [Krishnamurthy, 1991], [Litwin, 1990], [Chen, 1993] μεταβιβάζει ένα μεγάλο μέρος της στοιχειώδους εργασίας ολοκλήρωσης στους τελικούς χρήστες, παρέχοντας τους μια εύκαμπτη γλώσσα για πολλαπλές βάσεις δεδομένων, στην οποία όλα τα τοπικά σχήματα είναι μεταφρασμένα. Η βασισμένη σε οντολογίες προσέγγιση [Hammer & McLeod, 1996], [Kahng & McLeod, 1996], [Wiederhold, 1994] [Mena, 2000], [Ullrich, 2001], [Wand & Weber, 1990] χρησιμοποιεί οντολογίες στη θέση του καθολικού σχήματος και ακολούθως μεταφράζει τα επιμέρους σχήματα από και προς αυτές.

Οι οντολογίες αποτελούν ένα ικανό εργαλείο που είναι αποτελεσματικό σε πολλά προβλήματα ολοκλήρωσης βάσεων δεδομένων. Εντούτοις, ο αποφασιστικός παράγοντας για τη χρήση τους στα πλαίσια του συστήματός μας είναι η επιθυμητή υποστήριξη για τη φυσική γλώσσα. Το MKBEEM έχει ως σκοπό να είναι ένα μια πύλη για πολύγλωσσο ηλεκτρονικό εμπόριο, που προσφέρει πρόσβαση στους διάφορους παροχείς περιεχομένου (πληροφοριών ή υπηρεσιών). Ένα τέτοιο σύστημα αντιμετωπίζει τρεις προκλήσεις:

- Να ενσωματώσει τα διάφορα, ενδεχομένως ετερογενή, συστήματα παροχών περιεχομένου.
- Να μεσολαβήσει μεταξύ των συστημάτων των παροχών και του πελάτη.
- Να υποστηρίξει τα ερωτήματα χρηστών σε φυσική γλώσσα.

Αυτή η ανάλυση είναι προσχεδιασμένη έτσι ώστε να διαφοροποιήσει μεταξύ του εμπρός και του πίσω μέρους του συστήματος, δηλ. την περιοχή των πελατών και την περιοχή των προμηθευτών. Αυτή η διαφοροποίηση είναι κατά ένα μεγάλο μέρος τεχνητή. Η μόνη ουσιαστική δικαιολόγηση της είναι η ροή των πληροφοριών, η οποία είναι πάντα από τα συστήματα παροχών περιεχομένου στον πελάτη. Εάν επιλέξουμε να αγνοήσουμε αυτό το χαρακτηριστικό, οι τρεις παραπάνω προκλήσεις συγκλίνουν σε μια: την ολοκλήρωση ανάμεσα σε διάφορα συστήματα πληροφοριών, κάθε ένα από τα οποία έχει την δική του αναπαράσταση δεδομένων και το δικό του μοντέλο γνώσης. Ειδικότερα, κάθε σύστημα παροχών περιεχομένου χρησιμοποιεί τη βάση δεδομένων και το αντίστοιχο σχήμα της, ενώ ο πελάτης χρησιμοποιεί τη φυσική γλώσσα, την γνώση του κόσμου και την κοινή λογική του. Σύμφωνα με αυτή τη θεώρηση, ο πελάτης μπορεί να αντιμετωπιστεί ως ένα ακόμα σύστημα προς ολοκλήρωση, που έχει το ιδιαίτερο του μοντέλο γνώσης και ένα ασυνήθιστο και περίπλοκο σύστημα αναπαράστασης της.

Αυτή η περισσότερο αφηρημένη θεώρηση παρέχει μια διαισθητική αιτιολόγηση της επιλογής των οντολογιών ως μηχανισμό ολοκλήρωσης. Οι τεχνικές καθολικού σχήματος και γλωσσών πολλαπλών βάσεων δεδομένων δεν είναι επαρκείς σε αυτό το πλαίσιο, αφού απευθύνονται συγκεκριμένα σε συστήματα βάσεων δεδομένων. Και οι δύο υποθέτουν ότι τα επιμέρους συστήματα έχουν σχεσιακά (entity-relationship) ή το πολύ αντικειμενοστραφή μοντέλα δεδομένων. Αυτή η υπόθεση σαφώς δεν ισχύει για το σύστημα των πελατών, όπου οι ερωτήσεις εκφράζονται σε ανθρώπινη γλώσσα και το υποκείμενο μοντέλο γνώσης είναι πάρα πολύ σύνθετο για να εκφραστεί με ένα σχεσιακό μοντέλο.

Η προσέγγιση με οντολογίες, αντίθετα, είναι βασισμένη σε έναν φορμαλισμό με πολύ μεγαλύτερη εκφραστική ισχύ. Είναι σε θέση να εκφράσει το σύνθετο μοντέλο γνώσης των ανθρώπων, τουλάχιστον σε περιορισμένες περιοχές (domains). Επιπλέον μπορεί να παρέχει τις υπηρεσίες αιτιολόγησης και εξαγωγής συμπερασμάτων (inference and reasoning) που είναι αναγκαίες για την υποστήριξη των διαδικασιών επεξεργασίας φυσικής γλώσσας. Υπάρχει εκτενής έρευνα για την εφαρμογή των οντολογιών στο πρόβλημα ολοκλήρωσης βάσεων δεδομένων, καθώς επίσης και στην κατανόηση φυσικής γλώσσας [Dahlgren, 1995]. Ως εκ τούτου η οντολογική προσέγγιση είναι ιδανική για ένα σύστημα που περιλαμβάνει και τις δύο αυτές τις περιοχές.

Καταλήγουμε λοιπόν στο συμπέρασμα ότι η οντολογική προσέγγιση στην ολοκλήρωση βάσεων δεδομένων έχει εντελώς διαφορετική δυναμική από τις υπόλοιπες μεθόδους που έχουν κατά καιρούς χρησιμοποιηθεί, ή χρησιμοποιούνται. Η δυναμική ξεπερνά κατά πολύ τα πλεονεκτήματα που η οντολογική προσέγγιση παρουσιάζει στο καθεαυτό πρόβλημα της ολοκλήρωσης δεδομένων. Η βασική συνιστώσα της έγκειται στο γεγονός ότι δίνει νέες δυνατότητες πρόσβασης στα δεδομένα αυτά, οργανώνοντας τα κάτω από μια σημασιολογική δομή πολύ πλουσιότερη από τα συνήθη μοντέλα δεδομένων.

Επομένως αυτή η μέθοδος δίνει νέα διάσταση στην ολοκλήρωση δεδομένων, κάνοντας τα άμεσα διαθέσιμα σε πλέον εξελιγμένες εφαρμογές, οι οποίες επίσης βασίζονται στη χρήση οντολογιών. Οι δύο κύριες τέτοιες εφαρμογές, που αρκούν για να αναδείξουν την οντολογική ολοκλήρωση βάσεων δεδομένων σε εξαιρετικά σημαντική τεχνική, είναι οι διαπροσωπείες φυσικής γλώσσας και, φυσικά, ο Σημασιολογικός Ιστός (Semantic Web) [Hendler, 2001], [McIlraith, 2001], [Decker, 2000].

Σε τέτοιου είδους εφαρμογές, το συγκριτικό πλεονέκτημα των οντολογιών είναι ότι μπορούν να περικλείσουν όλη την απαραίτητη γνώση για τον κόσμο σε μια ενιαία δομή. Χρησιμοποιούνται και για την ολοκλήρωση των διάφορων πηγών περιεχομένου, αλλά και για την επεξεργασία της ανθρώπινης γλώσσας. Το πλεονέκτημα να κωδικοποιούνται όλες αυτές οι πληροφορίες σε μια ενιαία δομή είναι τεράστιο σε ένα τέτοιο σύστημα όπου το κόστος, η δυνατότητα πραγματοποίησης (feasibility) και η συντήρηση των σημασιολογικών δομών είναι συχνά αποφασιστικοί παράγοντες επιτυχίας. Αυτό το χαρακτηριστικό διακρίνει την οντολογική προσέγγιση από άλλες προσεγγίσεις ολοκλήρωσης βάσεων δεδομένων και κάνει τις οντολογίες μια άριστη λύση για οποιοδήποτε σύστημα όπου γλωσσολογική γνώση και meta-data βάσεων δεδομένων ή άλλων πηγών πληροφορίας πρέπει να συνυπάρξουν.

5.1 ΥΠΑΡΧΟΥΣΕΣ ΠΡΟΣΕΓΓΙΣΕΙΣ ΣΤΗΝ

ΟΝΤΟΛΟΓΙΚΗ ΟΛΟΚΛΗΡΩΣΗ

Η χρησιμότητα των οντολογιών ως μέσο ολοκλήρωσης στα συστήματα ανάκτησης πληροφοριών έχει αναγνωρισθεί και έχει αξιοποιηθεί σε διάφορα συστήματα. Αναφέρουμε εδώ μερικά πρόσφατα επιτυχή παραδείγματα, τα οποία βέβαια δεν αποτελούν με κανένα τρόπο έναν πλήρη κατάλογο.

Το Ontobroker [Fensel, 1998], [Decker, 1999] είναι ένα σύστημα ανάκτησης πληροφοριών από το WWW βασισμένο στον εμπλουτισμό των σελίδων HTML με

οντολογική πληροφορία. Το σύστημα προσφέρει μια γραφική διαπροσωπεία δημιουργίας και εκτέλεσης ερωτήσεων (queries) μέσω της οποίας οι χρήστες μπορούν να εισαγάγουν ερωτήσεις που εκφράζονται σε οντολογικούς όρους. Βασίζεται σε frame-logic και περιλαμβάνει μια μηχανή αιτιολόγησης (reasoning engine), που χρησιμοποιείται για την επεξεργασία ερωτήσεων και αποτελεσμάτων. Οι δημιουργοί αναγνωρίζουν το πρόβλημα της εισαγωγής οντολογικής πληροφορίας στις ιστοσελίδες και προσφέρουν ένα απλό σχήμα οντολογικής επισημείωσης σελίδων και ένα υποστηρικτικό πλαίσιο που επιτρέπει την σταδιακή εισαγωγή των οντολογικών σημειώσεων, σε διακριτά βήματα.

Το InfoSleuth [Bayardo, 1997], [Jacobs, 1996] είναι ένα σύστημα βασισμένο σε ευφυείς πράκτορες για την ανάκτηση πληροφοριών από δομημένες ή ημι-δομημένες πηγές. Έχει ένα εκτεταμένο μοντέλο πρακτόρων, στο οποίο οι πράκτορες επικοινωνούν με μηνύματα, με τη βοήθεια μιας εξειδικευμένης οντολογίας υπηρεσιών. Όπως και στο Ontobroker, οι χρήστες εκφράζουν τις ερωτήσεις τους σε οντολογικούς όρους, χρησιμοποιώντας όρους από διάφορες οντολογίες περιοχών. Το στρώμα πρακτόρων του συστήματος έχει υλοποιηθεί κατά το μεγαλύτερο μέρος σε Java, αλλά χρησιμοποιεί ακόμη LISP, CLIPS και LDL ++ (μια γλώσσα βάσεων δεδομένων βασισμένη σε formal logic, με αντικειμενοστρεφείς επεκτάσεις).

Το σύστημα OBSERVER [Mena, 2000] χρησιμοποιεί πολλαπλές οντολογίες για να ανακτήσει πληροφορίες από συστήματα γεωγραφικών πληροφοριών (Geographic Information Systems, GIS), δίνοντας έμφαση στη συνεργασία των πολλαπλών οντολογιών. Έχει μια κατανεμημένη αρχιτεκτονική που αποτελείται από κόμβους συστατικών (component nodes), όπου κάθε συστατικό περιέχει την δική του αποθήκευση στοιχείων, τις οντολογίες και τον επεξεργαστή ερωτήσεων (query processor) του. Η συνεργασία μεταξύ των κόμβων ρυθμίζεται από τους εξυπηρετητές οντολογίας τους, μέσω ενός κεντρικού

συστατικού που χρησιμοποιείται για τη λειτουργικότητα των οντολογιών. Το σύστημα χρησιμοποιεί CLASSIC, μια γλώσσα βασισμένη σε description logic.

Ανάμεσα σε αυτές και αρκετές άλλες διαθέσιμες προσεγγίσεις στην ολοκλήρωση πληροφορίας με οντολογίες, το σύστημα YACOB [Sattler, 2004] πλησιάζει περισσότερο τη λογική του Ozone, καθώς αντιμετωπίζει τα προβλήματα της κλιμάκωσης και επέκτασης του συστήματος, καθώς νέες πηγές δεδομένων προστίθενται σε αυτό. Το YACOB είναι ένα σύστημα για ανάκτηση πληροφορίας από πηγές που μπορούν να υποστηρίξουν την εξαγωγή των δεδομένων τους σε XML. Οι οντολογίες του συστήματος είναι κατασκευασμένες ειδικά για το σκοπό αυτό και κωδικοποιημένες σε RDF(S). Οι απαραίτητες αντιστοιχίσεις από τα περιφερειακά σχήματα δεδομένων στις οντολογίες του συστήματος κωδικοποιούνται επίσης σε RDF και περιλαμβάνονται στο μεταμοντέλο της γνωστικής περιοχής. Έτσι, δεδομένων των αντιστοιχιών αυτών, οι μετατροπές από τις περιφερειακές απεικονίσεις δεδομένων στην κεντρική, γίνονται αυτόματα από το σύστημα με χρήση XSLT. Το σύστημα υποστηρίζει την πραγματοποίηση ερωτημάτων, δομημένων ή με λέξεις-κλειδιά, με τη χρήση της CQuery, μιας γλώσσας για ερωτήματα XML βασισμένης στην XQuery. Η χρήση της υποδομής XML και η απλότητα της RDF σε σχέση με άλλες γλώσσες κωδικοποίησης οντολογιών, κάνουν το YACOB μια προσέγγιση πολύ πιο πρακτική και εύκολη στην χρήση από τις άλλες προσεγγίσεις της βιβλιογραφίας.

Συγκρινόμενο με το YACOB, το Ozone δεν επιτρέπει αναζητήσεις με λέξεις-κλειδιά και δεν αναλαμβάνει την αυτόματη μετατροπή των δεδομένων από την δομή και αναπαράσταση της πηγής σε αυτές του Ozone. Ο μετασχηματισμός αυτός, όπως και η υποχρέωση της υλοποίησης του interface του Ozone, βαρύνει τις πηγές. Έτσι δεν υπάρχει η ανάγκη να τεθούν συγκεκριμένοι περιορισμοί για αυτές, όπως στην περίπτωση του YACOB. Οι πηγές που υποστηρίζουν SQL λαμβάνουν το πλεονέκτημα της διαθεσιμότητας εργαλείων που υποστηρίζουν την κατασκευή της αντιστοίχισης και του ίδιου του interface, αλλά άλλοι τύποι

πηγών μπορούν επίσης να χρησιμοποιηθούν καταβάλλοντας κάποια επιπλέον προσπάθεια για την ολοκλήρωση τους με το Ozone. Σε αντίθεση, το YACOB θέτει κάποιους περιορισμούς για τις πηγές προκειμένου να καταστεί εφικτή η αυτόματη μετατροπή που περιγράφηκε παραπάνω. Συγκεκριμένα, οι πηγές πρέπει να μπορούν να μεταδώσουν τα δεδομένα τους σε XML μορφή, και επίσης πρέπει να μπορούν να διαχειριστούν απλές προτάσεις XPath. Η μετατροπή από XPath στην εγγενή αναπαράσταση της πηγής βαρύνει την πηγή. Ουσιαστικά λοιπόν και το YACOB προϋποθέτει την ύπαρξη ενός μεσολαβητή ανάμεσα σε αυτό και την πηγή που διαχειρίζεται το μετασχηματισμό ερωτημάτων και αποτελεσμάτων προς τις δύο κατευθύνσεις, σε ευθεία αντιστοιχία με το interface του Ozone προς τις πηγές.

Τέλος, το YACOB αντιμετωπίζει με πολύ διαφορετικό από το Ozone τρόπο το θεμελιώδες πρόβλημα της οντολογικής ολοκλήρωσης, αυτό της ανάγκης εξοικείωσης με οντολογίες από πλευράς των προγραμματιστών εφαρμογών και πηγών δεδομένων: οι όροι που χρησιμοποιούνται στα ερωτήματα, αν και δομούνται με χρήση XML, είναι οντολογικοί όροι που αναφέρονται στις οντολογίες του συστήματος. Ο προγραμματιστής της εφαρμογής χρειάζεται έτσι καλή κατανόηση αυτών των οντολογιών για να χρησιμοποιήσει το σύστημα. Όμως η δόμηση των οντολογιών σε RDF επιτρέπει την εύκολη κατανόηση τους ακόμα και χωρίς οντολογικές γνώσεις. Φυσικά η RDF δεν είναι πανάκεια, η ευχρηστία της αντισταθμίζεται από την έλλειψη εκφραστικού πλούτου και την αδυναμία υποστήριξης λογικών συνειρμών (βλέπε παρ. 4.3.1). Επομένως η προσέγγιση του YACOB το δεσμεύει στη χρήση μόνο σε μία οντολογική γλώσσα και το αποστερεί από τις εκφραστικές και λογικές δυνατότητες πιο περίπλοκων γλωσσών. Το Ozone, εν αντιθέσει, είναι εντελώς ανεξάρτητο από τις οντολογίες και μπορεί να χρησιμοποιήσει ταυτόχρονα πολλές διαφορετικές οντολογικές πλατφόρμες, ακόμα και τις πλέον περίπλοκες, με το κόστος κάποιας επιπλέον πολυπλοκότητας και κόπου στη διασύνδεση. Έτσι όσο αφορά την αναπαράσταση

δεδομένων και γνώσης, το Ozone βρίσκεται ανάμεσα στις κλασικές προσεγγίσεις και το YACOB στον άξονα αύξουσας απλότητας – φθίνοντα πλούτου χαρακτηριστικών. Αντίθετα, όσο αφορά την αναπαράσταση ερωτημάτων το Ozone βρίσκεται δεξιά του YACOB στον ίδιο άξονα, δεδομένου ότι χρησιμοποιεί τις ίδιες δομές για την αναπαράσταση δεδομένων και ερωτημάτων, ενώ το YACOB ορίζει μια νέα γλώσσα ερωτημάτων για το σκοπό αυτό.

5.2 ΠΡΟΒΛΗΜΑΤΑ

Αν και οι βασισμένες σε οντολογίες τεχνικές ολοκλήρωσης είναι σε ώριμο στάδιο, τα υπάρχοντα συστήματα ολοκλήρωσης αποτυγχάνουν να καταδείξουν αντίστοιχο βαθμό ωριμότητας. Τα περισσότερα είναι πρωτότυπα ή καθαρά ερευνητικά συστήματα, όπου η έμφαση είναι στις οντολογίες και τις μηχανές αιτιολόγησης. Κατά συνέπεια, αναπτύσσονται στο μέγιστο μέρος σε γλώσσες λογικού προγραμματισμού και κάνουν χρήση σύνθετων εννοιών, εργαλείων και μοντέλων αναπαράστασης γνώσης.

Οι γλώσσες λογικού προγραμματισμού είναι ένα πολύ χρήσιμο ακαδημαϊκό και ερευνητικό εργαλείο. Ωστόσο, οι ίδιες οι γλώσσες αλλά και τα εργαλεία ανάπτυξης που τις συνοδεύουν δεν έχουν σχεδιαστεί για μεγάλης κλίμακας βιομηχανική χρήση. Ακόμα χειρότερα, η ανάπτυξη λογισμικού με τέτοιες γλώσσες δεν έχει τύχει συστηματικής μελέτης από την πλευρά της διοίκησης ολικής ποιότητας και της διαχείρισης έργων λογισμικού. Κατά συνέπεια, δεν υπάρχουν καθορισμένες μεθοδολογίες και διαδικασίες για την ομαδική ανάπτυξη, συντήρηση, έλεγχο και διόρθωση, και επαναχρησιμοποίηση λογισμικού με τέτοια εργαλεία, στα πλαίσια του συνολικού κύκλου ζωής ενός έργου πληροφορικής.

Τα προβλήματα που δημιουργεί η εξάρτηση από γλώσσες λογικού προγραμματισμού των οντολογικών συστημάτων σε επίπεδο μηχανικής λογισμικού μπορούν να συνοψιστούν στις παρακάτω κατηγορίες:

- Προβλήματα αρχιτεκτονικής
- Προβλήματα ολοκλήρωσης με άλλα υποσυστήματα, ή με λογισμικό που έχει αναπτυχθεί σε άλλες γλώσσες
- Προβλήματα αναπαράστασης γνώσης και δεδομένων
- Προβλήματα συντήρησης, επεκτασιμότητας και διαχείρισης γενικότερα.

Τα προβλήματα αυτά αναλύονται στις επόμενες παραγράφους.

5.2.1 *Ανελαστική Αρχιτεκτονική*

Τα υπάρχοντα οντολογικά συστήματα ολοκλήρωσης χαρακτηρίζονται από ασαφή ή και εντελώς ανύπαρκτη δομή. Αυτό οφείλεται σε δύο, κυρίως λόγους. Ο πρώτος είναι ότι αυτά τα συστήματα συνήθως αναπτύσσονται για λόγους καθαρά ερευνητικούς, κατά συνέπεια η έμφαση είναι στην επιτυχή λειτουργία του συστήματος για λόγους επίδειξης. Αντίθετα, παράγοντες όπως οι επιδόσεις, η επεκτασιμότητα και η συντήρηση δεν απασχολούν καθόλου αφού το σύστημα δεν τίθεται ποτέ σε πλήρη επιχειρησιακή λειτουργία και δεν ακολουθεί τον κύκλο ζωής ενός πραγματικού προϊόντος λογισμικού.

Εκτός από την απλή έλλειψη ενδιαφέροντος, υπάρχει και ένας δεύτερος, πολύ βασικότερος λόγος για τα προβλήματα αρχιτεκτονικής και δομής τέτοιων συστημάτων. Αυτός είναι ότι τα ίδια τα εργαλεία και οι μεθοδολογίες που χρησιμοποιούνται για την ανάπτυξη τους είναι κατασκευασμένα, σε μικρότερο ή μεγαλύτερο βαθμό, με βάση τις ίδιες προτεραιότητες που αναφέρθηκαν παραπάνω για τα οντολογικά συστήματα καθαυτά: πρόκειται, δηλαδή, για εργαλεία ακαδημαϊκής και ερευνητικής χρήσεως, που στην συντριπτική πλειοψηφία τους δεν αποτελούν εμπορικά προϊόντα, στα οποία παράγοντες όπως οι επιδόσεις, η επεκτασιμότητα και η συντήρηση δεν έχουν ιδιαίτερο βάρος στην τελική αξιολόγηση του εργαλείου. Το αποτέλεσμα είναι ότι τα συστήματα

που προκύπτουν από τέτοια εργαλεία έχουν εγγενείς αδυναμίες όσο αφορά την διακριτή εσωτερική τους δόμηση.

5.2.2 Επικοινωνία με Άλλα Υποσυστήματα

Αντίστοιχα ισχύουν στην επικοινωνία με άλλα υποσυστήματα, ιδίως όταν τα δεύτερα είναι κατασκευασμένα χρησιμοποιώντας διαφορετικές γλώσσες ή εργαλεία προγραμματισμού. Η επικοινωνία πρέπει πρώτα να αντιμετωπίσει το σαφή καθορισμό προγραμματιστικών διεπιφανειών, ο οποίος σε οντολογικά συστήματα είναι δύσκολος λόγω της χαλαρής αρχιτεκτονικής τους, που καθιστά δύσκολο να ξεχωρίσει κανείς πού τελειώνει η διεπιφάνεια ενός συστήματος με τον έξω κόσμο, και πού αρχίζει η εσωτερική δομή του. Με άλλα λόγια, προκειμένου ένα οντολογικό σύστημα να γίνει προσβάσιμο είναι πολύ συνηθισμένο ένα πολύ μεγάλο μέρος του, ή και ολόκληρο το σύστημα, να πρέπει να εκτεθεί προς τα έξω και να αποκαλυφθεί μεγάλο μέρος της εσωτερικής δομής του.

Κάτι τέτοιο φυσικά είναι σε πλήρη αντίθεση με την καθολικά αποδεκτή μοντέρνα αντίληψη της μηχανικής λογισμικού, η οποία υποστηρίζει ότι η επικοινωνία μεταξύ συστημάτων πρέπει να γίνεται εκθέτοντας προς τα έξω ένα πολύ μικρό μέρος του συστήματος, έτσι ώστε μεταγενέστερες αλλαγές στο εσωτερικό του να μην επηρεάζουν τα συστήματα που επικοινωνούν μαζί του. Τέτοιες αλλαγές είναι συχνές και πολυπληθείς στη διάρκεια ζωής των προϊόντων λογισμικού, σαν συνέπεια προσπαθειών επέκτασης, διόρθωσης λαθών, αναβάθμισης εσωτερικών λειτουργιών, επικοινωνίας με τρίτα συστήματα, προσαρμογής σε νέες επιχειρησιακές απαιτήσεις και κανόνες, και λοιπά. Τα κόστη που οι αλλαγές σε ένα σύστημα επιφέρουν σε άλλα συστήματα που επικοινωνούν μαζί του, είναι ένας ουσιώδης παράγοντας που οδηγεί τις εταιρίες στην ανάσχεση της εξέλιξης των συστημάτων τους.

Φυσικά, ο καθορισμός των διεπιφανειών μεταξύ συστημάτων και οι συνέπειες που αυτός έχει για την συντήρηση, επεκτασιμότητα και ολικό κόστος τους είναι τελικά μόνο η κορυφή του παγόβουνου. Σε ένα επιχειρησιακό περιβάλλον μεγάλης κλίμακας υπάρχουν ακόμα απαιτήσεις που αφορούν τις επιδόσεις, την κατανάλωση και διαχείριση πόρων, καθώς και κλιμάκωσης (escalation) ή καταμερισμού του φόρτου (load balancing) του συστήματος. Τα εργαλεία λογικού προγραμματισμού δεν προσφέρουν καμία εγγενή υποστήριξη για τέτοια χαρακτηριστικά, που σημαίνει ότι, όπου αυτά είναι απαραίτητα, πρέπει να αναπτυχθούν κατά περίπτωση, ανάλογα με τις ιδιαίτερες δυνατότητες και περιορισμούς που προσφέρονται.

Τα θέματα κλιμάκωσης και επιδόσεων, ιδιαίτερα, αποτελούν σημαντικότατο πρόβλημα σε εφαρμογές που χρησιμοποιούν εργαλεία λογικού προγραμματισμού. Αυτό οφείλεται στην έλλειψη βελτιστοποίησης των εργαλείων και την εγγενή απληστία των γλωσσών λογικού προγραμματισμού ως προς την κατανάλωση υπολογιστικών πόρων. Σε συνδυασμό με την σφιχτή και αδόμητη αρχιτεκτονική τέτοιων συστημάτων, που καθιστά πολύ δύσκολη ή και συνήθως αδύνατη την κατανομή (distribution) του συστήματος σε περισσότερους του ενός υπολογιστές, οδηγούμαστε συχνά σε αδυναμία ικανοποίησης των προδιαγραφών επιδόσεων του συστήματος.

5.2.3 Ετερογένεια Αναπαράστασης

Είναι καλά κατανοητό ότι η ολοκλήρωση στο σημασιολογικό επίπεδο ετερογενών πηγών στοιχείων του συστήματος μπορεί να επιτευχθεί με τη βοήθεια των οντολογιών περιοχών, ειδικότερα με την παροχή αντιστοίχισης μεταξύ των οντολογικών εννοιών και του σχήματος κάθε πηγής δεδομένων. Εντούτοις, η ετερογένεια εμφανίζεται όχι μόνο στο σημασιολογικό επίπεδο αλλά και στο επίπεδο αναπαράστασης δεδομένων. Σε αυτό το επίπεδο, οι οντολογίες δεν βοηθούν καθόλου τη λύση του προβλήματος, αντιθέτως το περιπλέκουν περισσότερο.

Σε ένα σύστημα όπου οι οντολογίες χρησιμοποιούνται για να ολοκληρώσουν ετερογενείς βάσεις δεδομένων προκειμένου να τροφοδοτηθεί μια εμπορική εφαρμογή βασισμένη στον Ιστό, τρεις φορμαλισμοί αναπαράστασης στοιχείων είναι πιθανό να συνυπάρξουν. Αφενός, οι βάσεις δεδομένων χρησιμοποιούν αναπαράσταση δεδομένων στοιχείων βάσει σχεσιακών μοντέλων (entity-relationship, ER), που συνήθως είναι κάποια διάλεκτος της SQL. Αφετέρου, οι οντολογίες είναι πιθανό να κωδικοποιηθούν σε μια γλώσσα αναπαράστασης γνώσης. Τέλος, η μεγάλη πλειοψηφία των σύγχρονων εφαρμογών γράφεται σε αντικειμενοστραφείς γλώσσες και στηρίζεται έτσι σε ένα αντικειμενοστραφές μοντέλο του κόσμου.

Είναι σαφές ότι ο αποδοτικός μετασχηματισμός των στοιχείων και των μετα-δεδομένων (meta-data) μεταξύ αυτών των τριών φορμαλισμών αναπαράστασης είναι τόσο σημαντικός όσο η ίδια η σημασιολογική ολοκλήρωση. Αυτό αποτελεί το πρόβλημα της ολοκλήρωσης αναπαράστασης. Στη διδακτορική μου διατριβή επιδιώκω τον καθορισμό και την περιγραφή αντικειμενοστραφών δομών και μηχανισμών που μπορούν να επιτρέψουν την εύκολη και αποδοτική πρόσβαση σε μια ομοσπονδία βάσεων δεδομένων, η οποία υιοθετεί οντολογίες για τη σημα/σιολογική ολοκλήρωσή της. Η αντικειμενοστραφής γλώσσα είναι μια προφανής υποψήφια για τη θέση της καθιερωμένης γλώσσας αναπαράστασης δεδομένων του συστήματος, αφού χρησιμοποιείται για την ανάπτυξη της εφαρμογής. Άλλες εναλλακτικές λύσεις θα ήταν μια διάλεκτος SQL, η γλώσσα αντιπροσώπευσης γνώσης της οντολογίας, ή ακόμα και ένας άλλος φορμαλισμός διαφορετικός και από τους τρεις παραπάνω.

Κάποιος θα μπορούσε να υποστηρίξει ότι ο βέλτιστος υποψήφιος φορμαλισμός αναπαράστασης θα ήταν πράγματι η γλώσσα αναπαράστασης γνώσης της οντολογίας. Αυτή η θέση υποστηρίζεται από το γεγονός ότι αυτός είναι ο πλουσιότερος των τριών συνυπαρχόντων φορμαλισμών και έχει την ικανότητα να αντικαταστήσει τους άλλους δύο, χωρίς απώλεια πληροφορίας.

Στην πραγματικότητα, έχει ήδη ολοκληρωθεί με επιτυχία αρκετή έρευνα σε αυτήν την κατεύθυνση. Η βασική αδυναμία αυτών των προσεγγίσεων είναι ότι είναι εντελώς ακατάλληλες για εφαρμογή στη βιομηχανία. Είναι βασισμένες σε γλώσσες λογικού προγραμματισμού (logic programming), οι οποίες σήμερα έχουν μηδενική σχεδόν χρήση σε βιομηχανικές εφαρμογές. Το αποτέλεσμα είναι ότι, αν και υπάρχουν μέθοδοι για τη διασύνδεση τέτοιων υποσυστημάτων με αντικειμενοστραφή προγράμματα, ένα σημαντικό μέρος της εφαρμογής πρέπει να αναπτυχθεί σε κάποια γλώσσα λογικού προγραμματισμού.

5.3 ΤΟ OKBC

Το OKBC, που αναλύεται εκτεταμένα στην παράγραφο 4.2.2, αποτελεί την πλέον καταξιωμένη προσπάθεια γεφύρωσης των οντολογιών με τον αντικειμενοστραφή κόσμο. Είναι ένα πρωτόκολλο ανταλλαγής γνώσης μεταξύ συστημάτων τα οποία δεν χρησιμοποιούν απαραίτητα την ίδια γλώσσα για την περιγραφή της. Ως τέτοιο, το OKBC δίνει μια λύση στο πρόβλημα της πρόσβασης σε οντολογίες ανεξάρτητα από την γλώσσα.

Όμως το γεγονός ότι το OKBC δεν είναι κατασκευασμένο για την πρόσβαση σε οντολογίες από εφαρμογές, αλλά για την ανταλλαγή γνώσης, θέτει περιορισμό στην αποτελεσματικότητά του αν χρησιμοποιηθεί για πρόσβαση. Οι δύο κύριοι περιορισμοί αφορούν την αναπαράσταση γνώσης και την πολυπλοκότητα του πρωτοκόλλου.

Το OKBC βασίζεται σε ένα μοντέλο αναπαράστασης γνώσης βασισμένο σε πλαίσια (frames). Η πρόσβαση σε βάσεις γνώσης που δεν χρησιμοποιούν αυτό το μοντέλο είναι δυνατή μόνο με την προσαρμογή της γνώσης στο μοντέλο των πλαισίων. Η επιλογή αυτή περιορίζει τη δυνατότητα πρόσβασης σε οντολογίες που χρησιμοποιούν διαφορετικά μοντέλα, χωρίς παράλληλα να προσφέρει κάποιο πλεονέκτημα στις εφαρμογές που πρέπει να το χρησιμοποιήσουν. Το μοντέλο αναπαράστασης γνώσης, αν και απλό συγκρινόμενο με άλλα

οντολογικά μοντέλα αναπαράστασης, είναι γενικά πολύπλοκο και ξένο προς τα ευρέως διαδεδομένα μοντέλα αναπαράστασης δεδομένων (π.χ. αντικειμενοστραφές, σχεσιακό, ιεραρχικό).

Όσο αφορά το ίδιο το πρωτόκολλο, η κατάσταση είναι αντίστοιχη. Το OKBC προσφέρει μια χαμηλού επιπέδου διαπροσωπεία προς βάσεις γνώσης. Η χρήση της απαιτεί καλή κατανόηση της δομής της βάσης σε αφηρημένο επίπεδο, καλή γνώση της ίδιας της διαπροσωπείας και εξοικείωση με το μοντέλο αναπαράστασης γνώσης. Στην πράξη, η χρήση του OKBC από μια εφαρμογή που αντλεί στοιχεία από οντολογίες θα απαιτήσει την κατασκευή ενός υποσυστήματος διαμεσολάβησης το οποίο θα συνθέτει τις χαμηλού επιπέδου μεθόδους του OKBC σε μεθόδους υψηλότερου επιπέδου, που να έχουν νόημα στα πλαίσια της εφαρμογής, και θα μεταφράζει από το μοντέλο πλαισίων στο μοντέλο γνώσης της εφαρμογής, που συνήθως θα είναι απλούστερο.

Συνοψίζοντας, το OKBC είναι ένα πρωτόκολλο φτιαγμένο για την ανταλλαγή γνώσης που μπορεί να χρησιμοποιηθεί για την πρόσβαση σε οντολογίες, έχοντας όμως σαφείς αδυναμίες χρηστικότητας. Είναι ένα γενικευμένο πρωτόκολλο χαμηλού επιπέδου προσανατολισμένο στην αναλυτική επικοινωνία της γνώσης μεταξύ συστημάτων εξοικειωμένων με τη δόμηση και μοντελοποίηση της. Η χρήση του για πρόσβαση σε γνώση από εφαρμογές παρουσιάζει το μειονέκτημα ενός δύσχρηστου μοντέλου γνώσης και μιας πολύ χαμηλού επιπέδου διαπροσωπείας που θα πρέπει να χρησιμοποιηθεί όχι απευθείας, αλλά μέσω κάποιου συστήματος διαμεσολάβησης (mediation module).

6 OZONE

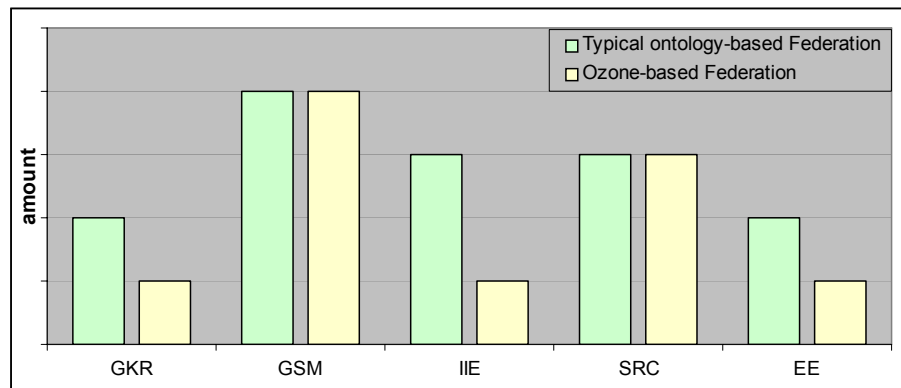
Το φυσικό επόμενο βήμα στην εξέλιξη των οντολογικών συστημάτων ολοκλήρωσης είναι να κινηθούν από το επικεντρωμένο στις οντολογίες μοντέλο του κεφαλαίου 5 προς μια αρθρωτή (component-based) αρχιτεκτονική, όπου το οντολογικό υποσύστημα ολοκλήρωσης αποτελεί ένα back-end τμήμα που υποστηρίζει μια ή περισσότερες εφαρμογές. Το γεγονός ότι οι οντολογίες είναι στενά δεμένες με το δικό τους πρότυπο προγραμματισμού και μοντελοποίησης γνώσης, το οποίο παίρνει τη θέση των συμβατικών μοντέλων δεδομένων, είναι σημαντικό εμπόδιο για τους υπεύθυνους ανάπτυξης της εφαρμογής και τους μηχανικούς λογισμικού. Κατά συνέπεια προκύπτει η ανάγκη για μια αρχιτεκτονική όπου οι οντολογίες μπορούν να συνυπάρξουν με συμβατικές, ευρέως διαδεδομένες αντικειμενοστραφείς πλατφόρμες.

Αυτές οι εκτιμήσεις διαμορφώνουν την κατεύθυνση που ακολουθώ στην εργασία μου. Αναπτύσσω τη μεθοδολογία Ozone [Stavroulas, 2002], [Stavroulas, 2002a] που θα παρέχει βελτιώσεις σε σχέση με τα υπάρχοντα οντολογικά συστήματα πληροφοριών στα ακόλουθα σημεία:

- Απομόνωση των βασισμένων σε λογική περιγραφής (description logics) συστατικών. Αναπτύσσω ένα αρθρωτό (modular) μοντέλο δόμησης συστημάτων που περιορίζει τη χρήση γλωσσών λογικού

προγραμματισμού στο εσωτερικό του οντολογικού υποσυστήματος. Το υπόλοιπο του συστήματος μπορεί να είναι καθαρά αντικειμενοστρεφές.

- Απομόνωση του σύνθετου οντολογικού μοντέλου γνώσης. Προτείνω ένα μινιμαλιστικό μοντέλο γνώσης που συνιστά μια απλοποίηση του οντολογικού μοντέλου γνώσης, με έντονες ομοιότητες με το σχεσιακό (entity-relationship) μοντέλο. Αυτό το μοντέλο είναι πολύ απλούστερο στην κατανόηση και χρησιμοποίηση, ενώ εξακολουθεί να είναι επαρκές για τις ανάγκες τόσο των προγραμματιστών της εφαρμογής, όσο και των παρόχων περιεχομένου (content providers).
- Παροχή εργαλείου για τη διατύπωση ερωτήσεων. Σχεδιάζω μια αντικειμενοστραφή δομή που εφαρμόζει το παραπάνω μοντέλο γνώσης και προσφέρει λειτουργίες για τη διατύπωση και το χειρισμό απλών ερωτήσεων, βασισμένων σε φόρμες. Βοηθά έτσι τους υπεύθυνους για την ανάπτυξη της εφαρμογής να απεικονίσουν γραφικά ερωτήματα μέσω φορμών που ο χρήστης μπορεί εύκολα να συμπληρώσει χωρίς γνώσεις βάσεων δεδομένων, SQL κλπ.
- Σύνδεση βάσεων δεδομένων. Η διασύνδεση με βάσεις δεδομένων είναι όσο το δυνατόν απλούστερη. Επιπλέον, επιχειρώ να τυποποιήσω μια πρότυπη διαδικασία που μπορεί να ακολουθηθεί για να πραγματοποιηθεί η απαραίτητη αντιστοίχιση οντολογίας – σχήματος βάσης, που είναι αναγκαία ώστε να προσχωρήσει μια βάση στην ομοσπονδία.



- GKR:** Global Knowledge Required, amount of global knowledge that each component is supposed to have
- GSM:** Global Structures Maintained, globally, in order to allow information sharing and exchange between components
- IIE:** Initial Integration Effort, the amount of overhead effort required for each component, before it can join the system
- SRC:** Staticity of Relationships between Concepts
- EE:** Exchange Effort, the amount of computer time and user involvement required during the actual sharing and exchange

Σχήμα 14: Ο αντίκτυπος της μεθοδολογίας Ozone στην προσπάθεια και τις γνώσεις που απαιτούνται για να ενταχθεί μια βάση δεδομένων σε μια ομοσπονδία βάσεων

Στο [Aslan, 1999] παρουσιάζεται μια σύγκριση των διάφορων προσεγγίσεων στην ολοκλήρωση βάσεων δεδομένων. Στο [Stavroulas, 2002] απεικονίζεται μια σύγκριση του Ozone σε σχέση με τις τυποποιημένες προσεγγίσεις στην οντολογική ολοκλήρωση μιας ομοσπονδίας βάσεων δεδομένων. Το Ozone πλεονεκτεί σε τρεις πτυχές. Κατ' αρχάς, μειώνει το ποσό γνώσης που κάθε συστατικό του συστήματος πρέπει να διατηρεί για τον κόσμο ή άλλα συστατικά, με την εισαγωγή του απλουστευμένου μοντέλου γνώσης. Δεύτερον, μειώνει ουσιαστικά την αρχική (overhead) προσπάθεια και το αντίστοιχο κόστος που απαιτείται ώστε μια νέα βάση δεδομένων να προσχωρήσει στην ομοσπονδία, καθώς δεν απαιτείται αυτή να αφομοιώσει σύνθετα μοντέλα γνώσης και να χρησιμοποιήσει γλώσσες ή αναπαραστάσεις λογικού προγραμματισμού. Τέλος, μειώνει την προσπάθεια και τους υπολογιστικούς πόρους που απαιτείται για κάθε ανταλλαγή δεδομένων, επειδή δεν αναγκάζει τα υποσυστήματα εφαρμογής και

βάσεων δεδομένων να εκφράσουν και να χειριστούν ερωτήσεις ή αποτελέσματα σε περίπλοκες και αργές στην εκτέλεση οντολογικές εκφράσεις.

6.1 ΑΝΑΠΑΡΑΣΤΑΣΗ ΓΝΩΣΗΣ

Τα οντολογικά μοντέλα αναπαράστασης είναι πολύ πιο πλούσια, τόσο από τα μοντέλα δεδομένων που χρησιμοποιούνται στο σχεδιασμό βάσεων δεδομένων όσο και από τα αντικειμενοστραφή μοντέλα, αφού προορίζονται να καλύψουν ένα πολύ ευρύτερο πεδίο εφαρμογών. Επίσης, διαφέρουν σημαντικά και στις θεμελιώδεις έννοιες (primitives) που χρησιμοποιούν για να περιγράψουν τη γνώση. Τυπικά, η γνώση οργανώνεται σε ταξινομίες εννοιών (concepts), οι οποίες είναι αντίστοιχες με τις ιεραρχίες κλάσεων του αντικειμενοστραφούς σχεδιασμού, αλλά έχουν μεγαλύτερη εκφραστική ισχύ. Το μεγαλύτερο μέρος αυτής της ισχύος είναι άχρηστο στα συστήματα ολοκλήρωσης πληροφορίας, αφού οι πάροχοι της πληροφορίας, συνήθως σχεσιακές ή ιεραρχικές πηγές δεδομένων, δεν έχουν αντίστοιχη υποδομή εκφραστικής δύναμης. Έτσι, στο Ozone επιλέγουμε να θυσιάσουμε την εκφραστική ισχύ ήδη από το επίπεδο διαμεσολάβησης που απομονώνει τις οντολογίες, ώστε αυτή αποκόπτεται πλήρως τόσο από τις εφαρμογές όσο και από τις πηγές δεδομένων – που ούτως ή άλλως δεν θα μπορούσαν να την εκμεταλλευθούν. Αυτή η επιλογή δίνει ως αντάλλαγμα την ευκολία και απλότητα στην πρόσβαση, με τη συνεπαγόμενη οικονομία πόρων και αύξηση επιδόσεων. Στη συνέχεια περιγράφουμε αναλυτικά το μοντέλο αναπαράστασης που χρησιμοποιεί το Ozone, και εξηγούμε την εφαρμογή του στην αναπαράσταση δεδομένων, γνώσης αλλά και ερωτημάτων.

6.1.1 Το Μοντέλο Γνώσης του Ozone

Το Ozone χρησιμοποιεί ένα σχετικά απλό μοντέλο αναπαράστασης, που διαθέτει την *ελάχιστη* δυνατή εκφραστικότητα απαραίτητη για την υποστήριξη της πρόσβασης, εκτέλεσης απλών ερωτημάτων και άντλησης αποτελεσμάτων

από μια ομοσπονδία πηγών δεδομένων. Είναι ένα αντικειμενοστραφές μοντέλο, σχεδιασμένο ώστε να διατηρεί τη μέγιστη δυνατή ομοιότητα με το σχεσιακό μοντέλο.

Στο μοντέλο του Ozone μια οντολογία ορίζεται ως μια ταξινόμια εννοιών (concepts). Κάθε έννοια απαρτίζεται από ένα όνομα, μια λίστα από ιδιότητες (attributes) και μια λίστα από σχέσεις (relations) προς άλλες έννοιες. Οι σχέσεις είναι δυαδικές και πολυμερείς σχέσεις μπορούν να οριστούν μόνο μέσω κλάσεων συσχετισμού (associative classes). Τόσο οι σχέσεις όσο και οι ιδιότητες μπορούν να έχουν πολλαπλές τιμές. Οι σχέσεις διαχωρίζονται από τις ιδιότητες με βάση το είδος των τιμών τους. Οι μεν ιδιότητες λαμβάνουν ατομικές τιμές, δηλαδή αδόμητες, οι δε σχέσεις λαμβάνουν τιμές που αποτελούν έννοιες (concept instances) και έχουν δομή. Έτσι, η διάκριση μεταξύ σχέσεων και ιδιοτήτων είναι ίσως τεχνητή, από την οντολογική σκοπιά, ωστόσο επιλέχθηκε καθότι η έννοια των ατομικών ιδιοτήτων είναι σε πλήρη αντιστοιχία με το σχεσιακό μοντέλο. Επιβάλλοντας αυτή την τεχνητή διάκριση απεξαρτούμε τις έννοιες της σχέσης και της ιδιότητας από την ερμηνεία του εκάστοτε οντολογικού μοντέλου, και δίνουμε ένα μονοσήμαντο και γενικό ορισμό που είναι καλά κατανοητός στον αντικειμενοστραφή και το σχεσιακό κόσμο.

Μια πλήρης εικόνα των δυνατοτήτων αναπαράστασης του μοντέλου του Ozone δίνεται στο Σχήμα 15, όπου γίνεται μια σύγκριση του μοντέλου με αυτό της Ontolingua, που είναι ένα από τα πλέον ισχυρά σε έκφραση οντολογικά μοντέλα.

	Ozone	Ontolingua
General		
Concept taxonomies	Yes	Yes
Inheritance	Single	Multiple
Concept instances	Yes	Yes
Relation instances (facts)	No	Yes
Axioms	No	Yes (2 nd order logic)

Attributes		
Instance attributes	Yes	Yes
Concept attributes	No	Yes
Default values	No	Yes
Type constraints	Yes	Yes
Multi-valued	Yes	Yes
Cardinality constraints	Yes	Yes
Relations		
Multiplicity	Binary	n-ary
Type constraints	Yes	Yes
Integrity constraints	No	Yes
Functions	No	Yes

Σχήμα 15: Σύγκριση δυνατοτήτων αναπαράστασης μεταξύ Ozone και Ontolingua

Όπως προκύπτει από το σχήμα, οι βασικοί περιορισμοί του Ozone είναι οι εξής:

1. Δεν υπάρχει υποστήριξη για αξιώματα
2. Δεν επιτρέπεται η πολλαπλή κληρονομικότητα
3. Δεν υποστηρίζονται αρχικές (default) τιμές
4. Άμεση υποστήριξη μόνο για διμερείς σχέσεις, έμμεση υποστήριξη για πολυμερείς μέσω κλάσεων συσχετισμού (associative classes)
5. Δεν υποστηρίζονται συναρτήσεις (δηλαδή, ν-μερείς σχέσεις όπου το ν-οστό όρισμα ορίζεται μονοσήμαντα όταν τα πρώτα ν-1 είναι γνωστά)

Το μοντέλο του Ozone είναι πολύ λιγότερο εκφραστικό από το αντίστοιχο της Ontolingua, αλλά εξακολουθεί να είναι περισσότερο εκφραστικό από τα σχεσιακά μοντέλα. Κατά συνέπεια, σε μια ομοσπονδία που βασίζεται σε σχεσιακές βάσεις δεν υπάρχει απώλεια της συνολικής εκφραστικότητας του μοντέλου του συστήματος, γίνεται όμως ένας επανακαθορισμός των ορίων μεταξύ των πλέον εκφραστικών τμημάτων (οντολογίες) και των λιγότερο εκφραστικών (εφαρμογή και πηγές δεδομένων), με το Ozone να βρίσκεται στο

μέσο τόσο από πλευράς αρχιτεκτονικής συστήματος όσο και από πλευράς εκφραστικότητας. Η υιοθέτηση αυτού του μοντέλου αποκρύπτει από τις εφαρμογές και τις πηγές δεδομένων το μεγαλύτερο μέρος της πολυπλοκότητας των οντολογικών μοντέλων. Ωστόσο, οι οντολογίες διατηρούν τη δυνατότητα να χρησιμοποιήσουν εσωτερικά την επιπλέον εκφραστικότητα είτε για λόγους μοντελοποίησης είναι για υποστήριξη της μηχανής λογικής τους.

6.1.2 Η Meta-Class

Η αντικειμενοστραφής απεικόνιση του μοντέλου αναπαράστασης γνώσης και δεδομένων του Ozone βασίζεται σε μια μετακλάση (meta-class). Χρησιμοποιούμε αυτό τον όρο για να καταδείξουμε ότι τα instances της κλάσης αντιπροσωπεύουν κλάσεις και όχι αντικείμενα. Αυτή η πρόταση αφορά τη σημασιολογία της κλάσης. Όσο αφορά τη σύνταξη της σε όρους αντικειμενοστραφούς προγραμματισμού, η meta-class είναι μια κανονική κλάση και έχει ως instances κανονικά αντικείμενα. Η μετακλάση προσφέρει ένα σημαντικό πλεονέκτημα: αποτελεί μια μοναδική δομή που μπορεί να χρησιμοποιηθεί για αναπαραστήσει τόσο έννοιες, όσο και instances εννοιών αλλά ακόμα και ερωτήματα πάνω σε έννοιες. Αυτό επιτυγχάνεται με μια διαφοροποίηση στην ερμηνεία της σημασιολογίας της, κατά περίπτωση (δεδομένα ή ερώτημα). Στη συνέχεια περιγράφεται η κλάση αυτή και εξηγείται η χρήση της για την αναπαράσταση οντολογικής γνώσης, ερωτημάτων και δεδομένων.

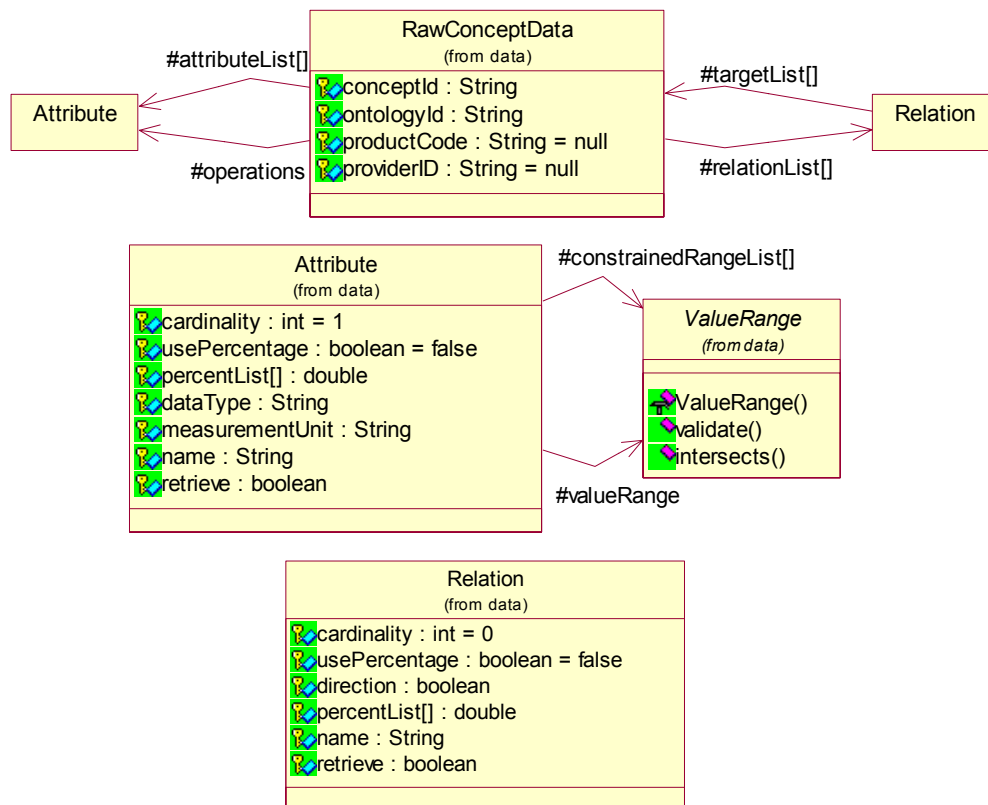
Ένα instance της κλάσης συνδέεται με μια οντολογική έννοια και αναπαριστά ένα σύνολο από πραγματικά αντικείμενα στην περιοχή γνώσης (domain) υπό εξέταση. Αυτό το σύνολο μπορεί να ταυτίζεται με το σύνολο αντικειμένων που αντιπροσωπεύει η ίδια η έννοια, ή μπορεί να αποτελεί ένα υποσύνολο του. Έτσι, αν $c \in O$ είναι μια έννοια της οντολογίας O , $mc(c)$ είναι ένα instance της meta-

class συνδεδεμένο με την έννοια c , και $Inst(c)$ είναι το σύνολο όλων των αντικειμένων που αναπαριστά η c , τότε

$$Inst(mc(c)) \subseteq Inst(c)$$

Στο Σχήμα 16 φαίνεται ο ορισμός της μετακλάσης σε UML[OMG, 2003], όπως υλοποιήθηκε σε Java για το πειραματικό σύστημα. Η αναπαράσταση των ιδιοτήτων και σχέσεων της έννοιας επιτυγχάνεται με δύο πίνακες αντικειμένων Attribute και Relation, αντίστοιχα. Η κλάση Attribute αναπαριστά τα βασικά χαρακτηριστικά μιας ιδιότητας, ανεξάρτητα από τον τύπο της. Τα χαρακτηριστικά της ιδιότητας που εξαρτώνται από τον τύπο της αναπαριστώνται με χρήση μιας υποκλάσης της ValueRange. Η σχετική ιεραρχία φαίνεται στο Σχήμα 17. Για κάθε τύπο ιδιότητας ορίζεται και μια υποκλάση. Οι τύποι που υποστηρίζονται εγγενώς από το Ozone είναι: ακέραιοι και πραγματικοί αριθμοί, ημερομηνίες/ώρες, αλφαριθμητικά και σύμβολα. Οι υποκλάσεις της ValueRange μοντελοποιούν τα χαρακτηριστικά των τιμών κάθε τύπου ιδιότητας.

Instances υποκλάσεων της ValueRange χρησιμοποιούνται για να καταδείξουν τις δυνατές, τις επιθυμητές ή τις πραγματικές τιμές μιας ιδιότητας στα πλαίσια αναπαράστασης γνώσης, ερωτημάτων και δεδομένων αντίστοιχα. Για το σκοπό αυτό, η κλάση Attribute περιέχει ένα βασικό αντικείμενο ValueRange που περιγράφει τις δυνατές τιμές της ιδιότητας με βάση τον οντολογικό της ορισμό. Περιέχει επίσης ένα πίνακα από αντικείμενα ValueRange που περιγράφουν είτε επιθυμητές είτε πραγματικές τιμές, ανάλογα με την περίπτωση (ερώτημα ή δεδομένα). Ο πίνακας είναι απαραίτητος για τις περιπτώσεις ιδιοτήτων με πολλαπλότητα τιμών (cardinality) μεγαλύτερη της μονάδας.



Σχήμα 16: Οι βασικές κλάσεις του Ozone σε UML, με την μετακλάση (RawConceptData) στο πάνω μέρος

Οι ακέραιοι και οι πραγματικοί αριθμοί μοντελοποιούνται με δύο σχεδόν πανομοιότυπες κλάσεις, την LongValueRange και την DoubleValueRange αντίστοιχα. Οι δύο κλάσεις επιτρέπουν τον ορισμό ενός κλειστού διαστήματος τιμών για το αντίστοιχο χαρακτηριστικό, καθώς και τον ορισμό του ελάχιστου βήματος αύξησης των τιμών. Αυτό το χαρακτηριστικό είναι θεμελιώδες στην αναπαράσταση γνώσης και ερωτημάτων, και χρησιμεύει τόσο για να ορίσει την ακρίβεια με την οποία είναι νοητός ο προσδιορισμός της ιδιότητας, όσο και για να καθορίσει το βήμα σε περιπτώσεις ιδιοτήτων που δεν λαμβάνουν συνεχείς τιμές. Για παράδειγμα, κατά τη μοντελοποίηση του ύψους ενός ανθρώπου σε μέτρα, ο σχεδιαστής της οντολογίας μπορεί να αποφασίσει ότι έχουν νόημα τιμές με ακρίβεια εκατοστού κατά μέγιστο. Έτσι θα ορίσει το βήμα για την αντίστοιχη ιδιότητα στο 0.01. Αξίζει να σημειωθεί ότι με αυτόν τον τρόπο η κλάση DoubleValueRange μπορεί να χρησιμοποιηθεί και για την αναπαράσταση

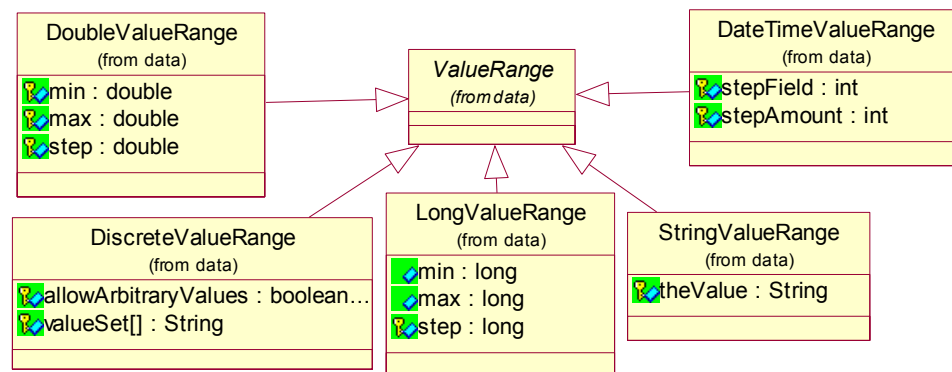
ακεραίων. Ωστόσο κάτι τέτοιο δεν είναι πρακτικό, αφού η ξεχωριστή κλάση `LongValueRange` δίνει καλύτερες επιδόσεις και αποφεύγει την ανάγκη μετατροπής από πραγματικούς σε ακέραιους στις πηγές δεδομένων.

Η κλάση `DateTimeValueRange` μπορεί να χρησιμοποιηθεί για να αναπαραστήσει είτε μια στιγμή, δηλαδή ένα σημείο στον άξονα του χρόνου, ή ένα χρονικό διάστημα. Η δυνατότητα καθορισμού διαστημάτων είναι αντίστοιχη με αυτήν για τις αριθμητικές κλάσεις. Το βήμα μπορεί να προσδιοριστεί και εδώ, όπου όμως η κατάσταση είναι αρκετά πιο περίπλοκη εξαιτίας του γεγονότος ότι η μέτρηση του χρόνου δεν βασίζεται σε δεκαδικές κλίμακες. Αντιμετωπίζουμε αυτό το πρόβλημα ορίζοντας για το βήμα τόσο την τιμή του, όσο και την μονάδα του. Η μονάδα δεν ορίζεται αυθαίρετα, αλλά σε σχέση με το πεδίο της ημερομηνίας/ώρας στο οποίο αναφέρεται (χρόνος, μήνας, ημέρα, ώρα, λεπτό κλπ).

Οι ιδιότητες με αλφαριθμητικές τιμές, που αναπαριστώνται με την κλάση `StringValueRange`, είναι οι απλούστερες. Μπορούν να δεχθούν οποιοδήποτε αλφαριθμητικό σαν τιμή, με μόνο περιορισμό το μήκος του.

Τέλος, η κλάση `DiscreteValueRange` αναπαριστά ιδιότητες που λαμβάνουν τιμές από ένα διακριτό σύνολο. Τέτοιες ιδιότητες καλούνται συχνά `LoV` (`List of Values`). Η `DiscreteValueRange` επιτρέπει τον προσδιορισμό των επιτρεπτών τιμών, καθώς και του κατά πόσο το σύνολο τιμών είναι κλειστό ή ανοιχτό. Κλειστά `LoV` μπορούν να πάρουν τιμές μόνο από το σύνολο τιμών τους, ενώ τα ανοιχτά (`unbound`) `LoV` μπορούν να δεχθούν οποιοδήποτε άλλο αλφαριθμητικό ως τιμή. Αυτό είναι πολύ χρήσιμο στην περίπτωση που είναι είτε αδύνατο είτε δύσκολο να καθοριστούν όλες οι πιθανές τιμές, οπότε ένα σύνολο αντιπροσωπευτικών, μη περιοριστικών τιμών μπορεί να ενισχύσει τη χρηστικότητα και να συμπληρώσει τη σημασιολογία της ιδιότητας. Η δοκιμή στην πράξη, μέσω της πειραματικής υλοποίησης, έδειξε ότι τα ανοιχτά συμβολικά σύνολα είναι χρήσιμα για ιδιότητες όπως «αγαπημένο χρώμα

πελάτη», όπου το σύνολο τιμών περιείχε συχνά χρησιμοποιούμενα χρώματα. Είναι επίσης χρήσιμα σαν εργαλείο αποτίμησης του οντολογικού μοντέλου, αφού μπορούν να χρησιμοποιηθούν σε επίπεδο άλφα ή βήτα τεστ της οντολογίας για να συγκεντρωθούν επιπλέον τιμές που δεν έχουν περιληφθεί στο μοντέλο.



Σχήμα 17: Η ιεραρχία της κλάσης ValueRange σε UML

Οι σχέσεις αντιμετωπίζονται με παρόμοιο τρόπο. Κάθε σχέση αναπαρίσταται με ένα instance της κλάσης Relation, που περιέχει μια αναφορά στην οντολογική σχέση που αντιπροσωπεύει. Ένα αντικείμενο Relation συνδέει δύο instances της meta-class, που αναπαριστούν τα δύο μέρη της σχέσης. Η μέγιστη πολλαπλότητα της σχέσης μπορεί επίσης να οριστεί, ενώ ένα πεδίο κατεύθυνσης δίνει τη δυνατότητα αναπαράστασης αντίστροφων σχέσεων. Η κλάση δίνει τη δυνατότητα αναπαράστασης σχέσεων ένα-προς-ένα και ένα-προς-πολλά, επιτρέποντας μόνο ένα αντικείμενο αφετηρίας αλλά πολλά αντικείμενα προορισμού μέσω ενός πίνακα. Τα αντικείμενα προορισμού προστίθενται μέσω της αντίστοιχης μεθόδου, η οποία πραγματοποιεί και τον έλεγχο πολλαπλότητας κατά την πρόσθεση.

6.1.3 Αναπαράσταση Γνώσης, Ερωτήσεων και Δεδομένων

Αντικείμενα της meta-class χρησιμοποιούνται για την αναπαράσταση γνώσης, ερωτημάτων και δεδομένων. Η σημασιολογία της κλάσης διαφοροποιείται

ανάλογα με την περίπτωση για το σκοπό αυτό. Στην πρώτη περίπτωση, της αναπαράστασης γνώσης, ένα instance της meta-class χρησιμοποιείται για να μεταδώσει την περιγραφή μιας έννοιας, όπως αυτή είναι αποθηκευμένη στην οντολογία. Το instance περιέχει μια αναφορά στην έννοια που αντιπροσωπεύει, ενώ οι ιδιότητες και οι σχέσεις του μοντελοποιούνται με αντικείμενα Attribute, Relation και RawConceptData (για τους προορισμούς σχέσεων) όπως εξηγήθηκε παραπάνω. Τελικά όλη η πληροφορία που απαιτείται για την περιγραφή της έννοιας (στα πλαίσια της εκφραστικής δυνατότητας του Ozone) βρίσκεται μέσα στο instance της meta-class.

Στην περίπτωση αυτή, όλοι οι περιορισμοί που περιέχονται στο αντικείμενο αυτό, όπως πολλαπλότητες τιμών, διαστήματα τιμών κλπ, προέρχονται από την οντολογική περιγραφή της έννοιας. Έτσι, η μεταβλητή valueRange προσδιορίζει τις *δυνατές* τιμές κάθε ιδιότητας. Η constrainedRangeList θα είναι κενή. Οι σχέσεις θα περιέχουν τη μέγιστη δυνατή πολλαπλότητα και θα περιέχουν αναφορές σε αντικείμενα προορισμού που θα καθορίζουν τον επιτρεπτό συνδυασμό ιδιοτήτων των εννοιών προορισμού των σχέσεων.

Κατά την αναπαράσταση ερωτημάτων το αντικείμενο της meta-class συμπληρώνεται με τον ίδιο τρόπο, με τη διαφορά ότι οι μεταβλητές constrainedRangeList περιέχουν πλέον αντικείμενα ValueRange που προσδιορίζουν τις *επιθυμητές* τιμές των αντίστοιχων ιδιοτήτων. Για παράδειγμα, έστω μια έννοια «Τηλεόραση» με μια ιδιότητα «Διαγώνιος». Η μεταβλητή valueRange μπορεί να περιέχει ένα αντικείμενο LongValueRange που καθορίζει ένα διάστημα τιμών από 6 ως 60 ίντσες με βήμα 0.5, αναπαριστώντας το σύνολο των δυνατών τιμών της διαγωνίου μιας τηλεόρασης. Στα πλαίσια ενός συγκεκριμένου ερωτήματος, η μεταβλητή constrainedRangeList θα μπορούσε να προσδιορίσει ένα πιο στενό διάστημα τιμών, για παράδειγμα 29 με 32 ίντσες, για να προσδιορίσει ότι μόνο τηλεοράσεις με τιμές διαγωνίου στην περιοχή αυτή είναι επιθυμητές. Αν η constrainedRangeList περιέχει περισσότερα του ενός

αντικείμενα ValueRange που προσδιορίζουν περιορισμούς στο ερώτημα, τότε το Ozone υποθέτει μια σχέση OR μεταξύ τους.

Η constrainedRangeList χρησιμοποιείται επίσης και στην αναπαράσταση των αποτελεσμάτων ενός ερωτήματος, δηλαδή των ίδιων των δεδομένων. Εδώ πλέον κάθε τιμή αντιπροσωπεύει ένα πραγματικό αντικείμενο που βρέθηκε στη βάση δεδομένων. Στην περίπτωση αυτή πρέπει να σημειωθεί ότι υπάρχει μια διαφοροποίηση στη σημασιολογία των κλάσεων Attribute και Relation. Οι πολλαπλότητες που επιστρέφονται δεν είναι πλέον οι μέγιστες δυνατές, αλλά το πραγματικό πλήθος τιμών ή σχέσεων που το συγκεκριμένο αντικείμενο περιέχει. Για παράδειγμα, έστω δύο έννοιες «πατέρας» και «παιδί» με την προφανή σχέση μεταξύ τους. Στην περίπτωση της αναπαράστασης γνώσης, η σχέση αυτή θα μοντελοποιηθεί με ένα αντικείμενο Relation με άπειρη πολλαπλότητα. Αν όμως θελήσουμε να αναπαραστήσουμε ένα πραγματικό πρόσωπο, τον Πέτρο, που έχει δύο παιδιά, τη Μαρία και την Αλεξάνδρα, τότε το instance της meta-class που αντιπροσωπεύει την έννοια «πατέρας» και αναφέρεται στον Πέτρο, η σχέση πατρότητας θα αναφέρει πολλαπλότητα ακριβώς δύο. Αντίστοιχα, η constrainedRangeList περιέχει τις πραγματικές, κι όχι τις επιθυμητές τιμές. Εδώ επίσης η πολλαπλότητα κάθε ιδιότητας θα είναι ίση με τον αριθμό των αντικειμένων που περιέχονται στην constrainedRangeList για το συγκεκριμένο αντικείμενο.

Η πρόσβαση σε ομοσπονδίες πηγών δεδομένων είναι αρκετά απλή με αυτήν την προσέγγιση. Το τυπικό σενάριο τριών βημάτων που αντιμετωπίζουν οι προγραμματιστές εφαρμογών είναι: ανάκτηση της περιγραφής μιας έννοιας, σύνθεση και εκτέλεση του ερωτήματος, ανάκτηση και απεικόνιση των αποτελεσμάτων. Η ανάκτηση της περιγραφής της έννοιας πραγματοποιείται με την αποστολή στο Ozone του ονόματος της έννοιας και της αντίστοιχης οντολογίας, και επιστροφή ενός αντικειμένου meta-class με καθορισμένες σχέσεις, ιδιότητες και περιορισμούς επί αυτών. Η σύνθεση του ερωτήματος

μεταφράζεται στη δημιουργία αντικειμένων ValueRange και πρόσθεση τους στην constrainedRangeList. Όλες οι κλάσεις της οικογένειας ValueRange προσφέρουν μια μέθοδο validate() που ελέγχει ότι οι περιορισμοί του ερωτήματος δεν παραβιάζουν τον ορισμό της έννοιας (δηλαδή υπακούουν στον περιορισμό που κωδικοποιείται στο valueRange). Στο σημείο αυτό ο προγραμματιστής μπορεί να επιλέξει τις ιδιότητες και σχέσεις που τον ενδιαφέρει να ανακτήσει, στην περίπτωση που δεν τον ενδιαφέρουν όλες, θέτοντας αντίστοιχα την τιμή της μεταβλητής retrieve. Αυτό είναι ιδιαίτερα χρήσιμο για τις σχέσεις που, αν ανακτηθούν πλήρως, μπορεί να οδηγήσουν σε πολύ περίπλοκα ερωτήματα σε SQL. Στη συνέχεια το meta-class instance στέλνεται στο Ozone, που το διανέμει στις πηγές δεδομένων που μπορούν να το χειριστούν. Τελικά, η απάντηση που επιστρέφεται στην εφαρμογή είναι μια λίστα από γνώριμα αντικείμενα meta-class που μπορούν να παρασταθούν γραφικά για εμφάνιση στο χρήστη ακριβώς με τον ίδιο τρόπο που χρησιμοποιήθηκε για την απεικόνιση της περιγραφής της έννοιας.

6.2 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΚΑΙ ΛΕΙΤΟΥΡΓΙΑ

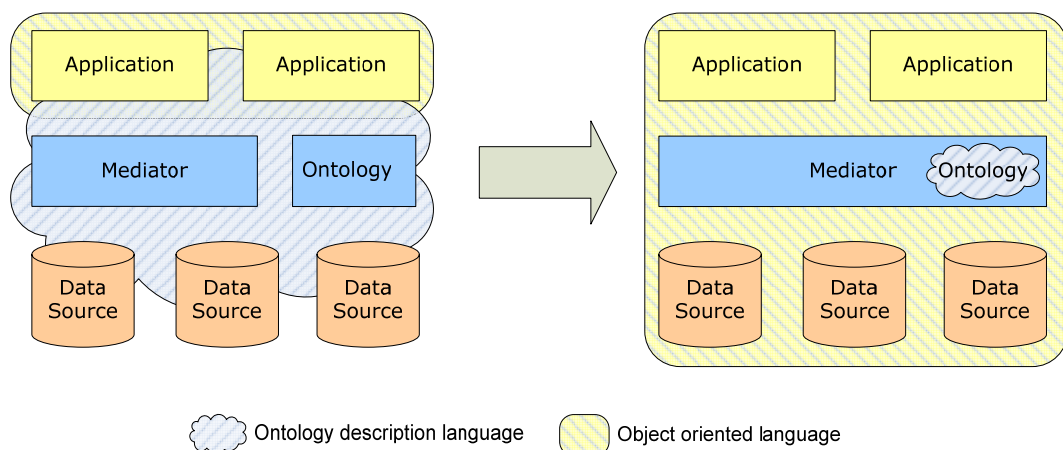
6.2.1 Το Πρόβλημα

Υπάρχει μεγάλος όγκος ερευνητικής δουλειάς που καταδεικνύει τη σημασία των οντολογιών στη σημασιολογική ολοκλήρωση συστημάτων, ήδη από τα μέσα της δεκαετίας του 1990. Ωστόσο, η υιοθέτηση της μεθόδου στη βιομηχανία είναι μέχρι στιγμής ανύπαρκτη.

Το επιχείρημα που αναπτύσσουμε στη συνέχεια είναι ότι η αδυναμία υιοθέτησης οφείλεται στο ακατάλληλο των μεθόδων, εργαλείων και γλωσσών που χρησιμοποιούνται στις οντολογίες για βιομηχανική χρήση, σε συνδυασμό με την άναρχη αρχιτεκτονική των πειραματικών συστημάτων και την έλλειψη δομημένων σχεδιαστικών προτύπων κατασκευής τους, που εκθέτει στην ανάγκη

χρήσης αυτών των μεθόδων και εργαλείων όλα τα συστήματα που συμμετέχουν ή αλληλεπιδρούν με την ομοσπονδία πηγών δεδομένων. Αυτό οδηγεί σε ένα πρότυπο απαγορευτικό για βιομηχανική εφαρμογή, στο οποίο το μοντέλο αναπαράστασης γνώσης και δεδομένων, η γλώσσα προγραμματισμού και η πλατφόρμα ανάπτυξης της οντολογίας βρίσκονται στο επίκεντρο του ολοκληρωμένου συστήματος και χρησιμοποιούνται για την επικοινωνία μεταξύ όλων των υποσυστημάτων.

Αυτή η προσέγγιση απεικονίζεται σχηματικά αριστερά στο Σχήμα 18. Το ολοκληρωμένο σύστημα διαχωρίζεται σε τρία επίπεδα (layers). Στο χαμηλότερο επίπεδο βρίσκονται οι πηγές δεδομένων. Το δεύτερο επίπεδο πραγματοποιεί την ολοκλήρωση τους και ονομάζεται επίπεδο ομοσπονδίας, ολοκλήρωσης, ομογενοποίησης ή διαμεσολάβησης (mediation). Το επίπεδο αυτό καλύπτει τις ιδιαιτερότητες κάθε πηγής και παρουσιάζει προς τα πάνω μια ομογενοποιημένη ομοσπονδία. Στην κορυφή βρίσκεται το επίπεδο εφαρμογών, στο οποίο βρίσκονται οι εφαρμογές που χρησιμοποιούν τα ομοσπονδιακά δεδομένα.



Σχήμα 18: Η ορατότητα των οντολογιών και η έκταση χρήσης γλώσσας περιγραφής οντολογίας στην κλασική προσέγγιση ολοκλήρωσης με οντολογίες (αριστερά) και στο Ozone (δεξιά)

Το επίπεδο ολοκλήρωσης αναλαμβάνει και τον μετασχηματισμό της σημασιολογίας και της αναπαράστασης μεταξύ των πηγών δεδομένων και των

εφαρμογών. Αποτελείται από τις οντολογίες, που παρέχουν τη δυνατότητα ομογενοποίησης της σημασιολογίας, και άλλα υποσυστήματα διαμεσολάβησης που παρέχουν την απαραίτητη λειτουργικότητα για τη δημιουργία και την πρόσβαση στην ομοσπονδία πηγών, καθώς και το μετασχηματισμό αναπαράστασης. Στην πράξη, δεν υπάρχουν διακριτά όρια μεταξύ του επιπέδου εφαρμογών και του επιπέδου ολοκλήρωσης, όπως αυτά που φαίνονται στο σχήμα. Κατά κανόνα υπάρχει μόνο μια εφαρμογή και το επίπεδο ολοκλήρωσης με τις οντολογίες του είναι αναπόσπαστο τμήμα της. Αυτό σημαίνει ότι υπάρχει ένα επίπεδο πάνω από τις πηγές δεδομένων και όχι δύο, πράγμα που οδηγεί σε μια άμορφη αρχιτεκτονική συστήματος.

Στο μοντέλο αυτό, τόσο στην παραλλαγή του με τα δύο επίπεδα όσο και σε αυτή με τα τρία, υπάρχει η ανάγκη πολύ στενής αλληλεπίδρασης μεταξύ των εφαρμογών και των οντολογιών. Ουσιαστικά όλο το σύστημα χτίζεται γύρω από τις οντολογίες και όλα τα υποσυστήματα είναι εξαρτημένα από αυτές, με αποτέλεσμα να πρέπει να υιοθετήσουν την δική τους αναπαράσταση και προγραμματιστικό μοντέλο για να μπορέσουν να κατασκευάσουν ερωτήματα και να διαχειριστούν δεδομένα. Ιδιαίτερα το επίπεδο ολοκλήρωσης πρακτικά ταυτίζεται με την οντολογία, με την έννοια ότι χτίζεται πάνω στην πλατφόρμα της οντολογίας και μοιράζεται, όπως και τα υπόλοιπα υποσυστήματα, το μοντέλο και την αναπαράσταση δεδομένων της οντολογίας.

Από τη σκοπιά της βιομηχανικής κατασκευής λογισμικού αυτή η προσέγγιση παρουσιάζει σημαντικά προβλήματα. Στο κεφάλαιο 3 αναλύθηκαν οι γλώσσες, οι πλατφόρμες και οι τεχνικές που χρησιμοποιούνται για την ανάπτυξη και επεξεργασία οντολογιών. Είναι σαφές ότι οι μέθοδοι αναπαράστασης και οι γλώσσες που χρησιμοποιούνται είναι ιδιαίτερα πολύπλοκες και σαφώς διαφοροποιημένες από τα σημερινά βιομηχανικά πρότυπα (industrial standards). Επίσης δεν υπάρχει κάποια τυποποίηση στο χώρο των γλωσσών περιγραφής οντολογιών, αν και γλώσσες όπως η RDF [Brickley, 1999] και η OIL [Horrocks,

00] έχουν αναδειχθεί σε de facto σημεία αναφοράς. Οι οντολογικές γλώσσες πρέπει να επιλέξουν μια σχέση ισορροπίας ανάμεσα στην εκφραστικότητα και την υποστήριξη λογικών λειτουργιών, έτσι ώστε διαφορετικές γλώσσες παρουσιάζουν διαφορετικές δυνατότητες και δεν υπάρχουν γλώσσες που να είναι κατάλληλες για όλες τις χρήσεις. Προβλήματα που θεωρούνται λυμένα στις βιομηχανικές γλώσσες προγραμματισμού, όπως η συντηρησιμότητα, η υποστήριξη ομαδικής δουλειάς, η βελτιστοποίηση κλπ, δεν έχουν εξεταστεί καθόλου στα πλαίσια αυτών των γλωσσών. Συμπερασματικά, η χρήση τέτοιων γλωσσών σε βιομηχανικά έργα είναι αντίθετη με τις πρακτικές της βιομηχανίας και επιφέρει σημαντικά κόστη εκπαίδευσης προσωπικού, ανάπτυξης εργαλείων και διαχείρισης κινδύνων. Οι γλώσσες λογικού προγραμματισμού είναι πολύ διαδεδομένες στην ερευνητική κοινότητα εδώ και δεκαετίες, ωστόσο έχουν πολύ μικρή διείσδυση στη βιομηχανία μέχρι τώρα. Δεν υπάρχει κάποιο στοιχείο που να αφήνει περιθώρια να υποθέσει κανείς ότι αυτό θα αλλάξει στο σύντομο μέλλον.

Καταλήγουμε έτσι στο συμπέρασμα ότι η μέθοδος ολοκλήρωσης πηγών δεδομένων με οντολογίες, παρά το μεγάλο δυναμικό της και τα πλεονεκτήματα που προσφέρει, αποκλείεται από την εφαρμογή στη βιομηχανία και παραμένει ένα εργαστηριακό πείραμα εξαιτίας των προβληματικών μεθόδων εργαλείων που χρησιμοποιεί και της χαοτικής αρχιτεκτονικής της, που επιβάλλει την χρήση αυτών των μεθόδων και εργαλείων σε όλο το εύρος των εμπλεκόμενων στην ολοκλήρωση συστημάτων.

Αυτό είναι το πρόβλημα που επιχειρούμε να αντιμετωπίσουμε με το Ozone. Οι οντολογίες είναι μια τεχνολογία που μπορεί να επιτύχει τη σημασιολογική διαμεσολάβηση μεταξύ των εφαρμογών και των πηγών δεδομένων. Προσπαθήσαμε να αναπτύξουμε την άλλη όψη του νομίσματος: μια τεχνολογία που να επιτυγχάνει τη διαμεσολάβηση ανάμεσα στις οντολογίες από τη μια, και

την εφαρμογή και τις πηγές δεδομένων από την άλλη, σε επίπεδο αναπαράστασης, μετασχηματισμών και μεθοδολογιών.

Με το Ozone, οι οντολογίες περικλείονται στο εσωτερικό του επιπέδου ολοκλήρωσης και δεν βρίσκονται σε άμεση επαφή ούτε με τις εφαρμογές ούτε με τις βάσεις. Έτσι μεταβαίνουμε στο μοντέλο που φαίνεται στο δεξιό μέρος στο Σχήμα 18. Εδώ πλέον όλη η επαφή με τις οντολογίες γίνεται μέσω του επιπέδου ολοκλήρωσης, έτσι ώστε οι εφαρμογές και οι βάσεις δεδομένων απομονώνονται από τα περίπλοκα μοντέλα αναπαράστασης και τις γλώσσες λογικού προγραμματισμού. Όλη η αλληλεπίδραση τους πραγματοποιείται μέσω του επιπέδου ολοκλήρωσης με χρήση αντικειμενοστραφών μεθόδων αναπαράστασης. Το αντικειμενοστραφές μοντέλο είναι ευρύτατα διαδεδομένο που παράγει επαναχρησιμοποιήσιμο και εύκολα συντηρήσιμο κώδικα, και δίνει τη δυνατότητα αντιστοίχισης μεταξύ της προδιαγραφής, του σχεδιασμού και του παραγόμενου κώδικα. Έτσι το Ozone επιτρέπει την επικοινωνία με το ολοκληρωμένο σύστημα με βάση το πλέον διαδεδομένο βιομηχανικό πρότυπο στον προγραμματισμό.

6.2.2 Αρχιτεκτονική

Το Ozone είναι ένα πλαίσιο ολοκλήρωσης (integration framework). Στο [Armstrong, 1999] ένα πλαίσιο ολοκλήρωσης ορίζεται ως «... η υλοποίηση ενός συνόλου από διαπροσωπείες και κανόνες αλληλεπίδρασης που επιτρέπουν και ελέγχουν την επικοινωνία ανάμεσα σε συστήματα». Οι διαπροσωπείες εξ ορισμού εξαρτώνται από τα συστήματα που συνδέουν. Ωστόσο, είναι επιθυμητό να διατηρείται ο πυρήνας του πλαισίου ολοκλήρωσης (το κομμάτι δηλαδή που υλοποιεί τους κανόνες αλληλεπίδρασης) κατά το δυνατό ανεξάρτητο των συστημάτων που ολοκληρώνει. Αυτή η ιδιότητα προσφέρει στο πλαίσιο ολοκλήρωσης την δυνατότητα να ολοκληρώνει νέα συστήματα εύκολα. Ένα πλαίσιο που φέρει αυτή την ιδιότητα ονομάζεται συνήθως ανοιχτό (open), ελαστικό (flexible) ή επεκτάσιμο (expandable).

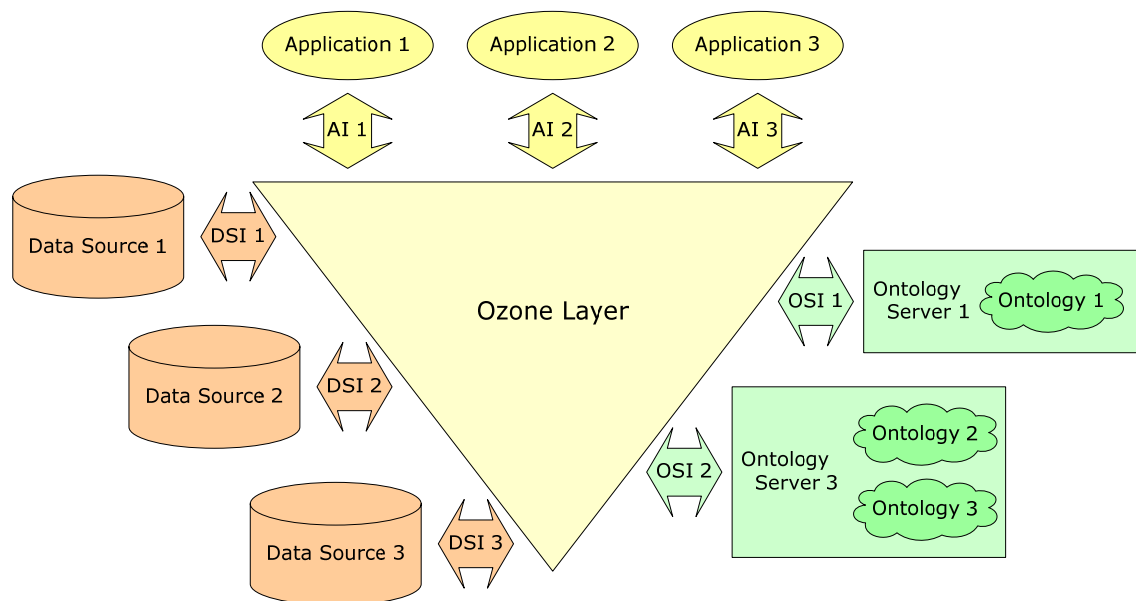
Το Ozone έχει σχεδιαστεί με στόχο να είναι όσο πιο ανοιχτό είναι δυνατό. Από πλευράς λοιπόν επεκτασιμότητας, οι προτεραιότητες είναι πρωτίστως προς την κατεύθυνση των πηγών δεδομένων, ιδιαίτερα σχεσιακών βάσεων. Ακολουθούν οι εφαρμογές και τρίτες έρχονται οι οντολογίες. Προκειμένου να εξασφαλίσουμε αυτή την επεκτασιμότητα επιδιώκουμε να τυποποιήσουμε την προδιαγραφή των interfaces προς τα τρία διαφορετικά είδη συστημάτων ώστε, όταν μια νέα πηγή δεδομένων, εφαρμογή ή οντολογία υλοποιήσει το τυποποιημένο interface να συνδέεται αρμονικά με το σύστημα χωρίς να υπάρχει ανάγκη επανασχεδιασμού στην πλευρά του Ozone.

Η προσπάθεια τυποποίησης απέδωσε πολύ καλά για τα interfaces προς πηγές δεδομένων, διότι οι λειτουργίες που προσφέρουν είναι σχετικά απλές και καλά κατανοητές. Οι εφαρμογές αποτελούν πολύ πιο δύσκολη περίπτωση λόγω της μεγάλης ποικιλομορφίας τους. Εδώ δεν μπορούμε να δώσουμε μια γενικευμένη λύση που να καλύπτει όλες τις πιθανές ανάγκες εφαρμογών. Κάτι τέτοιο θα ήταν πρακτικά ανέφικτο, αλλά ακόμα και στο βαθμό που θα μπορούσε να γίνει θα οδηγούσε σε ένα γιγάντιο, δύσχρηστο, υδροκέφαλο σύστημα ολοκλήρωσης. Επιδιώκοντας λοιπόν την απλότητα και τη δυνατότητα ρεαλιστικής χρηστικότητας του συστήματος πάνω από όλα, επιλέξαμε να περιορίσουμε την περιοχή εφαρμογών που επιδιώκει να καλύψει το Ozone από πλευράς λειτουργιών που υποστηρίζει, περιορίζοντας τις σε αυτές που είναι απαραίτητες για την πρόσβαση και ανάκτηση πληροφορίας με υποστήριξη απλών ερωτημάτων, εξασφαλίζοντας έτσι τη μέγιστη δυνατή ελαστικότητα από πλευράς χρηστικότητας του συστήματος, αλλά και ελαχιστοποιώντας το μέγεθος των interfaces και το συνεπακόλουθο κόστος υλοποίησης τους.

Η τελική πρόκληση από πλευράς επεκτασιμότητας αφορά τη σύνδεση νέων οντολογιών με το σύστημα. Προς αυτή την κατεύθυνση, διαχωρίζουμε τις οντολογίες από το πλαίσιο ολοκλήρωσης που έτσι αναδύεται ως μια αυτόνομη οντότητα με βάση τον ορισμό που δόθηκε παραπάνω για τα πλαίσια

ολοκλήρωσης, και δεν είναι απλώς ένα περίβλημα για την οντολογία του συστήματος.

Με βάση τα παραπάνω, μπορούμε τώρα να δούμε την αρχιτεκτονική του Ozone σε μεγαλύτερη λεπτομέρεια, αναλύοντας περαιτέρω την διαστρωματωμένη αρχιτεκτονική που φαίνεται αριστερά στο Σχήμα 18 ώστε να διαχωρίσουμε πλήρως τις οντολογίες από τον μηχανισμό ολοκλήρωσης.



Σχήμα 19: Η αρχιτεκτονική του Ozone

Το αποτέλεσμα φαίνεται στο Σχήμα 19. Το επίπεδο ομοσπονδίας έχει αναλυθεί στα δύο βασικά συστατικά του, δηλαδή τις πλατφόρμες οντολογιών και τον πυρήνα του Ozone, το Object-Oriented Ontology Access Layer (O₃AL), στον οποίο οφείλει και την ονομασία του. Η αρχιτεκτονική αυτή μπορεί να υποστηρίξει πολλαπλές οντολογίες, και η κάθε οντολογία είναι εγκατεστημένη στη δική της πλατφόρμα, η οποία μπορεί να παρέχει και μηχανή λογικής για να καλύψει τις ανάγκες της ίδιας της οντολογίας καθώς και την υλοποίηση του interface με το Ozone.

Ο διαχωρισμός των οντολογιών από τον πυρήνα του επιπέδου ολοκλήρωσης μας δίνει τη δυνατότητα να προσθέτουμε οντολογίες στο σύστημα χωρίς αλλαγή

σε αυτό, ωστόσο η διασύνδεση μιας νέας οντολογίας δεν είναι τόσο απλή όσο η διασύνδεση μιας πηγής δεδομένων. Οι οντολογικές πλατφόρμες είναι πολύ διαφορετικές μεταξύ τους, με αποτέλεσμα να είναι αδύνατο να δώσουμε τυποποιημένα, παραμετροποιήσιμα interface templates ή ακόμα και οδηγούς για την ανάπτυξη τους, όπως κάνουμε για τις βάσεις δεδομένων παρακάτω. Έτσι περιοριζόμαστε στην σαφή προδιαγραφή του interface που θα πρέπει μια οντολογία να υλοποιήσει για να συνδεθεί με το Ozone. Αυτό δίνει τη δυνατότητα στις εφαρμογές και τις πηγές δεδομένων να χρησιμοποιήσουν πολλαπλές διαφορετικές οντολογίες, αλλά φυσικά δεν ολοκληρώνει αυτές τις οντολογίες μεταξύ τους. Η οντολογική ολοκλήρωση είναι πέρα από τα πλαίσια της παρούσας έρευνας και αποτελεί ένα ιδιαίτερα ελκυστικό και πολύπλοκο πρόβλημα [Omelayenko, 2001].

Ακόμα σημαντικότερη είναι η δυνατότητα απεξάρτησης από την αναπαράσταση δεδομένων και γνώσης των οντολογιών, και υιοθέτησης ενός μοντέλου αναπαράστασης που να είναι προσαρμοσμένο στις ανάγκες των εφαρμογών και των πηγών δεδομένων. Στην αρχιτεκτονική αυτή, το Ozone είναι ένα τριμερές interface ανάμεσα στις εφαρμογές, τις πηγές δεδομένων και τις οντολογίες. Διαχωρίζει το πίσω μέρος του συστήματος (δηλαδή τις πηγές δεδομένων) από το εμπρός μέρος (τις εφαρμογές) που υλοποιεί τη διαπροσωπεία με τους χρήστες, τους επιχειρησιακούς κανόνες και τη λογική. Επίσης, διαμεσολαβεί ανάμεσα στα δύο αυτά μέρη και τις οντολογίες εξασφαλίζοντας έτσι την απομόνωση τους από τα υπόλοιπα υποσυστήματα. Τέλος, οριοθετεί την χρήση των ιδιαίτερων μεθόδων αναπαράστασης και προγραμματιστικών προτύπων. Η χρήση γλωσσών περιγραφής οντολογιών περιορίζεται δεξιά από το Ozone και φαίνεται με πράσινο χρώμα στο σχήμα. Οι πηγές δεδομένων χρησιμοποιούν σχεσιακά, ιεραρχικά ή άλλα μοντέλα. Η έκταση χρήσης αυτών των μοντέλων περιορίζεται στα αριστερά του Ozone και φαίνεται με πορτοκαλί. Τέλος, το ίδιο το Ozone είναι καθαρά αντικειμενοστραφές, όπως

και το interface που δίνει προς τις εφαρμογές. Οι εφαρμογές μπορούν να χρησιμοποιούν όποιο προγραμματιστικό πρότυπο επιλέξουν, αλλά η διασύνδεση με το Ozone γίνεται με αντικειμενοστραφή τρόπο.

Τα βέλη διπλής κατεύθυνσης στο Σχήμα 19 υποδηλώνουν τα interfaces μεταξύ του Ozone και των εξωτερικών συστημάτων. Κάθε νέο σύστημα που προσαρτάται στην ομοσπονδία πρέπει να υλοποιήσει ένα τέτοιο interface. Αυτά τα interfaces έχουν μεγάλη σημασία για τη λειτουργία του συστήματος, και το μεγαλύτερο μέρος της δουλειάς και του κόστους προσάρτησης ενός εξωτερικού συστήματος δαπανάται για την κατασκευή του αντίστοιχου interface. Ιδιαίτερα τα interfaces των πηγών δεδομένων εκτελούν το σημαντικότερο ίσως μέρος της επεξεργασίας ερωτημάτων, αυτό του μετασχηματισμού του ερωτήματος από την γενική αναπαράσταση, στην εγγενή αναπαράσταση της πηγής.

Ένας από τους στόχους της παρούσας έρευνας είναι η τυποποίηση όλων των προγραμματιστικών interfaces του συστήματος. Αποδεικνύεται ότι αυτό είναι ένα δύσκολο εγχείρημα, καθώς κάθε σύστημα φέρει τις δικές του ανάγκες, ιδιότητες και ιδιομορφίες. Δεδομένου όμως του μεγάλου βαθμού τυποποίησης στις πηγές δεδομένων, τουλάχιστον όταν περιοριζόμαστε σε σχεσιακές βάσεις, και του περιορισμού της εξεταζόμενης περιοχής εφαρμογών στο e-commerce, είναι δυνατό να γίνει ένας διεξοδικός ορισμός των interfaces στην πλευρά των πηγών δεδομένων και των οντολογιών, και να παγιωθεί ένα αρκετά σταθερό interface προς την πλευρά της εφαρμογής. Ένας διεξοδικός ορισμός του interface προς τις εφαρμογές, ακόμα και για μια περιορισμένη περιοχή, απαιτεί τον πειραματισμό με ένα μεγάλο πλήθος πραγματικών εφαρμογών. Η διαθεσιμότητα πλήθους εφαρμογών είναι δύσκολο να εξασφαλιστεί, έτσι κάτι τέτοιο δεν κατέστη δυνατό να γίνει στα πλαίσια αυτής της δουλειάς. Πρέπει όμως να σημειωθεί ότι η σημασία της τυποποίησης αυτού του interface είναι σαφώς μικρότερη σε σχέση με τα άλλα δύο, για τον απλό λόγο ότι είναι πολύ συχνότερη η πρόσθεση μιας πηγής δεδομένων ή μιας οντολογίας σε μια

ομοσπονδία, παρά μιας νέας εφαρμογής. Αυτή ήταν και η περίπτωση στην πειραματική υλοποίηση μας.

6.2.3 Η Διαδικασία Ολοκλήρωσης

Το βασικό σενάριο που εξετάζουμε για την κατασκευή ενός ολοκληρωμένου συστήματος με το Ozone, που θεωρούμε το πλέον χαρακτηριστικό, αφορά την ανάπτυξη εφαρμογής η οποία αντλεί δεδομένα από διάφορες πηγές δεδομένων. Οι απαραίτητες οντολογίες για την ολοκλήρωση των πηγών δεδομένων μπορεί να υπάρχουν ήδη, ή να πρέπει να αναπτυχθούν από το μηδέν. Αφού συλλεχθούν οι προδιαγραφές της εφαρμογής, νέες οντολογίες για την υποστήριξη της μπορούν να αναπτυχθούν. Η κατασκευή των οντολογιών γίνεται από μια μεικτή ομάδα που απαρτίζεται από ειδικούς σε οντολογίες και ειδικούς στην περιοχή εφαρμογής που θα καλυφθεί. Στη συνέχεια αναπτύσσεται το interface που συνδέει την οντολογική πλατφόρμα με το Ozone. Αυτή η εργασία χρειάζεται γνώσεις για την οντολογική πλατφόρμα και ικανότητες ανάπτυξης λογισμικού σε Java. Παράλληλα εξελίσσεται η ανάπτυξη της εφαρμογής, και του interface της με το Ozone. Αυτή η εργασία χρειάζεται μόνο ικανότητες ανάπτυξης λογισμικού σε Java και δεν απαιτεί πλέον οντολογική γνώση.

Απομένει η διασύνδεση των πηγών δεδομένων με το ολοκληρωμένο σύστημα, μια εργασία που επαναλαμβάνεται τόσες φορές όσες είναι οι πηγές δεδομένων. Η ομάδα εργασίας που αναπτύσσει αυτά τα interfaces χρειάζεται πολύ καλή γνώση της εγγενούς γλώσσας ερωτημάτων της πηγής (συνήθως κάποια διάλεκτο SQL όταν έχουμε σχεσιακές βάσεις), καλή κατανόηση του σχήματος των δεδομένων και της σημασιολογίας τους, και την ικανότητα ανάπτυξης λογισμικού σε Java. Έτσι αυτή η ομάδα θα πρέπει να περιλαμβάνει προγραμματιστές Java εξοικειωμένους με το Ozone, αναλυτές δεδομένων ειδικευμένους στη συγκεκριμένη πηγή, και διαχειριστές ή προγραμματιστές με γνώση στην πλατφόρμα της συγκεκριμένης πηγής.

Για σχεσιακές βάσεις, η ανάπτυξη του interface προς το Ozone μπορεί να διαιρεθεί στις ακόλουθες εργασίες:

- Δημιουργία μιας απεικόνισης οντολογίας-σχήματος, η οποία αντιστοιχίζει έννοιες της οντολογίας σε οντότητες του σχήματος.
- Δημιουργία των λειτουργιών μετασχηματισμού που μεταφράζουν τα αντικείμενα ερωτημάτων του Ozone στην εγγενή γλώσσα της βάσης, με χρήση της παραπάνω απεικόνισης.
- Δημιουργία των λειτουργιών μετασχηματισμού που μετατρέπουν το σύνολο αποτελεσμάτων των ερωτημάτων από την εγγενή αναπαράσταση της βάσης στην αναπαράσταση του Ozone.
- Ανάπτυξη του προγραμματιστικού σκελετού για τη οργάνωση των δύο προηγούμενων ομάδων λειτουργιών σε μια ολοκληρωμένη λειτουργική οντότητα που να υπακούει στην προδιαγραφή του interface του Ozone.

Με αυτή τη διαίρεση επιχειρείται ο διαχωρισμός των εργασιών σε τμήματα που να μπορούν να τα εκτελέσουν αυτόνομα ομάδες με την ελάχιστη δυνατή μίξη γνώσεων. Έτσι, η πρώτη εργασία εκτελείται από τους αναλυτές δεδομένων με υποστήριξη από ειδικούς οντολογιών. Η δεύτερη και η τρίτη εκτελούνται από διαχειριστές ή προγραμματιστές βάσεων και προγραμματιστές Java. Τέλος η τέταρτη πραγματοποιείται από προγραμματιστές Java.

Ο διαχωρισμός ανάμεσα στα συστατικά στοιχεία του συστήματος που προσφέρει το Ozone μας επιτρέπει να περιγράψουμε αυτή τη διαδικασία σε μεγάλη λεπτομέρεια, η οποία μπορεί να καταγραφεί σαν μια σειρά από σταθερά και καλά ορισμένα βήματα. Επίσης, η ημιαυτοματοποίηση αυτής της διαδικασίας είναι δυνατή. Για το σκοπό αυτό μπορεί να αναπτυχθεί ένα εργαλείο απεικόνισης που καθοδηγεί τους χρήστες του στην δημιουργία της απεικόνισης σχήματος-οντολογίας και δίνει ως έξοδο την απεικόνιση σε μια τυποποιημένη δομή. Η δομή αυτή χρησιμοποιείται από ένα επίσης τυποποιημένο πρόγραμμα Java που

ονομάζουμε *interface template*. Το πρόγραμμα αυτό προσφέρει έτοιμες τις λειτουργίες των τριών επόμενων βημάτων. Η κωδικοποίηση του σκελετού του τελευταίου βήματος με γενικό τρόπο δεν παρουσιάζει δυσκολία. Για την τυποποίηση των άλλων δύο βημάτων, βασιζόμαστε στην υπόθεση ότι η βάση είναι σχεσιακή και υποστηρίζει διασύνδεση με JDBC[Ellis, 2001]. Αυτή είναι μια πολύ βάσιμη υπόθεση που ισχύει για την συντριπτική πλειοψηφία των εμπορικών και *public domain* βάσεων δεδομένων. Ακόμα και με αυτά τα εργαλεία, σε αρκετές περιπτώσεις μπορεί να απαιτηθεί η επέμβαση στο *interface template* από προγραμματιστή, είτε για διορθώσεις στο κομμάτι του μετασχηματισμού των αποτελεσμάτων, είτε για βελτιστοποίηση των ερωτημάτων SQL που παράγει το εργαλείο αντιστοίχησης. Η βελτιστοποίηση εξαρτάται από την παραμετροποίηση, μοντέλα χρήσης και άλλες ιδιότητες της βάσης, και είναι αδύνατο να γίνει αυτόματα. Η διόρθωση στους μετασχηματισμούς μπορεί να χρειαστεί σε περιπτώσεις που δεν υπάρχει αναλυτική αποθήκευση οντοτήτων στη βάση, αλλά αντί αυτού αποθηκεύονται κατηγορίες οντοτήτων και δυνατές τιμές παραμέτρων. Αυτό συμβαίνει για παράδειγμα σε βάσεις δεδομένων προϊόντων που δημιουργούνται κατά παραγγελία, έτσι ώστε η βάση δεν κρατά πραγματικά προϊόντα αλλά μάλλον τις περιγραφές τους. Πρόκειται στην ουσία για μια βάση γνώσης, όχι μια βάση δεδομένων. Ακόμα όμως και σε αυτές τις περιπτώσεις, η δουλειά και το κόστος που απαιτείται για την δημιουργία του *interface* μειώνονται με τη χρήση των εργαλείων που περιγράψαμε σε λιγότερο από το μισό[Stavroulos, 2002a].

6.2.4 Διατύπωση, Επεξεργασία και Εκτέλεση Ερωτημάτων

Η διενέργεια ερωτημάτων και η ανάκτηση αποτελεσμάτων από την ομοσπονδία πηγών δεδομένων πραγματοποιείται με την ακόλουθη διαδικασία:

1. *Επιλογή εννοιών*: Η εφαρμογή εξετάζει τις οντολογίες του συστήματος, διασχίζοντας τις ταξινομίες τους, για να εντοπίσει έννοιες ενδιαφέροντος.

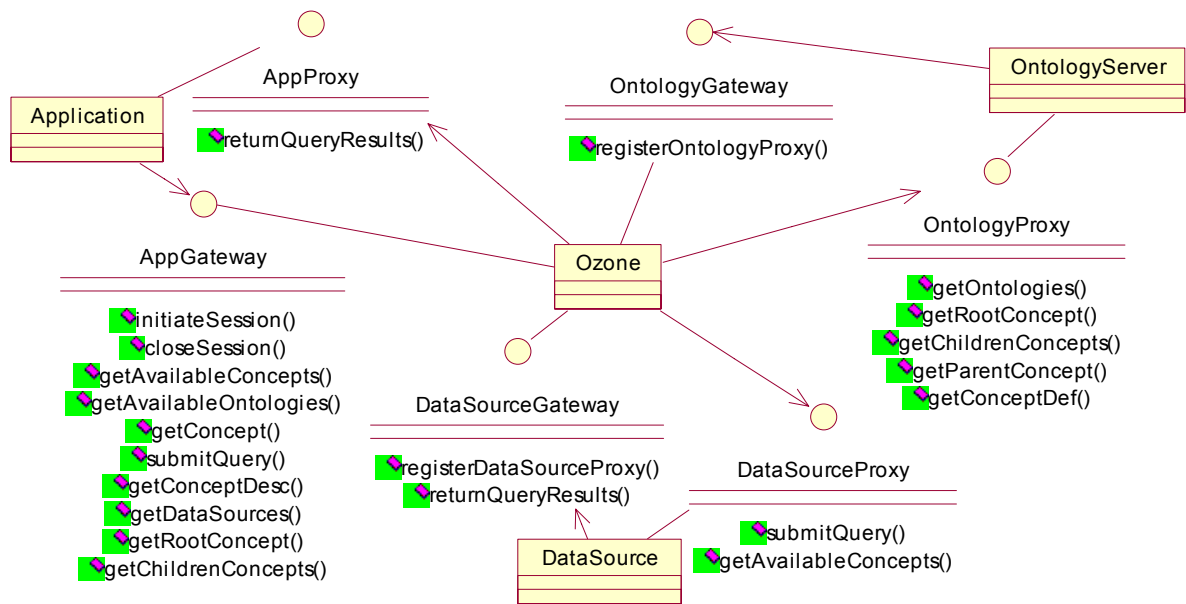
2. *Ανάκτηση εννοιών:* Όταν μια έννοια ενδιαφέροντος εντοπιστεί, μια λεπτομερής περιγραφή της έννοιας ανακτάται από την οντολογία. Αυτή η περιγραφή κωδικοποιείται με τη μορφή ενός αντικειμένου meta-class που αναπαριστά γνώση, όπως εξηγήθηκε στην παράγραφο 6.1.3.
3. *Διατύπωση ερωτημάτων:* Η εφαρμογή επεξεργάζεται το αντικείμενο meta-class προσθέτοντας περιορισμούς στις ιδιότητες του και στις ιδιότητες οποιωνδήποτε εμφωλιασμένων αντικειμένων περιέχει, που αντιπροσωπεύουν έννοιες συσχετισμένες με την αρχική. Έτσι το αντικείμενο μεταμορφώνεται από φορέα γνώσης σε φορέα του ερωτήματος.
4. *Επιλογή πηγών δεδομένων:* Το αντικείμενο ερωτήματος που προετοιμάστηκε παραπάνω στέλνεται στο Ozone. Το Ozone εξετάζει τον κατάλογο πηγών δεδομένων που είναι διαθέσιμες στην ομοσπονδία, για να εντοπίσει πηγές που διαθέτουν instances της έννοιας που αφορά το ερώτημα. Το ερώτημα στέλνεται σε όσες πηγές εντοπιστούν για να εκτελεστεί.
5. *Μετασχηματισμός ερωτήματος:* Το ερώτημα λαμβάνεται από το πρόγραμμα που υλοποιεί το Ozone interface κάθε πηγής δεδομένων. Το πρόγραμμα πραγματοποιεί το μετασχηματισμό του ερωτήματος από το αντικείμενο meta-class στην εγγενή αναπαράσταση της πηγής.
6. *Εκτέλεση ερωτήματος:* Το ερώτημα εκτελείται στην πηγή και τα αποτελέσματα επιστρέφονται στο interface, στην εγγενή αναπαράσταση δεδομένων της πηγής.
7. *Μετασχηματισμός αποτελεσμάτων:* Το interface λαμβάνει τα αποτελέσματα, τα μορφοποιεί κατάλληλα και τα μετασχηματίζει σε αντικείμενα meta-class. Τα αποτελέσματα στέλνονται στο Ozone σαν ένας πίνακας από τέτοια αντικείμενα.
8. *Συγκέντρωση αποτελεσμάτων:* Το Ozone λαμβάνει τα αποτελέσματα από όλες τις πηγές, τα συγκεντρώνει σε μία ενιαία δομή και τα επιστρέφει στην εφαρμογή. Αυτό το βήμα ολοκληρώνει τον κύκλο διαχείρισης ερωτημάτων.

Παρατηρούμε ότι η εμπλοκή της οντολογικής πλατφόρμας περιορίζεται στα βήματα 1 και 2. Συγκεκριμένα, χρησιμοποιείται μόνο για την ανάκτηση της απαραίτητης γνώσης για να διατυπωθεί το ερώτημα. Από τη στιγμή όμως της υποβολής του ερωτήματος και μετά, δεν έχει καμία συμμετοχή στη διαδικασία, η οποία πραγματοποιείται αποκλειστικά από τον πυρήνα του Ozone και τις πηγές.

Η απεμπλοκή της οντολογικής πλατφόρμας από τα κρίσιμα στάδια του μετασχηματισμού του ερωτήματος, της εκτέλεσης του και του μετασχηματισμού των αποτελεσμάτων είναι ιδιαίτερα σημαντική για την επεκτασιμότητα, την ικανότητα κλιμάκωσης (scalability), τις επιδόσεις και την αξιοπιστία του συστήματος. Στη βιβλιογραφία δεν υπάρχουν μελέτες επιτυχούς εφαρμογής οντολογιών σε πολύ μεγάλες βάσεις δεδομένων και θέματα όπως αξιοπιστία, υψηλή διαθεσιμότητα (high availability) και ανοχή σε λάθη (fault tolerance) δεν έχουν εξετασθεί. Το Ozone προσφέρει μια λύση που παρακάμπτει εντελώς αυτά τα προβλήματα.

6.3 ΤΥΠΟΠΟΙΗΣΗ ΤΩΝ INTERFACES

Τα τέσσερα βασικά συστατικά του συστήματος (πυρήνας Ozone, πηγές δεδομένων, οντολογίες και εφαρμογές) συνδέονται μέσω τριών interfaces που λειτουργούν προς αμφότερες κατευθύνσεις και συνδέουν τον πυρήνα με τα τρία περιφερειακά συστατικά. Εδώ δίνουμε τους ορισμούς τους με τη μορφή Java Interfaces. Έτσι κάθε interface του Ozone, όπως αυτά φαίνονται στο Σχήμα 19, ορίζεται με τη βοήθεια δύο Java Interfaces. Το πρώτο ορίζει τις μεθόδους που το Ozone υλοποιεί και το εξωτερικό σύστημα μπορεί να χρησιμοποιήσει. Αντίστροφα, το δεύτερο ορίζει τις μεθόδους που το εξωτερικό σύστημα πρέπει να υλοποιήσει προκειμένου να μπορέσει το Ozone να επικοινωνήσει μαζί του. Τα έξι Java Interfaces που προκύπτουν έτσι φαίνονται στο Σχήμα 20.



Σχήμα 20: Τα interfaces ανάμεσα στο Ozone και τα υπόλοιπα συστατικά του ολοκληρωμένου συστήματος σε UML

Το κάθε συστατικό του ολοκληρωμένου συστήματος υλοποιεί ένα «proxy» Interface και χρησιμοποιεί ένα «gateway» Interface του Ozone. Το ίδιο το Ozone υλοποιεί τα τρία gateway interfaces και χρησιμοποιεί λειτουργίες των περιφερειακών συστατικών μέσω των αντιστοιχων proxy interfaces. Η απομόνωση των συστατικών εξασφαλίζεται πλήρως έτσι, αφού το Ozone δεν χρειάζεται να γνωρίζει την εσωτερική δομή των άλλων συστατικών, και αυτά με τη σειρά τους μπορούν να αγνοήσουν την εσωτερική δομή του Ozone.

Η επικοινωνία του Ozone με τις οντολογικές πλατφόρμες εξασφαλίζεται μέσω του OntologyGateway και του OntologyProxy, που φαίνονται πάνω δεξιά στο σχήμα. Το πρώτο επιτρέπει την δήλωση διαθεσιμότητας μιας νέας οντολογίας στο σύστημα, μέσω μιας νέας ή μιας ήδη συνδεδεμένης οντολογικής πλατφόρμας. Η δήλωση πραγματοποιείται στέλνοντας στο Ozone το όνομα και μία σύντομη περιγραφή της οντολογίας, καθώς και μια αναφορά (reference) στο αντικείμενο που υλοποιεί τον OntologyProxy που θα πρέπει το Ozone να χρησιμοποιήσει για αποκτήσει πρόσβαση στη νέα οντολογία.

Στη συνέχεια το Ozone πρέπει να αποκτήσει κάποια εικόνα της ταξινόμιας που η νέα οντολογία περιέχει. Ανάλογα με τη στρατηγική επιδόσεων που εφαρμόζεται στο σύστημα, το Ozone μπορεί να παραμετροποιηθεί ώστε να ανασύρει και να αποθηκεύσει τοπικά την ταξινόμια αμέσως μόλις δηλωθεί μια νέα οντολογία, σε προγραμματισμένα χρονικά διαστήματα, όταν το σύστημα δεν έχει φόρτο, ή τέλος όταν κάποια εφαρμογή ζητήσει πρόσβαση σε αυτήν. Συνήθως είναι προτιμότερη η τοπική αποθήκευση της ταξινόμιας από την αρχή και η ανανέωση της σε τακτά χρονικά διαστήματα, αφού έτσι ελαχιστοποιείται ο χρόνος αναμονής της εφαρμογής όταν ζητήσει πρόσβαση στην οντολογία. Αντίθετα, η ανάκτηση αναλυτικών περιγραφών εννοιών συνήθως είναι καλύτερο να γίνεται κατά περίπτωση, όταν ζητηθεί. Η ανάκτηση της ταξινόμιας γίνεται πολύ εύκολα μέσω των μεθόδων `getRootConcept` και `getChildrenConcepts`, που προσφέρει η `OntologyProxy`. Η πρώτη επιστρέφει τη βάση της ταξινόμιας με τη μορφή ενός απλού αντικειμένου που περιέχει τον κωδικό της (GUID) και μια αλφαριθμητική περιγραφή, ενώ η δεύτερη επιστρέφει τις υποέννοιες της έννοιας που δίνεται σαν παράμετρος, με τη μορφή ενός πίνακα από τέτοια αντικείμενα. Η ταυτοποίηση κάθε έννοιας στο Ozone γίνεται με μονοσήμαντο τρόπο συνδυάζοντας τον κωδικό της οντολογίας στην οποία ανήκει με τον κωδικό της έννοιας. Η ανάκτηση της αναλυτικής περιγραφής μιας έννοιας γίνεται με τη μέθοδο `getConceptDesc` που επιστρέφει ένα αντικείμενο meta-class που έχει όλη την πληροφορία της έννοιας που δόθηκε σαν παράμετρος στη μέθοδο. Η λειτουργικότητα που παρέχει η `OntologyProxy` συμπληρώνεται από μια μέθοδο για την ανάκτηση όλων των οντολογιών που είναι διαθέσιμες στην τρέχουσα οντολογική πλατφόρμα.

Τα interfaces `DataSourceProxy` και `DataSourceGateway` επιτρέπουν την επικοινωνία ανάμεσα στο Ozone και τις πηγές δεδομένων (όχι μόνο τις βάσεις, όπως λανθασμένα υποδεικνύει η ονομασία τους, αλλά κάθε είδους πηγές). Με μινιμαλιστικό σχεδιασμό, μόνο τρεις μέθοδοι είναι απαραίτητες για να

ολοκληρωθεί αυτή η επικοινωνία. Η μέθοδος `registerDatabaseProxy` του `DataSourceGateway` λειτουργεί όπως η `OntologyGateway`: ειδοποιεί το σύστημα για τη διαθεσιμότητα μιας νέας πηγής δεδομένων. Οι άλλες δύο μέθοδοι χρησιμοποιούνται για την αποστολή ερωτημάτων και ανάκτηση των αποτελεσμάτων. Η `submitQuery` του `DataSourceProxy` παίρνει σαν παράμετρο ένα `meta-class` αντικείμενο που κωδικοποιεί ένα ερώτημα προς επεξεργασία. Αυτή η μέθοδος επιστρέφει μόλις η πηγή παραλάβει το ερώτημα. Όταν η πηγή ολοκληρώσει την εκτέλεση του ερωτήματος, χρησιμοποιεί τη μέθοδο `returnQueryResults` του `DataSourceGateway` για να στείλει τα αποτελέσματα στο Ozone. Η μέθοδος αυτή παίρνει ως παραμέτρους τον κωδικό του ερωτήματος και έναν πίνακα από αντικείμενα `meta-class` που περιέχει τα αποτελέσματα. Η χρήση μιας ξεχωριστής μεθόδου για την επιστροφή των αποτελεσμάτων είναι χρήσιμη γιατί η εκτέλεση των ερωτημάτων μπορεί να απαιτήσει μεγάλο χρονικό διάστημα, και η ασύγχρονη εκτέλεση της βοηθά το Ozone να διατηρήσει τον έλεγχο της διαδικασίας και να συντονίσει τα ερωτήματα που τρέχουν παράλληλα στις διαφορετικές πηγές δεδομένων. Η επιστροφή λαθών γίνεται με χρήση του μηχανισμού εξαιρέσεων (`exceptions`) της Java.

Το `interface` της εφαρμογής είναι, στην ουσία, η ένωση των δύο προηγούμενων `interfaces`. Μια εφαρμογή το χρησιμοποιεί για να αποκτήσει πρόσβαση στις πηγές δεδομένων και να εξετάσει τις οντολογίες. Έτσι οι κύριες λειτουργίες που μια εφαρμογή πρέπει να εκτελέσει είναι α) να λάβει καταλόγους των διαθέσιμων οντολογιών και πηγών δεδομένων στο σύστημα, β) να εξετάσει μια οντολογία λαμβάνοντας την ταξινόμια της αλλά και αναλυτική πληροφορία για έννοιες ενδιαφέροντος, γ) να θέσει ένα ερώτημα σε μία ή περισσότερες από τις διαθέσιμες πηγές δεδομένων και να παραλάβει τα αποτελέσματα.

Έτσι το `interface AppGateway` της εφαρμογής παρέχει μεθόδους που διακρίνονται στις εξής ομάδες:

- Διαχείριση επαφής (`initiateSession`, `closeSession`): Μέθοδοι για την έναρξη και τον τερματισμό μιας επαφής (`session`) με το Ozone, στη διάρκεια της οποίας η επικοινωνία των δύο συστημάτων είναι ασύγχρονη. Ο κωδικός της επαφής (`session id`) δίνεται ως παράμετρος σε όλες τις υπόλοιπες μεθόδους που περιγράφονται εδώ.
- Εξέταση οντολογιών (`getAvailableOntologies`, `getRootConcept`, `getChildrenConcepts`, `getConcept`): Μέθοδοι για τον εντοπισμό οντολογιών, την πλοήγηση στις ταξινομίες τους και την άντληση περιγραφών για τις έννοιες που περιέχουν.
- Εξέταση πηγών δεδομένων (`getDataSources`, `getAvailableConcepts`): Επιτρέπουν την επισκόπηση των πηγών δεδομένων που είναι συνδεδεμένες με το σύστημα. Επίσης, η μέθοδος `getAvailableConcepts` επιστρέφει μια λίστα με όλες τις έννοιες που έχουν `instances` στην πηγή δεδομένων που δίνεται ως παράμετρος, ή σε όλες τις πηγές του συστήματος αν η παράμετρος είναι κενή (`null`).
- Επεξεργασία ερωτημάτων (`submitQuery`): Θέτει ένα ερώτημα στο σύστημα για εκτέλεση. Δέχεται στις παραμέτρους της εκτός από τον κωδικό της επαφής το ερώτημα, σαν αντικείμενο `meta-class`, και προαιρετικά μια λίστα με πηγές δεδομένων στις οποίες αυτό θα πρέπει να απευθυνθεί. Το Ozone επιστρέφει με επιτυχία μόλις διανείμει το ερώτημα σε όλες τις πηγές. Η επιστροφή των αποτελεσμάτων γίνεται ασύγχρονα, με μια κλήση στην `returnQueryResults` που υλοποιεί η εφαρμογή και ορίζεται στο `AppProxy interface`. Η μέθοδος είναι σε πλήρη αντιστοιχία με αυτή που ορίζεται στο `DataSourceGateway`, ωστόσο διαφέρει στη

σημασιολογία της αφού εδώ επιστρέφονται τα συγκεντρωμένα αποτελέσματα από όλες τις πηγές που ερωτήθηκαν.

6.4 ΚΑΤΑΣΚΕΥΗ INTERFACES ΠΡΟΣ ΒΑΣΕΙΣ

ΔΕΔΟΜΕΝΩΝ

Σε αυτή την ενότητα εξηγούμε ότι είναι δυνατό να περιγραφεί η διαδικασία κατασκευής του interface προς σχεσιακές βάσεις δεδομένων σε μεγάλη λεπτομέρεια με αφηρημένο τρόπο, χωρίς δηλαδή αναφορά σε κάποια συγκεκριμένη στη βάση δεδομένων με την οποία επιχειρείται η σύνδεση. Δείχνουμε ότι αυτή η διαδικασία μπορεί να υποστηριχθεί από εργαλεία λογισμικού που την αυτοματοποιούν σε σημαντικό ποσοστό, με αποτέλεσμα την ελαχιστοποίηση της ανάγκης παραγωγής κώδικα για την παραγωγή του παραμετροποιημένου και προσαρμοσμένου στα πλαίσια της συγκεκριμένης βάσης δεδομένων interface προς το Ozone. Σε μια ομοσπονδία βάσεων δεδομένων η οποία αξιοποιείται από μία ή περισσότερες εφαρμογές, το έργο της προσθήκης βάσεων στο σύστημα είναι επαναληπτικό, αφού πρέπει να εκτελεστεί εκ νέου για κάθε βάση που προστίθεται στην ομοσπονδία. Ο χρόνος, ο κόπος και τελικά το κόστος αυτού του έργου είναι επομένως κρίσιμα για τη δυνατότητα μεγέθυνσης της ομοσπονδίας. Η διευκόλυνση αυτού του έργου είναι ζωτικής σημασίας για την επιτυχία της ομοσπονδίας. Η τυποποίηση του interface, που εξηγήθηκε στην ενότητα 6.3 είναι ένα ουσιώδες πρώτο βήμα προς αυτή την κατεύθυνση, αφού παγιώνει την προδιαγραφή του interface. Έτσι, το έργο ανάπτυξης του interface μπορεί να περιοριστεί στο σχεδιασμό και την υλοποίηση αποφεύγοντας τον κόπο αλλά και τους κινδύνους που εισέρχονται στην ανάλυση αναγκών.

Το επόμενο βήμα είναι να επιχειρήσουμε να κατατμήσουμε τη διαδικασία κατασκευής του interface σε αυτόνομες εργασίες με την ελάχιστη δυνατή αλληλεξάρτηση μεταξύ τους. Η αποσαφήνιση των ορίων και ρόλων του κάθε

τμήματος του ολοκληρωμένου συστήματος, που πραγματοποιήθηκε στην ενότητα 6.2.2, μας δίνει τη δυνατότητα να αντιμετωπίσουμε αυτή τη διαδικασία με αφηρημένους όρους. Ως προς τις πηγές δεδομένων, αυτό σημαίνει ότι μπορούμε να αγνοήσουμε τις τεχνικές διαφορές και ιδιομορφίες κάθε βάσης όταν εξετάζουμε την αλληλεπίδραση της στα πλαίσια του ολοκληρωμένου συστήματος της ομοσπονδίας. Με μόνη υπόθεση το ότι η βάση είναι σχεσιακή, μπορούμε να αναγνωρίσουμε τις ακόλουθες ομάδες εργασιών που προαπαιτούνται για την υλοποίηση του interface:

- Προσαρμογή της αναπαράστασης δεδομένων από το τοπικό σχήμα της βάσης στο κεντρικό του Ozone, που είναι βασισμένο στη meta-class.
- Αντιστοίχιση των οντολογικών εννοιών της ομοσπονδίας σε έννοιες του τοπικού σχήματος της βάσης.
- Μετασχηματισμός των ερωτημάτων από την κεντρική αναπαράσταση, που βασίζεται σε αντικείμενα meta-class, στην τοπική γλώσσα ερωτημάτων της βάσης.

Η αντιστοιχία μεταξύ του μοντέλου γνώσης του Ozone και του σχεσιακού μοντέλου καθιστά την πρώτη ομάδα εργασιών ευκολότερη. Η δεύτερη και η τρίτη είναι πιο περίπλοκες και είναι δυσκολότερο να αντιμετωπιστούν με γενικό τρόπο. Για τη δεύτερη ομάδα εργασιών τυποποιούμε μια διαδικασία που υποβοηθείται από εργαλεία λογισμικού και αντλεί από τους ειδικούς της βάσης την πληροφορία που χρειάζεται για να παραχθεί η αντιστοίχιση. Στη συνέχεια η αντιστοίχιση αυτή χρησιμοποιείται για να παραχθεί μια βιβλιοθήκη ερωτημάτων, η οποία σε κάθε οντολογική έννοια αντιστοιχίζει ένα παραμετροποιημένο ερώτημα σε SQL για χρήση μέσω JDBC, το οποίο μπορεί να χρησιμοποιηθεί για να ανακτηθούν instances της έννοιας. Για την τρίτη ομάδα εργασιών, δείχνουμε ότι είναι δυνατή η κατασκευή ενός γενικού προγράμματος το οποίο με

κατάλληλη παραμετροποίηση μπορεί να χρησιμοποιηθεί χωρίς την ανάγκη προγραμματισμού εκ νέου.

6.4.1 Η Διαδικασία Αντιστοίχισης

Η αντιστοιχία που είναι ο στόχος της δεύτερης ομάδας εργασίας μπορεί να παραχθεί ακολουθώντας μια διαδικασία της οποίας η δομή δεν εξαρτάται από τη συγκεκριμένη βάση δεδομένων και μπορεί να περιγραφεί γενικά. Η βασική ιδέα στην οποία στηρίζεται αυτή η διαδικασία είναι ότι μια οντολογική έννοια αντιστοιχίζεται σε μια οντότητα της βάσης, ένα υποσύνολο της, ή σε ένα συνδυασμό οντοτήτων. Για παράδειγμα, ας υποθέσουμε μια βάση οχημάτων όπου κάθε φέρμα αποτελεί μια ξεχωριστή οντότητα και φυλάσσεται σε ένα διαφορετικό πίνακα της βάσης. Τότε, η έννοια «car» της σχετικής οντολογίας θα αντιστοιχίζεται σε μια σειρά οντοτήτων της βάσης, για παράδειγμα «audi_car», «bmw_car», «toyota_car». Στην πράξη αυτό σημαίνει ότι όταν η εφαρμογή αναζητά instances του όρου car, το interface της βάσης θα πρέπει να τα αναζητήσει και στους τρεις πίνακες όπου φυλάσσονται οι παραπάνω οντότητες.

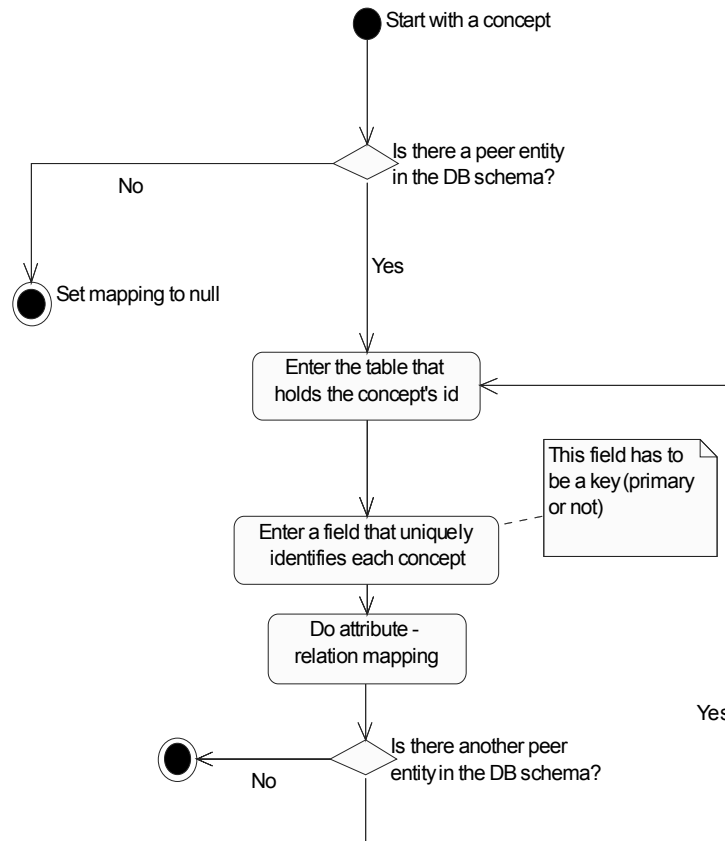
Το γεγονός ότι μία έννοια αντιστοιχίζεται σε μία οντότητα δεν σημαίνει όμως ότι όλη η πληροφορία που σχετίζεται οντολογικά με αυτή την έννοια περιέχεται στην συγκεκριμένη οντότητα. Αντίθετα, μπορεί να είναι κατακερματισμένη ανάμεσα σε διάφορες οντότητες της βάσης. Αυτό μπορεί να συμβαίνει καταρχήν επειδή το γνωστικό μοντέλο της οντολογίας είναι τελείως διαφορετικό στη σύλληψη του από το σχήμα της βάσης. Συχνότερα, αυτό μπορεί να είναι συνέπεια της αμφισημίας μεταξύ σχέσεων και ιδιοτήτων, που συζητήθηκε στα πλαίσια του μοντέλου γνώσης του Ozone στην παράγραφο 6.1.1. Συγκεκριμένα, είναι πολύ συχνό ένα χαρακτηριστικό της έννοιας που στην οντολογία κρίνεται ως περίπλοκο και μοντελοποιείται με μια σχέση προς μια δευτερεύουσα έννοια, στο σχήμα της βάσης να έχει σχεδιαστεί ως ένα απλό πεδίο. Το αντίστροφο είναι επίσης πιθανό αλλά λιγότερο συχνό, καθότι οι οντολογίες γενικά πραγματοποιούν διεξοδικότερη και αναλυτικότερη μοντελοποίηση των

γνωστικών περιοχών από τις βάσεις, οι οποίες περιορίζονται στα πλαίσια εξυπηρέτησης συγκεκριμένων αναγκών. Αντίστοιχα, μια σχέση στην οντολογία δεν αντικατοπτρίζεται απαραίτητα σε μια σχέση στη βάση. Αυτές οι περιπτώσεις είναι χειροπιαστές εκφράσεις του φαινομένου της ετερογένειας.

Για να αντιμετωπίσουμε την ετερογένεια, η διαδικασία αντιστοίχησης εξελίσσεται σε τρία βήματα. Στο πρώτο βήμα, μια οντολογική έννοια αντιστοιχίζεται σε μια ή περισσότερες οντότητες της βάσης. Στο δεύτερο βήμα αντιστοιχίζονται οι ιδιότητες της έννοιας, μία προς μία, σε πεδία της βάσης. Στο τρίτο βήμα αντιστοιχίζονται οι σχέσεις της έννοιας.

6.4.1.1 Αντιστοίχηση εννοιών

Στο πρώτο βήμα, μια οντολογική έννοια αντιστοιχίζεται σε μια ή περισσότερες οντότητες της βάσης. Κάθε μία από αυτές τις οντότητες πρέπει να έχει μια σχέση 1-1 ή 1-0 με την έννοια. Δηλαδή, ένα instance της έννοιας πρέπει να αντιστοιχίζεται σε ένα ή κανένα instance της οντότητας. Το πρώτο βήμα της διαδικασίας απεικονίζεται στο Σχήμα 21.



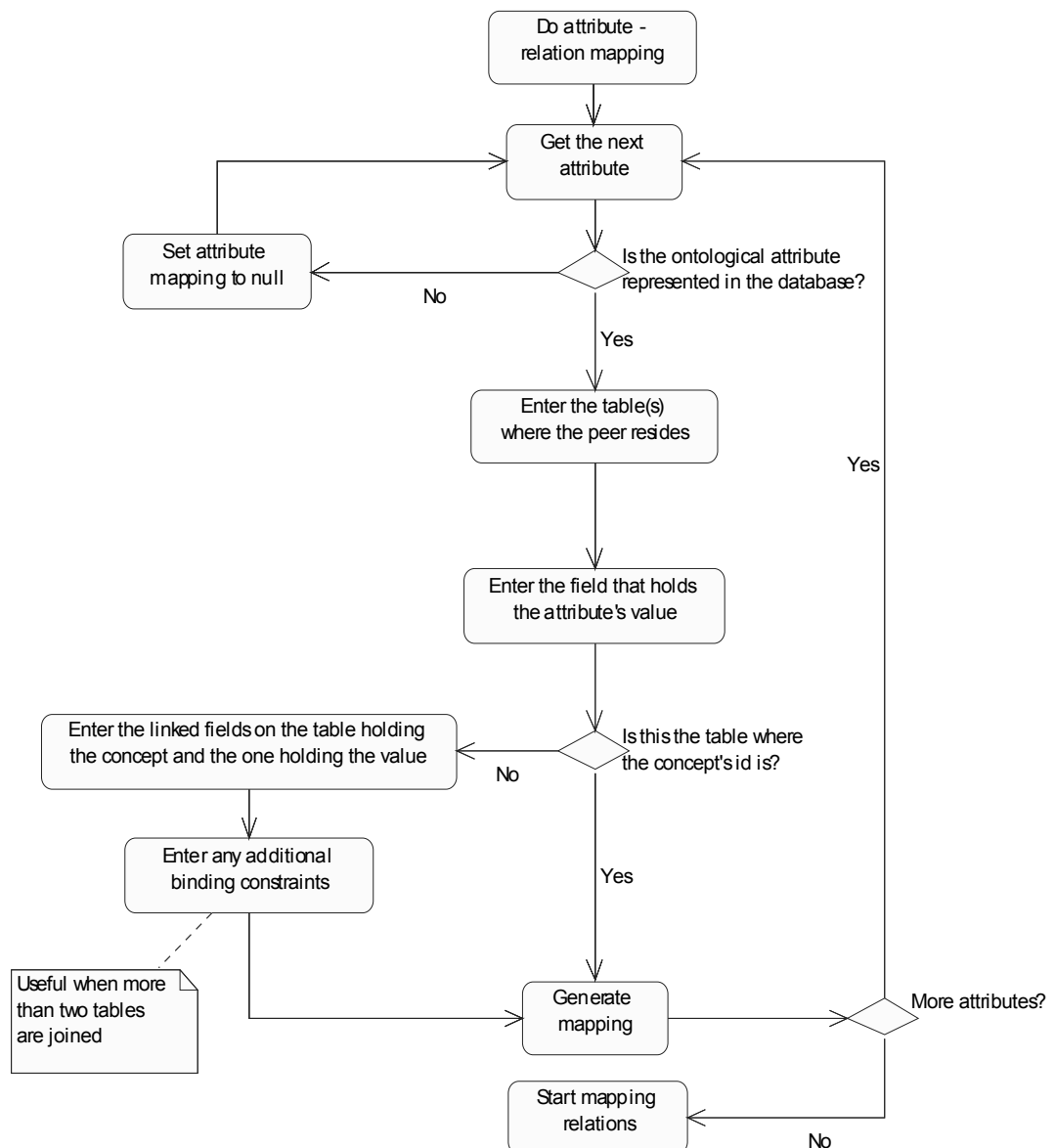
Σχήμα 21: Αντιστοίχιση μιας οντολογικής έννοιας σε οντότητες της τοπικής βάσης

Για κάθε οντότητα που μετέχει στην αντιστοίχιση, το όνομα του αντίστοιχου πίνακα καταγράφεται. Είναι σημαντικό να είναι γνωστό ένα κλειδί στον πίνακα αυτό, αν και όχι απαραίτητα το πρωτεύον κλειδί. Έτσι, το πεδίο που φιλοξενεί το κλειδί καταγράφεται επίσης. Για λόγους απλότητας αγνοούμε εδώ την περίπτωση που το κλειδί απαρτίζεται από πολλαπλά πεδία, όμως αυτή μπορεί να αντιμετωπιστεί επίσης, για παράδειγμα δημιουργώντας ένα υπολογιζόμενο πεδίο που συνθέτει ένα νέο κλειδί συνενώνοντας τα πολλαπλά πεδία του προηγούμενου κλειδιού.

Η διαδικασία συνεχίζεται με την αντιστοίχιση των ιδιοτήτων και των σχέσεων της έννοιας, η οποία περιγράφεται παρακάτω. Αν υπάρχουν και άλλες οντότητες που συμμετέχουν στην αντιστοίχιση, η διαδικασία επαναλαμβάνεται για κάθε μια από αυτές.

6.4.1.2 Αντιστοίχιση ιδιοτήτων

Η αντιστοίχιση ιδιοτήτων εξελίσσεται όπως φαίνεται στο Σχήμα 22. Το πρώτο γεγονός που πρέπει να ληφθεί υπόψη είναι ότι μια ιδιότητα μπορεί να μην υπάρχει καθόλου στη βάση, οπότε δεν μπορεί να γίνει αντιστοίχιση. Αν υπάρχει κάποιο σημασιολογικό αντίστοιχο της ιδιότητας, τότε αυτό μπορεί να είναι ένα πεδίο, μέρος ενός πεδίου (ή συνάρτηση ενός πεδίου γενικότερα), ή ένας συνδυασμός πεδίων.



Σχήμα 22: Αντιστοίχιση οντολογικών ιδιοτήτων στην τοπική βάση

Στην πρώτη περίπτωση, αν η ιδιότητα αντιστοιχεί σε πεδίο που βρίσκεται στον ίδιο πίνακα που έχει αντιστοιχιστεί η έννοια, τότε το όνομα του πεδίου αρκεί για να είναι δυνατή η εύρεση της τιμής του. Ωστόσο είναι πιθανό να χρειάζεται η ένωση (joining) ενός ή περισσότερων πινάκων για να βρεθεί αυτή η τιμή. Για να ενώσει κανείς τους πίνακες, χρειάζεται να ξέρει τα αντίστοιχα πεδία που πρέπει να συνδεθούν. Η διαδικασία προβλέπει το αντίστοιχο βήμα. Ωστόσο με αυτόν τον τρόπο δεν καλύπτονται όλες οι περιπτώσεις, αφού για παράδειγμα είναι δυνατό να χρειάζεται να συνδεθούν περισσότεροι του ενός πίνακες ή να προστεθεί κάποιος ειδικός περιορισμός στον πίνακα που συνδέεται. Αυτή η κατάσταση δεν μπορεί να αντιμετωπιστεί με αναλυτικό τρόπο, έτσι την αντιμετωπίζουμε προσφέροντας ένα επιπλέον πεδίο "binding constraints" όπου μπορούν να δοθούν οι τυχόν επιπλέον συνθήκες κατευθείαν σε SQL, όπως αυτό υποστηρίζεται από τη βάση.

Για παράδειγμα, ας υποθέσουμε πάλι την οντολογική έννοια car που έχει μια ιδιότητα make και μια συγκεκριμένη βάση όπου υπάρχει ο πίνακας car, που αντιστοιχεί στη συνώνυμη έννοια, ο πίνακας car_attribute όπου ορίζονται δυναμικά οι διάφορες ιδιότητες αυτοκινήτων όπως κατασκευαστής, μοντέλο, κυβισμός κλπ, και ο πίνακας car_attr_value που συνδέεται με πολλά-προς-ένα σχέσεις και με τους δύο προηγούμενους πίνακες και αποθηκεύει τις τιμές των ιδιοτήτων συγκεκριμένων αυτοκινήτων. Τότε, προκειμένου να ανακτήσει κανείς την τιμή της οντολογικής ιδιότητας car.make, θα πρέπει να συνθέσει ένα ερώτημα όπως το παρακάτω:

```
select value  
  
from car_attr_value, car_attribute  
  
where    car_id = 'X'  
  
        and car_attr_value.attr_id = car_attribute.attr_id  
  
        and car_attribute.name = 'make'
```

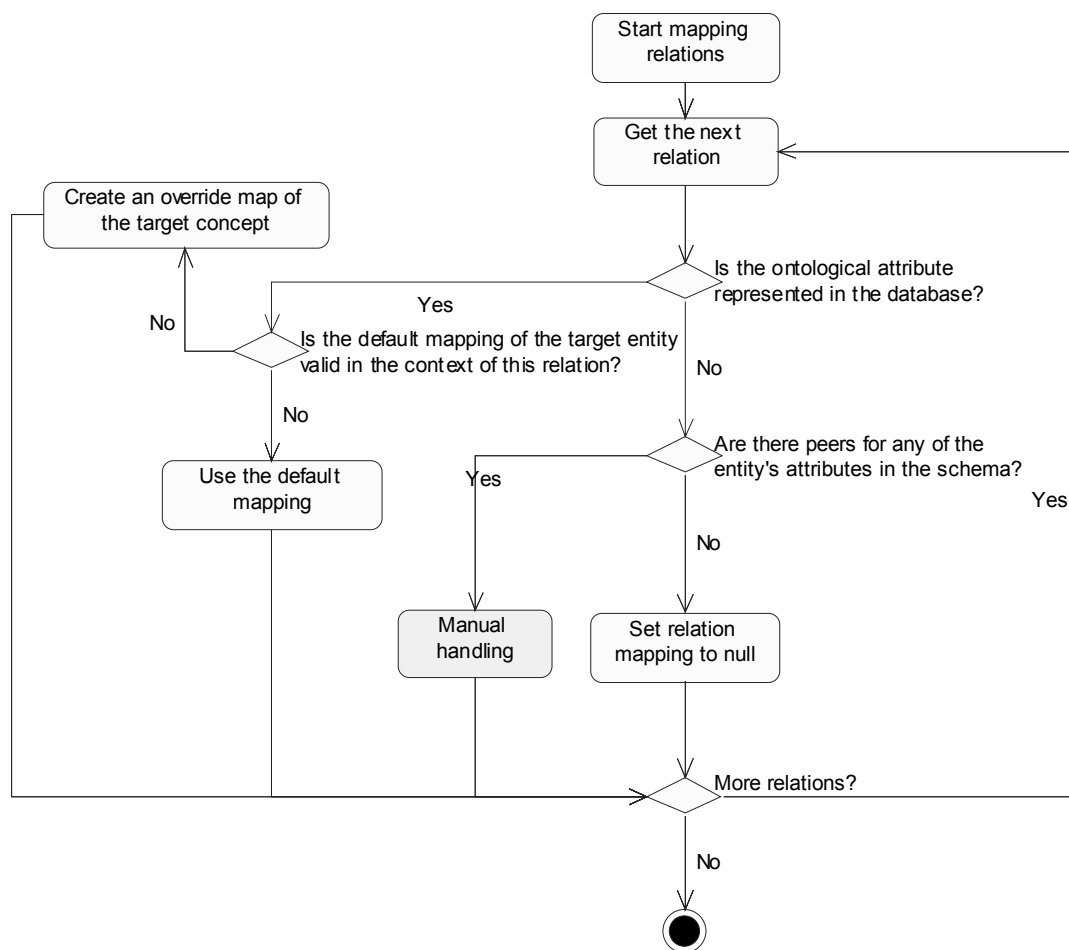
Στην περίπτωση αυτή, ο τελευταίος περιορισμός του ερωτήματος `car_attribute.name = 'make'` θα πρέπει να περαστεί σαν `binding constraint` αφού δεν υπάρχει άλλος τρόπος να συμπεριληφθεί στο πλαίσιο αντιστοίχησης.

Αν η ιδιότητα που προσπαθούμε να αντιστοιχήσουμε έχει πολλαπλότητα μεγαλύτερη της μονάδας, τότε αυτό το τμήμα της διαδικασίας επαναλαμβάνεται ώστε να οριστούν επιπλέον πεδία που αντιστοιχίζονται στην ιδιότητα. Για παράδειγμα, έστω μια έννοια `'contact info'` με ένα πεδίο `'contact phone number'`. Σε κάποια βάση, αυτή η ιδιότητα μπορεί να αντιστοιχίζεται σε τρία πεδία με ονόματα `'phone_no1'`, `'phone_no2'`, `'phone_no3'`. Και τα τρία αυτά πεδία θα πρέπει να αντιστοιχηθούν με την ιδιότητα.

Οι περιπτώσεις όπου μια ιδιότητα αντιστοιχίζεται σε τμήμα πεδίου, ή γενικότερα υπολογίζεται σαν συνάρτηση ενός πεδίου, δεν αντιμετωπίζονται εδώ. Τέτοιες περιπτώσεις προκύπτουν συνήθως όταν είτε η οντολογική ιδιότητα είτε το πεδίο της βάσης αντιπροσωπεύουν δομημένα αντικείμενα και όχι ατομικές τιμές. Τέτοιες περιπτώσεις είναι συνήθως αποτέλεσμα κακού σχεδιασμού, ή της ανάγκης περιορισμού πολυπλοκότητας και βελτιστοποίησης επιδόσεων σε βάρος της σχεδιαστικής τελειότητας. Σε αυτές τις περιπτώσεις θα πρέπει το πεδίο που καθορίζει την τιμή της ιδιότητας να αντιστοιχηθεί κανονικά με αυτήν, και στη συνέχεια να γίνει επέμβαση στον πίνακα αντιστοίχησης από το διαχειριστή της βάσης ώστε να δηλωθεί η συνάρτηση που δίνει την τιμή της ιδιότητας από την τιμή του πεδίου, κατά παράκαμψη της διαδικασίας που περιγράφεται εδώ. Η εξασφάλιση της δυνατότητας τέτοιας επέμβασης είναι βασική για την επιτυχία του εργαλείου αντιστοίχησης, αφού είναι αδύνατο να προβλεφθούν όλες οι περιπτώσεις όπου θα υπάρξει τέτοια ανάγκη. Επίσης η επέμβαση είναι αναγκαία και για λόγους βελτιστοποίησης των παραγόμενων ερωτημάτων (πχ εισαγωγή hints).

6.4.1.3 Αντιστοίχιση σχέσεων

Η αντιστοίχιση οντολογικών σχέσεων εξελίσσεται όπως φαίνεται στο Σχήμα 23. Οι οντολογικές σχέσεις συνδέουν έννοιες στην οντολογία. Η στρατηγική που εφαρμόζουμε εδώ δεν επιχειρεί την απευθείας απεικόνιση της οντολογικής σχέσης σε μια σχέση στη βάση. Αντί αυτού, προσπαθούμε να εκμεταλλευθούμε την απεικόνιση της έννοιας προορισμού της σχέσης για να εξάγουμε την αντιστοίχιση της.



Σχήμα 23: Αντιστοίχιση οντολογικών σχέσεων στην τοπική βάση

Υπάρχουν δύο προβλήματα που μπορούν να αντιμετωπιστούν με αυτή την προσέγγιση. Το πρώτο είναι ότι η αντιστοίχιση της έννοιας προορισμού εξαρτάται από τη σχέση. Για παράδειγμα, ας εξετάσουμε μια βάση στην περιοχή εφαρμογής των οχημάτων όπου φιλοξενούνται δύο πίνακες για ελαστικά

οχημάτων, ονομαζόμενοι `bike_tyres` και `car_tyres`. Ο πρώτος αποθηκεύει ελαστικά μοτοσικλετών, ενώ ο δεύτερος ελαστικά αυτοκινήτων. Όπως εξηγήθηκε παραπάνω, η οντολογική έννοια `tyre` που καλύπτει και τα δύο αυτά είδη θα αντιστοιχηθεί και με τους δυο πίνακες. Στα πλαίσια όμως της οντολογικής σχέσης `has(car, tyres)` μόνο ο δεύτερος πίνακας έχει νόημα. Για να αντιμετωπίσουμε αυτό το πρόβλημα, προσφέρουμε στα πλαίσια της διαδικασίας την δυνατότητα παράκαμψης της γενικής αντιστοίχισης της έννοιας προορισμού με μία ειδική, που έχει ισχύ μόνο στα πλαίσια της συγκεκριμένης σχέσης.

Το δεύτερο πρόβλημα παρουσιάζεται όταν η έννοια προορισμού της σχέσης δεν έχει αντίστοιχη οντότητα στη βάση, αλλά κάποιες από τις ιδιότητες της φιλοξενούνται στη βάση σαν πεδία άλλων οντοτήτων. Στο παράδειγμα παραπάνω, αυτό το πρόβλημα μπορεί να εμφανιστεί σε μία βάση που δεν διαθέτει καθόλου πίνακες για ελαστικά, αλλά ο πίνακας `car` περιέχει ένα πεδίο `tyre_size` που κωδικοποιεί τη διάσταση του ελαστικού του αυτοκινήτου. Στην οντολογία αυτό το πεδίο ανήκει στην οντότητα `tyre` που είναι συσχετισμένη με την έννοια `car`. Στις περισσότερες από αυτές τις περιπτώσεις είναι δυνατό να οριστεί μια ειδική αντιστοίχιση στα πλαίσια της σχέσης, η οποία να αντιστοιχίζει την έννοια που λείπει από τη βάση στην οντότητα που περιέχει μέρος της πληροφορίας της. Στο παράδειγμα, θα αντιστοιχήσουμε την έννοια `tyres` στα πλαίσια της σχέσης `has(car, tyres)` στον πίνακα `car`, και στη συνέχεια θα προχωρήσουμε στην αντιστοίχιση μόνο του πεδίου `tyre_size` στο αντίστοιχο πεδίο του πίνακα. Ωστόσο δεν είναι δυνατό να αποδείξουμε ότι αυτή η αντιμετώπιση μπορεί να χρησιμοποιηθεί σε όλες τις περιπτώσεις, επομένως αναγκαστικά υποθέτουμε ότι ενδέχεται σε κάποιες από αυτές να χρειαστεί επέμβαση από το διαχειριστή της βάσης.

Όπως και στην περίπτωση των ιδιοτήτων, η διαδικασία αντιστοίχισης της σχέσης μπορεί να επαναληφθεί για τις σχέσεις με πολλαπλότητα μεγαλύτερη της μονάδας.

6.4.2 Το Παραμετροποιήσιμο Interface Template

Η τυποποίηση των προδιαγραφών του interface και η αφηρημένη περιγραφή της διαδικασίας αντιστοίχησης καθιστούν δυνατή την επαναχρησιμοποίηση προγραμμάτων κατασκευασμένων ανεξάρτητα από την τοπική βάση που παραμετροποιούνται με βάση τα προϊόντα της διαδικασίας, μειώνοντας έτσι δραστικά την ανάγκη προγραμματισμού για την ολοκλήρωση της διασύνδεσης της βάσης στην ομοσπονδία. Ονομάζουμε αυτό το επαναχρησιμοποιήσιμο πρόγραμμα interface template. Αυτό είναι ανεπτυγμένο σε Java και δεν περιέχει καμία αναφορά σε συγκεκριμένη τοπική βάση. Για να υποστηρίξει μια συγκεκριμένη τοπική βάση, το πρόγραμμα παραμετροποιείται με τη βιβλιοθήκη ερωτημάτων που έχει παραχθεί από το εργαλείο αντιστοίχησης της προηγούμενης παραγράφου, και άλλες παραμέτρους που έχουν να κάνουν με τη σύνδεση με τη βάση μέσω JDBC. Προφανώς το μεγάλο μέρος του έργου παραμετροποίησης είναι η παραγωγή της αντιστοίχησης και η βελτιστοποίηση των παραγόμενων ερωτημάτων.

6.5 ΟΛΟΚΛΗΡΩΣΗ ΗΜΙ-ΔΟΜΗΜΕΝΗΣ

ΠΛΗΡΟΦΟΡΙΑΣ

Η ημι-δομημένη πληροφορία είναι μια έννοια που έχει αναδειχθεί παράλληλα με την εξέλιξη του WWW [Buneman, 1997]. Η ημι-δομημένη πληροφορία έχει κάποια δομή, η οποία όμως περιέχεται στην ίδια την πληροφορία. Δηλαδή, η ημι-δομημένη πληροφορία περιέχει τόσο τα δεδομένα όσο και το σχήμα τους σε ένα ενιαίο σύνολο. Σε κάποιες περιπτώσεις μπορεί να μην υπάρχει καθόλου εξωτερικός ορισμός του σχήματος, ενώ σε άλλες μπορεί να υπάρχει αλλά να μην θέτει αυστηρούς περιορισμούς στη μορφή των δεδομένων.

Το πλέον θεμελιώδες παράδειγμα ημι-δομημένης πληροφορίας είναι φυσικά ο παγκόσμιος ιστός. Ολόκληρος ο ιστός αλλά και τμήματα του αποτελούν πηγές

ημι-δομημένης πληροφορίας, αφού εμπίπτουν αμέσως στον ορισμό πιο πάνω. Έτσι είναι προφανής η σημασία της ημι-δομημένης πληροφορίας.

Η επεξεργασία ερωτημάτων και ο μετασχηματισμός ημι-δομημένων δεδομένων δεν είναι εφικτός με συμβατικά εργαλεία βάσεων δεδομένων. Σήμερα, η πλειοψηφία της έρευνας και των εφαρμογών ημι-δομημένης πληροφορίας βασίζονται στην XML [Bray, 1998]. Η XML είναι μια γλώσσα όπου τα δεδομένα περιγράφονται με ιεραρχικό τρόπο και το σχήμα τους εμφωλιάζεται σε αυτά. Έχουν αναπτυχθεί πρότυπα για την περιγραφή του σχήματος των XML δεδομένων, με βασικά το DTD και το XSD [Walmsley, 2004], και τον μετασχηματισμό τους, με κυριότερο το XSL – XSLT [Clark, 1999] [Grosso, 2001]. Για την επεξεργασία ερωτημάτων έχει προταθεί η γλώσσα XQuery [Boag, 2006] η οποία δεν είναι ακόμα καθιερωμένη ως πρότυπο, ωστόσο υπάρχουν κάποιες υλοποιήσεις ανάλογα με τη διαθέσιμη υποδομή.

Με τα εργαλεία αυτά και το αντικειμενοστραφές μοντέλο αναπαράστασης του Ozone, στο οποίο οποιαδήποτε ιεραρχική δομή είναι εύκολο να περιγραφεί, η ολοκλήρωση πηγών ημι-δομημένης πληροφορίας XML επιτυγχάνεται απλά. Ημι-δομημένες πηγές πληροφορίας άλλου είδους μπορούν επίσης να ολοκληρωθούν, αρκεί να κατασκευαστεί το αντίστοιχο interface κατά περίπτωση.

6.6 ΠΕΙΡΑΜΑΤΙΚΗ ΥΛΟΠΟΙΗΣΗ

Μια πειραματική υλοποίηση του Ozone πραγματοποιήθηκε σε Java [Stavroulas, 2002], και δοκιμάστηκε στα πλαίσια του ερευνητικού προγράμματος MKBEEM¹ [MKBEEM, 2006]. Σε αυτά τα πλαίσια σχεδιάστηκε,

¹ Το πρόγραμμα MKBEEM χρηματοδοτήθηκε εν μέρει από το πρόγραμμα IST της Ευρωπαϊκής Ένωσης, με το συμβόλαιο με αριθμό IST-1999-10589.

αναπτύχθηκε και δοκιμάστηκε σε βάθος μια πλατφόρμα ηλεκτρονικού εμπορίου, βασισμένη σε μια ομοσπονδία πηγών δεδομένων υποστηριζόμενη από έναν αριθμό οντολογιών. Στο σύστημα χρησιμοποιήθηκαν τρεις τελείως διαφορετικές πηγές δεδομένων, από τρεις διαφορετικούς παρόχους. Η πρώτη αφορούσε ταξιδιωτικές υπηρεσίες με έμφαση στα σιδηροδρομικά εισιτήρια, η δεύτερη είδη ένδυσης και υπόδησης και η τρίτη έπιπλα και είδη γραφείου. Η εφαρμογή έδινε δυνατότητες παρουσίασης και περιήγησης των καταλόγων προϊόντων και υπηρεσιών που ήταν διαθέσιμα στην πλατφόρμα, αναζήτησης προϊόντων και υπηρεσιών συγκεκριμένης κατηγορίας και παραμέτρων, παρουσίασης των αποτελεσμάτων στον πελάτη, επιλογής προϊόντων και υπηρεσιών, παραμετροποίησης τους όπου αυτό ήταν δυνατό (κυρίως στις ταξιδιωτικές υπηρεσίες) και υποβολής της αντίστοιχης παραγγελίας. Όλη η αλληλεπίδραση της εφαρμογής με τις οντολογίες και τις βάσεις δεδομένων των παρόχων, μέχρι το στάδιο της παραγγελιοληψίας, υποστηρίχθηκε από την πειραματική υλοποίηση του Ozone. Η παραγγελιοληψία υλοποιήθηκε με απευθείας συνδέσεις της εφαρμογής με τους παρόχους, αφού περιλάμβανε όχι μόνο την ανάγνωση αλλά και την εγγραφή και αλλαγή πληροφοριών στις αντίστοιχες βάσεις, λειτουργίες που είναι πέρα του πλαισίου του Ozone.

Το σύστημα ήταν χωρικά διεσπαρμένο κατά μήκος της Ευρώπης, με τις πηγές δεδομένων να βρίσκονται στην Αγγλία, τη Γαλλία και την Φινλανδία και το σύστημα ολοκλήρωσης και την εφαρμογή ηλεκτρονικού εμπορίου στην Ισπανία. Στην πειραματική υλοποίηση του Ozone και προκειμένου να υποστηριχθεί η πολύ βασική ανάγκη διασποράς του συστήματος σε ένα δίκτυο υπολογιστών τέτοιου μεγέθους, χρησιμοποιήθηκε τεχνολογία J2EE και τα συστατικά του συστήματος εμφωλιάστηκαν σε stateful ή stateless session enterprise Java beans. Αυτή η προσέγγιση αποδείχθηκε πολύ επιτυχημένη και οδήγησε σε ένα σύστημα με σταθερότητα, ευελιξία και καλές επιδόσεις. Αξίζει να σημειωθεί ότι στην πειραματική υλοποίηση η προσθήκη ή αφαίρεση υποσυστημάτων (πηγών

δεδομένων ή οντολογιών) από το ολοκληρωμένο σύστημα πραγματοποιείται στο χρόνο εκτέλεσης και δεν απαιτεί επανεκκίνηση του Ozone ή κάποιου από τα περιφερειακά συστατικά. Η δήλωση ενός νέου υποσυστήματος πραγματοποιείται στέλνοντας στο Ozone το όνομα και μία σύντομη περιγραφή της οντολογίας ή της πηγής, καθώς και τις J2EE παραμέτρους του proxy interface που θα πρέπει το Ozone να χρησιμοποιήσει για να αποκτήσει πρόσβαση στην οντολογία.

Η πλατφόρμα που κατασκευάστηκε στα πλαίσια του MKBEEM είναι σε θέση να υποστηρίξει τρεις διαφορετικές γλώσσες (αγγλικά, γαλλικά, φινλανδικά) και δίνει τη δυνατότητα αλληλεπίδρασης με το σύστημα μέσω ερωτημάτων διατυπωμένων σε φυσική γλώσσα. Έτσι οι χρήστες του συστήματος έχουν δύο επιλογές για να αναζητήσουν προϊόντα ή υπηρεσίες: μπορούν να θέσουν ένα ερώτημα σε φυσική γλώσσα, ή μπορούν να πλοηγηθούν μέσω των καταλόγων προϊόντων και υπηρεσιών που βρίσκονται κωδικοποιημένοι στις οντολογίες σαν ταξινομίες. Όταν φτάσουν στο προϊόν της επιλογής τους, το σύστημα εμφανίζει μια φόρμα που κατασκευάζεται δυναμικά με βάση την οντολογική περιγραφή του προϊόντος ή της υπηρεσίας που ανακτάται μέσω του Ozone. Οι χρήστες μπορούν να συμπληρώσουν στη φόρμα τις επιθυμητές τιμές του προϊόντος ή της υπηρεσίας που τους ενδιαφέρει και να θέσουν το αντίστοιχο ερώτημα στο σύστημα. Τόσο η κατασκευή της φόρμας όσο και η κωδικοποίηση του ερωτήματος με βάση τις συμπληρωμένες τιμές της πραγματοποιούνται με βάση αντικείμενα meta-class.

Η εξέλιξη της αλληλεπίδρασης με χρήση φυσικής γλώσσας εξελίσσεται διαφορετικά. Η διαχείριση του διαλόγου από την πλευρά του συστήματος γίνεται βάσει οντολογιών κατασκευασμένων για το σκοπό αυτό, και την υποστήριξη της μηχανής λογικής της οντολογικής πλατφόρμας. Ο διάλογος με το χρήστη εξελίσσεται με φυσική γλώσσα μέχρι το σημείο όπου το σύστημα έχει αποσαφηνίσει το προϊόν ή υπηρεσία που ενδιαφέρει το χρήστη. Μπορεί επίσης να έχει αντλήσει και τις επιθυμητές τιμές μιας ή περισσότερων ιδιοτήτων του,

στα πλαίσια του διαλόγου. Στο σημείο αυτό ο διάλογος ολοκληρώνεται και η ροή της διαδικασίας κατευθύνεται μέσω του Ozone. Αυτό πραγματοποιείται με μια κλήση στο Ozone για ανάκτηση της πληροφορίας για μια οντότητα (μέθοδος `AppGateway::getConcept`), η οποία δεν έχει στις παραμέτρους της τον συνήθη στατικό κωδικό της έννοιας, αλλά ένα δυναμικό κωδικό που αποτελεί έναν οντολογικό τύπο. Ο οντολογικός τύπος είναι το αποτέλεσμα του διαλόγου και προσδιορίζει το προϊόν που αναζητά ο χρήστης με όλους τους σχετικούς περιορισμούς στις παραμέτρους του. Η οντολογία στην οποία το Ozone δρομολογεί την κλήση αναλαμβάνει να «κατασκευάσει» το αντίστοιχο δυναμικό προϊόν. Στην ουσία, αυτό συνίσταται στο να επιστρέψει πίσω στο Ozone μια περιγραφή του προϊόντος σε ένα αντικείμενο `meta-class`, στο οποίο οι περιορισμοί που ο χρήστης έχει δηλώσει σε φυσική γλώσσα σε μια ιδιότητα απεικονίζονται στις μεταβλητές `constrainedRangeList` του αντίστοιχου αντικειμένου `ValueRange`. Όπως και στην περίπτωση της πλοήγησης καταλόγων, το αντικείμενο αυτό χρησιμοποιείται για τη δημιουργία και παρουσίαση στο χρήστη μιας φόρμας. Η διαφορά εδώ είναι ότι κάποιες παράμετροι της φόρμας είναι ήδη συμπληρωμένες, με βάση τον προηγούμενο διάλογο με το χρήστη. Ο χρήστης πλέον μπορεί να επισκοπήσει τη φόρμα, να διορθώσει τις συμπληρωμένες τιμές ή να συμπληρώσει επιπλέον περιορισμούς, και να την καταθέσει στο σύστημα για εκτέλεση.

Έτσι, ο ρόλος των οντολογιών στο MKBEEM είναι τριπλός. Πρώτον, χρησιμοποιούνται για την σημασιολογική ολοκλήρωση των πηγών δεδομένων. Δεύτερον, αποτελούν το σημασιολογικό λεξικό που χρειάζεται ώστε να ερμηνευθούν τα ερωτήματα φυσικής γλώσσας, και παρέχουν τις απαιτούμενες μηχανές λογικής. Και τρίτον, κωδικοποιούν το πολύπλοκο επιχειρηματικό μοντέλο και τους κανόνες λογικής που είναι απαραίτητοι για την δυναμική σύνθεση και διασύνδεση πακέτων ταξιδιωτικών υπηρεσιών, για την περιοχή του τουρισμού[Corcho, 2003]. Οι οντολογίες προσφέρουν τη δυνατότητα κάλυψης

και των τριών αυτών αναγκών με ένα μοναδικό σημασιολογικό μοντέλο της περιοχής εφαρμογής, αν και βέβαια το μοντέλο αυτό είναι ιδιαίτερα περίπλοκο.

Η πολυπλοκότητα και οι πολλαπλές λειτουργίες του συστήματος εξασφάλισαν βέλτιστες συνθήκες για τον πειραματισμό με το Ozone. Η ολοκλήρωση των πηγών δεδομένων πραγματοποιήθηκε με τον τρόπο που εξηγήθηκε παραπάνω, με την κατασκευή αντιστοιχιών οντολογίας-σχήματος και υλοποίηση από την πλευρά των οντολογιών του interface του Ozone.

Ένα από τα σημαντικότερα πειραματικά ευρήματα, ήταν η επαλήθευση της πολύ καλής υποστήριξης που προσφέρει το Ozone στην εφαρμογή, ιδιαίτερα στο κομμάτι της επαφής με το χρήστη – τον πελάτη του ηλεκτρονικού καταστήματος, στην περίπτωση αυτή. Για να κατασκευάσει η εφαρμογή τη φόρμα που χρησιμοποιεί ο χρήστης για την εισαγωγή των παραμέτρων του προϊόντος που επιθυμεί, ζητά από το Ozone ένα αντικείμενο meta-class που δίνει την οντολογική περιγραφή του προϊόντος σε αντικειμενοστραφή κωδικοποίηση. Η εφαρμογή εξάγει τα ονόματα, τα είδη τιμών και τους περιορισμούς των ιδιοτήτων από εκεί και απλά τα εμφανίζει στην οθόνη σαν πεδία σε μια φόρμα κατασκευασμένη με JSP. Οι σχέσεις οπτικοποιούνται σαν κουμπιά στην φόρμα τα οποία, όταν πατηθούν, ανοίγουν μια δεύτερη φόρμα βασισμένη στην οντολογική περιγραφή της έννοιας προορισμού της σχέσης. Μόλις η φόρμα συμπληρωθεί και κατατεθεί από το χρήστη, η εφαρμογή πρέπει μόνο να μεταφέρει τις συμπληρωμένες τιμές στις μεταβλητές `constrainedRangeList` των ιδιοτήτων και να στείλει το αντικείμενο πίσω στο Ozone για εκτέλεση του ερωτήματος. Έτσι όλη η αλληλεπίδραση βασίζεται πάνω σε μια συμπαγή δομή: την meta-class.

Η επίδραση αυτού του γεγονότος στην ανάπτυξη του Java τμήματος του συστήματος υπήρξε σαφής. Οι προγραμματιστές δεν χρειάστηκε σε καμία περίπτωση να διαχειριστούν οντολογικούς τύπους ή να μπουν στις λεπτομέρειες των οντολογικών μοντέλων. Αυτό ισχύει ακόμα και για την περίπτωση

ερωτημάτων σε φυσική γλώσσα. Όπως εξηγήθηκε, τα ερωτήματα φυσικής γλώσσας δρομολογούνται κατ' ευθείαν στο τμήμα της οντολογικής πλατφόρμας που κάνει την επεξεργασία φυσικής γλώσσας, το οποίο αναλαμβάνει τη διαχείριση του διαλόγου. Στο κομμάτι αυτό της διαδικασίας, η εφαρμογή απλά μεταφέρει ερωταπαντήσεις ανάμεσα στο χρήστη και το πρόγραμμα επεξεργασίας φυσικής γλώσσας. Με το πέρας του διαλόγου το τελευταίο δίνει στην εφαρμογή έναν οντολογικό τύπο, τον οποίο με τη σειρά της αυτή προωθεί στο Ozone όπως εξηγήθηκε παραπάνω, χωρίς να τον αναλύσει ή να τον κατανοεί.

Συνοψίζοντας, η πειραματική δοκιμή έδειξε ότι το απλουστευμένο μοντέλο γνώσης, ο αντικειμενοστραφής συμβολισμός και η δομημένη αρχιτεκτονική του ολοκληρωμένου συστήματος απέδωσαν την προσδοκώμενη μείωση στα κόστη και τους χρόνους ανάπτυξης του συστήματος. Το σύστημα υλοποιήθηκε σε δύο κύκλους, καθένας από τους οποίους είχε σαν αποτέλεσμα ένα πλήρως λειτουργικό πρωτότυπο. Ο κάθε κύκλος διήρκεσε περίπου ένα χρόνο και, το σημαντικότερο, απέδωσε ένα πειραματικό σύστημα που θεωρήθηκε πολύ επιτυχημένο στα πλαίσια της τυποποιημένης αξιολόγησης από τους χρήστες που ακολούθησε τη δοκιμή, ήδη από τον πρώτο κύκλο. Η εμπλοκή ειδικών στις οντολογίες υπήρξε απαραίτητη μόνο στην ανάπτυξη των ίδιων των οντολογιών και των προγραμμάτων διαχείρισης φυσικής γλώσσας. Τόσο ο πυρήνας του συστήματος ολοκλήρωσης όσο και η εφαρμογή αναπτύχθηκαν από μηχανικούς χωρίς καθόλου οντολογικές γνώσεις. Η ανάπτυξη των interfaces με τις πηγές δεδομένων έγινε από μεικτές ομάδες, σε χρόνους πολύ μικρότερους από αυτούς που είχαν αρχικά εκτιμηθεί στο χρονοδιάγραμμα του προγράμματος, με την υπόθεση ότι θα γίνονταν με βάση ένα μοντέλο βασισμένο περισσότερο σε οντολογίες. Έτσι μπορούμε να καταλήξουμε ότι η δοκιμή έδειξε ότι το Ozone πέτυχε να μειώσει την ανάγκη οντολογικών γνώσεων στην ομάδα ανάλυσης, σχεδιασμού και υλοποίησης του συστήματος και απλούστευσε σημαντικά την υλοποίηση των interfaces με τις πηγές δεδομένων.

7 ΑΣΦΑΛΗΣ ΑΝΑΖΗΤΗΣΗ ΗΜΙ-ΔΟΜΗΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΑΣ

Η εργασία στο Ozone οδήγησε σε ένα σύστημα που δίνει την δυνατότητα πρόσβασης, αναζήτησης και ανάκτησης δεδομένων από δομημένες και ημι-δομημένες πηγές πληροφορίας με ομογενή τρόπο. Για την αναζήτηση, το Ozone μεταφέρει τα κριτήρια του χρήστη στην πηγή και εναποθέτει στην πηγή το έργο της εκτέλεσης της αναζήτησης. Αυτός ο σχεδιασμός είναι προσανατολισμένος σε σχεσιακές βάσεις, που διαθέτουν ισχυρούς εγγενείς μηχανισμούς αναζήτησης. Ωστόσο, με τις ημι-δομημένες πηγές η κατάσταση είναι διαφορετική.

Στην περίπτωση της δομημένης πληροφορίας, το πρόβλημα της αναζήτησης είναι διεξοδικά αντιμετωπισμένο από όλες τις πλευρές. Αντίθετα, η αναζήτηση σε ημι-δομημένη ή αδόμητη πληροφορία είναι ένα πρόβλημα πιο πρόσφατο και αρκετά πιο περίπλοκο, και η διεξοδική αντιμετώπιση του δεν έχει ολοκληρωθεί. Ανάλογα με τις ιδιότητες της πληροφορίας και το είδος της εφαρμογής χρησιμοποιούνται τεχνικές αναζήτησης εντελώς διαφορετικές μεταξύ τους. Για παράδειγμα, στο διαδίκτυο έχουν αναπτυχθεί πολλές μηχανές αναζήτησης που εντοπίζουν έγγραφα με βάση λέξεις-κλειδιά ή γενικότερα βάσει του λεξιλογικού περιεχομένου του εγγράφου. Υπάρχουν ακόμα τεχνικές

αναζήτησης εικόνων μέσα σε έγγραφα, οι οποίες επίσης βασίζονται σε ανάλυση του κειμένου που περιβάλλει τις εικόνες, αλλά και τεχνικές αναζήτησης συγκεκριμένων προτύπων ή χαρακτηριστικών μέσα σε μία εικόνα. Οι τελευταίες είναι εντελώς διαφορετικές, και βασίζονται σε μαθηματική ανάλυση των χαρακτηριστικών της ίδιας της εικόνας. Έτσι, εξάρτηση του Ozone από τους εγγενείς μηχανισμούς αναζήτησης της πηγής δεδομένων είναι ένα σημαντικό μειονέκτημα στην περίπτωση των ημι-δομημένων πηγών.

Μια λογική προέκταση της δουλειάς στο Ozone είναι επομένως η ανάπτυξη μιας μεθόδου διαμεσολάβησης που να καλύπτει όχι μόνο τη σημασιολογία και την αναπαράσταση των δεδομένων, αλλά και τη μέθοδο αναζήτησης. Ακολουθώντας αυτή την κατεύθυνση και στοχεύοντας κυρίως σε πληροφορίες κειμένου ή πολυμέσων, κάναμε τη διαπίστωση ότι το πρόβλημα δεν είναι κρίσιμο όσο αφορά πληροφορίες που είναι κοινά διαθέσιμες. Στην περίπτωση αυτή υπάρχει ένας μεγάλος αριθμός μηχανών αναζήτησης διαθέσιμων στο internet που μπορούν να χρησιμοποιηθούν [Brin, 1998] [Google, 2006] [Sciencedirect, 2006]. Είναι πολύ πιο σημαντικό το πρόβλημα στην περίπτωση δεδομένων ιδιωτικής φύσεως, τα οποία δεν μπορούν να δημοσιευθούν ή πρέπει να αγοραστούν πρώτα.

Αυτό είναι το πλαίσιο της έρευνας που παρουσιάζεται σε αυτό το κεφάλαιο. Αναπτύξαμε ένα σύστημα που επιτρέπει τη διαμεσολάβηση ανάμεσα σε τελικούς χρήστες και παρόχους περιεχομένου, ώστε να μπορούν οι χρήστες να εκτελέσουν τους δικούς τους αλγορίθμους αναζήτησης πάνω στα δεδομένα των παρόχων και να λάβουν τα αποτελέσματα της αναζήτησης, χωρίς να έχουν τη δυνατότητα να υποκλέψουν τα δεδομένα. Ο κίνδυνος υποκλοπής δεν είναι ανύπαρκτος, είναι όμως ελεγχόμενος και εξηγούμε εκτενώς μεθόδους με τις οποίες μπορεί να περιοριστεί ή να εξαφανιστεί τελείως, ανάλογα με το είδος της εφαρμογής. Η δυνατότητα εφαρμογής του συστήματος σε υπαρκτές περιπτώσεις προβλημάτων εξετάζεται επίσης.

7.1 ΤΟ ΠΡΟΒΛΗΜΑ

Ως κίνητρο για την έρευνα που παρουσιάζεται εδώ, δίνουμε εδώ την περιγραφή μιας σημαντικής επιχειρησιακής περίπτωσης (business case) που αν και κοινή στον πραγματικό κόσμο απουσιάζει από το διαδίκτυο. Και στους δυο κόσμους μπορούμε να προσπελάσουμε δεδομένα δωρεάν. Ωστόσο για δεδομένα που δεν διατίθενται δωρεάν, στον ηλεκτρονικό κόσμο πρέπει κανείς να πληρώσει όλο το τίμημα τους των δεδομένων για να τα εξετάσει. Αυτό έρχεται σε αντίθεση με την δυνατότητα κάποιου να περιηγηθεί σε ένα βιβλιοπωλείο και να εξετάσει τα βιβλία που θέλει – σε μια τέτοια περίπτωση υπάρχει ελεύθερη αλλά προσωρινή πρόσβαση στα δεδομένα ενώ απαιτείται πληρωμή για μόνιμη πρόσβαση αφού ο πελάτης αποφασίσει ότι τα δεδομένα αυτά τον ενδιαφέρουν πραγματικά. Αυτή η δυνατότητα δεν παρέχεται στο διαδίκτυο. Οι πάροχοι πληροφορίας μπορούν να περιγράψουν τα δεδομένα τους, αλλά η εξέταση των ίδιων των δεδομένων, μέσω ακόμα και προσωρινής πρόσβασης, δεν επιτρέπεται για τον απλό λόγο ότι μπορεί κανείς εύκολα να κρατήσει ηλεκτρονικά αντίγραφα.

Το πλαίσιο PIVOTS (Private Information Viewing Offering Total Safety) [Yannopoulos, 2004] είναι μια προσπάθεια να διορθωθεί αυτή η έλλειψη του ηλεκτρονικού κόσμου. Το πρόβλημα περιγράφεται τεχνικά με τα ακόλουθα δεδομένα:

1. Ο πάροχος δεδομένων κατέχει δεδομένα που δεν πρέπει να διαρρεύσουν σε τρίτα μέρη αλλά πρέπει να είναι δυνατόν να χρησιμοποιηθούν για τον υπολογισμό διαφόρων μεγεθών που βοηθούν στην εκτίμηση τους από οποιονδήποτε το επιθυμεί.
2. Ο χρήστης επιθυμεί να αξιολογήσει τα δεδομένα με μέθοδο δικής του επιλογής (και άρα μη έμπιστη από τον πάροχο). Ο χρήστης ενδιαφέρεται να λάβει πίσω μόνο το αποτέλεσμα της αξιολόγησης.

3. Ο χρήστης θα πρέπει να μπορεί να στέλνει την μέθοδο της επιλογής του στον πάροχο και να μπορεί να λαμβάνει το αποτέλεσμα, χωρίς κίνδυνο έκθεσης των δεδομένων του παρόχου και χωρίς ανθρώπινη παρέμβαση από την πλευρά του παρόχου ή άλλου μέρους.

Η προτεινόμενη λύση βασίζεται σε ένα σχήμα Mobile Computing. Η ασφαλής εκτέλεση *μη έμπιστων* αλγορίθμων είναι επιθυμητή, ώστε να εμποδίζεται η παράνομη κτήση και επεξεργασία των δεδομένων από τον χρήστη. Ταυτόχρονα ο τύπος του επιστρεφόμενου αποτελέσματος είναι αυθαίρετα καθοριζόμενος από τον χρήστη και εξαρτάται αποκλειστικά από την εσωτερική σχεδίαση του αλγορίθμου. Ένας κακόβουλος χρήστης έχει πάντα την ευχέρεια να κωδικοποιήσει τμήμα των ιδιωτικών δεδομένων στην θεωρητικά έννομη έξοδό του, αλλά εξετάζουμε τρόπους εξάλειψης του εν λόγω κινδύνου. Η παρουσιαζόμενη αρχιτεκτονική δίνει την δυνατότητα στον χρήστη να στείλει τον εκτελέσιμο κώδικά του στον πάροχο όπου θα εκτελεσθεί σε ένα περιοριστικό περιβάλλον όπου απαγορεύεται να προσπελάσει οποιοδήποτε εξωτερικό πόρο χωρίς έλεγχο, και άρα να υποκλέψει τα ιδιωτικά δεδομένα. Η εφικτότητα της λύσης στηρίζεται στο γεγονός ότι χρήσιμα αποτελέσματα μπορούν να εξαχθούν με πολύ λιγότερα δεδομένα από αυτά που χρειάζονται για να κωδικοποιηθεί η πραγματική πληροφορία. Συνεπώς ο πάροχος επιβάλλει στον χρήστη αυστηρό περιορισμό στο εύρος ζώνης κωδικοποίησης του επιστρεφόμενου αποτελέσματος, κάτι όχι απόλυτα ασφαλές με την αυστηρή μαθηματική έννοια. Ωστόσο στις περισσότερες πρακτικές εφαρμογές το σύστημα μπορεί ναδειχθεί ότι είναι ασφαλές στην πράξη, αφού οι κακόβουλοι χρήστες είτε δεν επιτρέπεται να προσπελάσουν αρκετά δεδομένα ώστε να γίνει κλοπή σημαντικής πληροφορίας, είτε μπορούν να το κάνουν με κόστος μεγαλύτερο από την αξία της ίδιας της πληροφορίας, όπως εξηγούμε παρακάτω.

Το πρόβλημα αυτό μπορεί να αντιμετωπιστεί με λύσεις από το χώρο του Digital Rights Management (DRM) [Rosenblatt , 2001], [Samuelson, 1999],

[Kenny, 2002], [Kwok, 2000]. Για παράδειγμα, η προσωρινή περιήγηση στο περιεχόμενο ενός ηλεκτρονικού βιβλιοπωλείου μπορεί να επιτευχθεί μέσω DRM επιτρέποντας στους δυνητικούς πελάτες να έχουν δικαίωμα πρόσβασης περιορισμένου χρόνου στο ηλεκτρονικό περιεχόμενο που διατίθεται προς πώληση. Για τον λόγο αυτό παρακάτω θα επισημάνουμε τις διαφορές μεταξύ των εν λόγω προσεγγίσεων. Το DRM στηρίζεται σε μια πολύπλοκη τεχνολογική υποδομή (υλικό και λογισμικό), νομοθεσία, επιχειρηματικά μοντέλα, που θα πρέπει όλα τους να δουλεύουν σωστά ώστε πριν εφαρμοσθούν σε κάποιο εμπορικό σύστημα. Το προτεινόμενο σύστημα, εν αντιθέσει είναι ευκολότερο στην υλοποίηση, δεν εμπλέκει νομικά θέματα και έχει πιο σαφείς στόχους. Οι κίνδυνοι διαρροής δεδομένων σε συστήματα που χρησιμοποιούν το DRM είναι δύσκολο να αποσοβηθούν ή ποσοτικοποιηθούν με ακρίβεια, εξαιτίας της δυνατότητας των χρηστών να προσπελάσουν τα πλήρη δεδομένα, έστω και για μικρό χρονικό διάστημα. Το σύστημα δεν επιτρέπει ποτέ την άμεση προσπέλαση του χρήστη σε προστατευμένα δεδομένα και ο κίνδυνος κλοπής είναι καλά ορισμένος και μπορεί να μειωθεί στον επιθυμητό βαθμό, αλλάζοντας τις ρυθμίσεις του συστήματος. Αν το DRM φθάσει σε ένα επίπεδο ωριμότητας θα μπορούσε να δίνει την δυνατότητα προσωρινής προσωπικής περιήγησης στο προστατευόμενο περιεχόμενο, σε αντίθεση με το σύστημά μας, το οποίο όμως αποκλείοντας την ανθρώπινη πρόσβαση στα ιδιωτικά δεδομένα ενδείκνυται για περιπτώσεις όπου ο εν λόγω περιορισμός είναι μια απαίτηση, όπως η περίπτωση που παρουσιάζεται στο τμήμα 7.4.2. Συμπερασματικά, το πεδίο χρήσης του PIVOTS είναι εστιασμένο σε ένα υποσύνολο του DRM, είναι εύκολα και άμεσα αξιοποιήσιμο σε πραγματικές εφαρμογές σε αντίθεση με το DRM που απαιτεί μια μεγάλη νομική και τεχνική υποδομή πριν μπορέσει να αξιοποιηθεί ευρέως.

Στο τμήμα 7.2 περιγράφεται η αρχιτεκτονική του συστήματος. Δίνεται η δομή σε επίπεδο συστήματος και συζητούνται θέματα σχετικά με την υλοποίηση του συστήματος σε Java. Ιδιαίτερη έμφαση δίνεται στην χρήση του συστήματος

ασφαλείας της Java [Gong, 1999], [Lindholm, 2001], [Sun, 2006], και των πλατφορμών κινητών πρακτόρων [FIPA, 2000], [Wagner, 1998], [Nwana, 1999], [Wagner, 1999], [Jennings, 1998], [Huber, 2002]. Έτσι οδηγούμαστε σε μια περιγραφή χαμηλού επιπέδου, σε τεχνικούς όρους, που επιτρέπει την ενσωμάτωσή του συστήματος σε πρακτικές εφαρμογές.

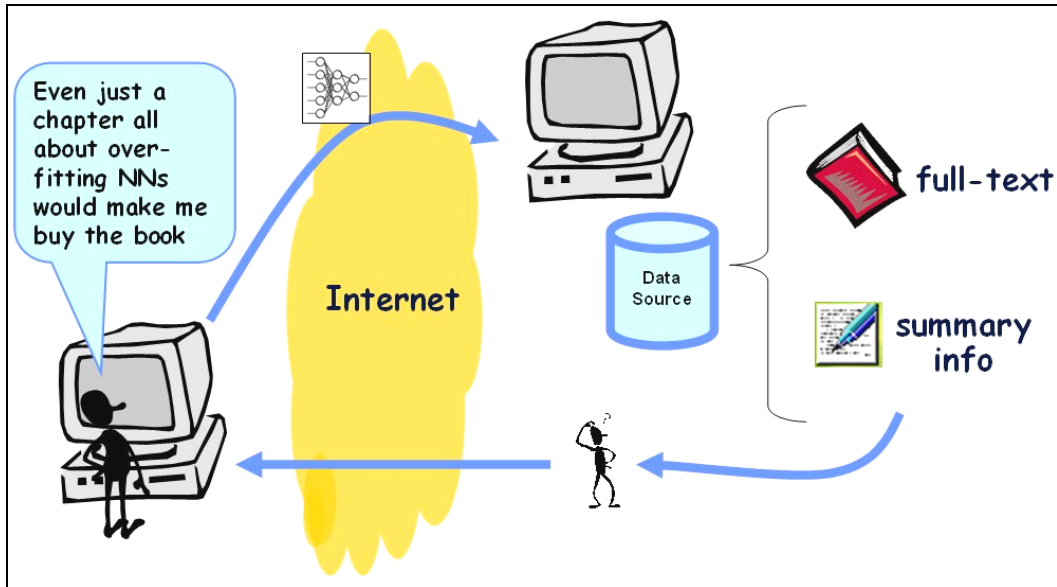
Στο τμήμα 7.3 γίνεται μια γενίκευση: ένα χρήσιμο επιχειρησιακό μοντέλο θα πρέπει να υπερβαίνει την υπόθεση ότι οι τελικοί χρήστες προγραμματίζουν τις δικές τους μεθόδους αξιολόγησης και αντί αυτού να υποστηρίζει την συνεργασία με μηχανές αναζήτησης παρεχόμενες από τρίτα μέρη. Έτσι παρουσιάζουμε μια αρχιτεκτονική που υποστηρίζει το εν λόγω μοντέλο και συζητούμε τα πλεονεκτήματα και τις πιθανές προκλήσεις που εγείρουν και τρόπους να αντιμετωπισθούν.

Το τμήμα 7.4 περιγράφει τρεις διαφορετικές στη φύση τους περιοχές εφαρμογής. Η πρώτη σχετίζεται με την υποστήριξη υποψήφιων αγοραστών βιβλίων ή άλλων μέσων να εντοπίσουν πραγματικά ενδιαφέρον υλικό. Αναπτύχθηκε στα πλαίσια του έργου MEMPHIS [MEMPHIS, 2000], [MEMPHIS, 2003], [Kasper, 2003]. Στο δεύτερο προτείνεται η βελτίωση της συνεργασίας μεταξύ οργανισμών ασφαλείας διεθνώς μέσω ελεγχόμενης διανομής δεδομένων και γίνεται χρήση εργαλείων επεξεργασίας εικόνας. Η τρίτη και τελική εφαρμογή βελτιώνει τη χρήση ιδιωτικών δεδομένων σε συστήματα μηχανικής μάθησης σε ιατρικές εφαρμογές επιτρέποντας την επικύρωση οποιωνδήποτε αναπτυγμένων αλγορίθμων με χρήση στοιχείων που, σήμερα, θα ήταν απολύτως μη διαθέσιμα στους ερευνητές.

7.1.1 ΠΑΡΑΔΕΙΓΜΑ ΕΦΑΡΜΟΓΗΣ

Εδώ εν συντομία περιγράφουμε μια βασική εφαρμογή του συστήματος ώστε να υποστηριχθούν οι τεχνικές περιγραφές των επόμενων δύο τμημάτων. Το

Σχήμα 24 οπτικοποιεί το πρόβλημα που πρόκειται να εξετάσουμε. Τα αλληλεπιδρώντα μέρη είναι εδώ ένας χρήστης που επιθυμεί να αγοράσει ένα βιβλίο και ένας προμηθευτής στοιχείων που, σε αυτήν την περίπτωση, είναι εκδότης που ενδιαφέρεται για την προώθηση και, ενδεχομένως, πώληση των βιβλίων του μέσω του Διαδικτύου. Σήμερα, αυτό γίνεται τυπικά με την δημοσίευση συνοπτικών πληροφοριών για κάθε βιβλίο σε κάποιο δικτυακό τόπο. Λογικά, αυτές οι συνοπτικές πληροφορίες δεν μπορούν να είναι εντελώς παραπλανητικές – πρέπει να δίνουν κάποια ιδέα του περιεχομένου κάθε βιβλίου , μαζί με άλλα στοιχεία όπως η τιμή και η ημερομηνία δημοσίευσης – αλλά γράφονται με στόχο περισσότερο τη διαφήμιση παρά μια αντικειμενική αξιολόγηση. Ακόμα και μια τίμια αξιολόγηση δεν θα πρόσφερε πολλά, καθώς κάθε χρήστης είναι πιθανό να έχει διαφορετικά ενδιαφέροντα και προτεραιότητες. Αυτό τυπικά εμποδίζει τις πωλήσεις μέσω Internet. Η φυσική επίσκεψη στο βιβλιοπωλείο έχει το τεράστιο πλεονέκτημα ότι πραγματικά αντίγραφα των βιβλίων μπορούν να ξεφυλλιστούν εκεί. Οι εκδότες έχουν σήμερα αναμφισβήτητα στην κατοχή τους ηλεκτρονικές εκδόσεις των βιβλίων τους, όμως αυτές είναι παραμένουν ζηλότυπα κρυμμένες από ενδεχόμενους αγοραστές.



Σχήμα 24: Σχηματική αναπαράσταση της βασικής εφαρμογής της μεθόδου

Στο Σχήμα 24, ο τελικός χρήστης είναι ικανός προγραμματιστής με πείρα στην επεξεργασία φυσικής γλώσσας. Θα ήθελε να μπορεί να στείλει στο δικτυακό τόπο του προμηθευτή δεδομένων ένα θεματικό ταξινομητή (topic classifier) – για παράδειγμα, έναν αλγόριθμο γλωσσικής επεξεργασίας βασισμένο σε νευρωνικό δίκτυο που εκπαιδεύεται σε προηγμένα γλωσσολογικά στοιχεία [Haykin, 1999] – ο οποίος να εκτελεστεί πάνω στο σύνολο του περιεχομένου κάθε βιβλίου. Το αποτέλεσμα, δηλαδή ένας απλός ακέραιος αριθμός που δείχνει τη σχετικότητα κάθε βιβλίου με το επιθυμητό θέμα, θα ήταν αρκετό για να προσδιορίσει ο χρήστης τι πρέπει να αγοραστεί. Αυτή η πληροφορία θα ήταν πολυτιμότερη από τη μεγαλύτερη αλλά αναξιόπιστη περίληψη που παρέχεται εξ ορισμού.

Πρέπει να σημειωθεί ότι ο προμηθευτής στοιχείων δεν έχει κανένα μέσο για να ελέγξει την αξιολόγηση που υπολογίζεται από το χρήστη. Στηριγμένοι σε αυτό το γεγονός, οι χρήστες θα μπορούσαν να εκμεταλλευτούν το χώρο του αποτελέσματος για διαφορετικούς σκοπούς. Παραδείγματος χάριν, ένας γλωσσολόγος να χρησιμοποιήσει την ευκαιρία που παρουσιάζεται προκειμένου

να υπολογίσει συχνότητες λέξεων ή γραμμάτων στις εργασίες των διάφορων συγγραφέων.

Πρέπει να τονιστεί ότι ο προμηθευτής στοιχείων πρέπει να είναι πρόθυμος να επιτρέψει στους χρήστες την ευκαιρία να εξαγάγουν οποιοσδήποτε αξιολογήσεις επιθυμούν από τα ιδιωτικά στοιχεία τους. Η μέθοδος που προτείνεται εδώ μπορεί να εφαρμοστεί μόνο σε περιπτώσεις όπου η προστασία των ίδιων των δεδομένων είναι επιθυμητή, αλλά και ικανή από μόνη της να εξασφαλίσει την ασφάλεια του συστήματος. Μια σύγκριση με το παραδοσιακό βιβλιοπωλείο μπορεί να είναι διαφωτιστική σε αυτό το σημείο: ο πελάτης που πηγαίνει σε ένα βιβλιοπωλείο με στόχο να ανακαλύψει ένα μικρό κομμάτι των πληροφοριών ξεφυλλίζοντας τα βιβλία μπορεί να αποφύγει να αγοράσει κάτι, αλλά γίνεται ανεκτός αφού μετά βίας θεωρείται απειλή για την εμπορική βιωσιμότητα του βιβλιοπωλείου. Η δυνατότητα που παρέχεται σε όλους τους άλλους αναγνώστες προκειμένου αυτοί να προσελκυστούν να αγοράσουν είναι πολύ περισσότερο σημαντική και αποτελεί ουσιαστικά τον πυρήνα του εμπορικού μοντέλου των βιβλιοπωλείων.

7.2 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΚΑΙ ΑΣΦΑΛΕΙΑ

Σε αυτό το τμήμα, εξηγούμε πώς οι προδιαγραφές, που συζητήθηκαν στην προηγούμενη εισαγωγή ως στόχος της έρευνας μας, μπορούν να καλυφθούν με λογισμικό. Η αρχιτεκτονική που παρουσιάζεται θα οδηγούσε σε τρύπες ασφάλειας εάν δεν υλοποιηθεί προσεκτικά, αλλά μια αναλυτική εξέταση αυτού του προβλήματος γίνεται σε επόμενη ενότητα, όπου περιγράφουμε τη δυνατότητα καθορισμού επιπέδων εξυπηρέτησης (service levels) ώστε να εξασφαλιστεί οποιοσδήποτε βαθμός ασφάλειας απαιτείται. Οι εκτιμήσεις ασφαλείας μας επικεντρώνονται εδώ στο να δείξουν ότι η υλοποίησή μας είναι πολύ αξιόπιστη υπό την έννοια ότι ακολουθεί ακριβώς την προδιαγραφή επιπέδου εξυπηρέτησης που έχει επιλεγεί.

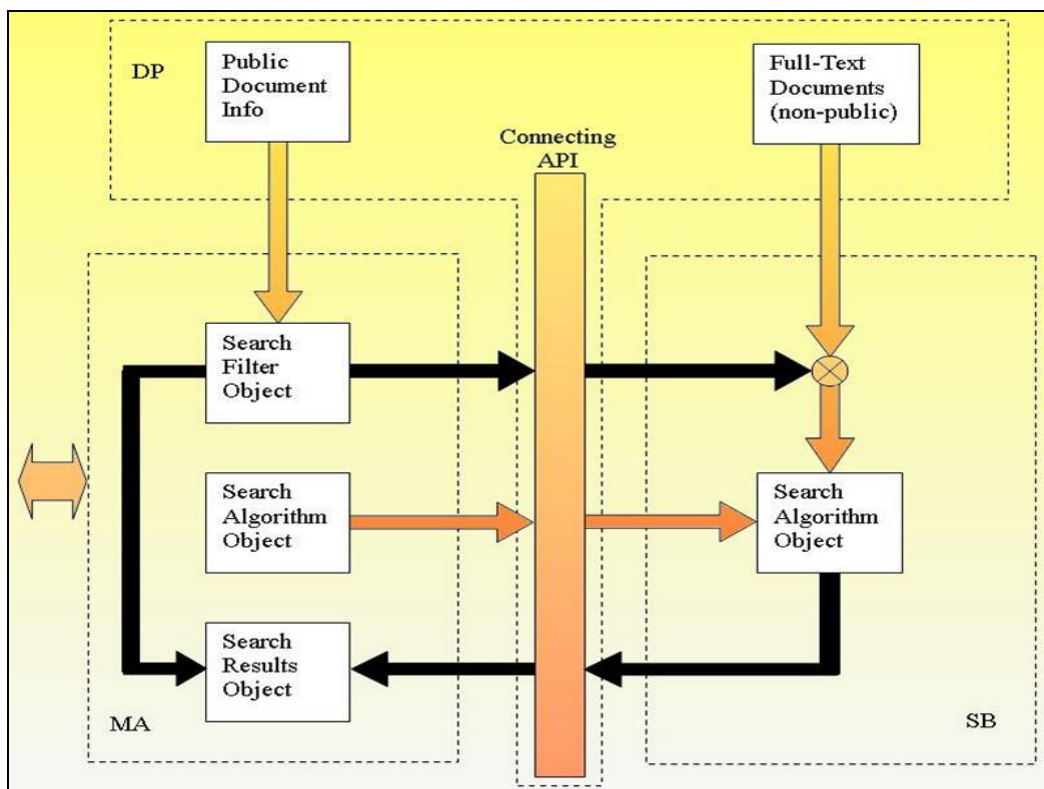
7.2.1 Αρχιτεκτονική Συστήματος

Για να περιγράψει το σχέδιο του συστήματος, είναι χρήσιμο να δοθούν μερικοί συμπαγείς ορισμοί για τους όρους που χρησιμοποιούμε σε αυτήν την έκθεση.

- **Έγγραφο:** Ουσιαστικά, είναι η μονάδα της προστατευόμενης πληροφορίας που προσφέρεται από τον προμηθευτή περιεχομένου. Στο παράδειγμα μας με τα βιβλιοπωλεία, κάθε βιβλίο που πωλείται χωριστά είναι ένα έγγραφο. Σε άλλες εφαρμογές, ένα έγγραφο μπορεί να είναι μια εγγραφή βάσεων δεδομένων, μια εικόνα, ένα XML έγγραφο, μια ιστοσελίδα κ.λπ.
- **Αλγόριθμος αναζήτησης:** Ο αλγόριθμος αναζήτησης είναι μια λειτουργία που εισάγει ένα ενιαίο έγγραφο και το αξιολογεί, σύμφωνα με τα δικά του κριτήρια. Ένας διαφορετικός αλγόριθμος αναζήτησης μπορεί να χρησιμοποιηθεί για κάθε έγγραφο.
- **Εκτίμηση εγγράφου:** Ένας πίνακας από bytes που είναι έξοδος από τον αλγόριθμο αναζήτησης. Κανονικά, η εκτίμηση εγγράφου μεταφράζεται, ανάλογα του αλγορίθμου, σε ένα μέτρο της σχετικότητας του εγγράφου με τα κριτήρια του αλγορίθμου. Εντούτοις, δεν υπάρχει κανένας έλεγχος για το τι περιέχει πραγματικά. Το μέγεθος του πίνακα είναι περιορισμένο από ένα οριζόμενο από την εφαρμογή μέγιστο μήκος.
- **Κοινές πληροφορίες εγγράφων:** Είναι το μικρό μέρος των πληροφοριών σε ένα έγγραφο που ο προμηθευτής στοιχείων επιλέγει να το δημοσιοποιήσει δωρεάν. Στο παράδειγμα μας με τα βιβλιοπωλεία αυτό θα ήταν ο τίτλος του βιβλίου, ο συντάκτης, το ISBN, και ίσως μια μικρή περίληψη ή μια κριτική. Το σύστημα χρησιμοποιεί αυτό το στοιχείο για σκοπούς βελτιστοποίησης της αναζήτησης. Επιτρέπεται να είναι κενό χωρίς επίδραση στη λειτουργία του συστήματος. Σε πολλές περιπτώσεις, η μορφοποίηση των εγγράφων θα είναι προφανής από την υλοποίηση – για την περίπτωση μας, έγγραφα απλού κειμένου παρουσιάζονται μόνο σαν

σειρές χαρακτήρων – ή από την ίδια την εφαρμογή – για παράδειγμα, στην περίπτωση εικόνων. Στα εμπορικά συστήματα όπου αυτό δεν ισχύει, πλήρη μεταδεδομένα (meta-data) που περιγράφουν το μορφότυπο του ιδιωτικού εγγράφου μπορούν να περιληφθούν σε αυτό το σημείο.

Η αρχιτεκτονική ενός συστήματος PIVOTS απεικονίζεται στο Σχήμα 25. Αυτή η αρχιτεκτονική εδρεύει στην πλευρά του προμηθευτή περιεχομένου και χρησιμοποιείται κατά περίπτωση από το χρήστη [Yannopoulos, 2002], [Yannopoulos, 2002a]. Τρία διαφορετικά λειτουργικά μπλοκ διακρίνονται στο σχήμα.



Σχήμα 25: Η αρχιτεκτονική του PIVOTS

Ο πυρήνας του συστήματος ονομάζεται Sandbox και είναι εκεί όπου ο προμηθευτής στοιχείων εκτελεί τον μη έμπιστο κώδικα του χρήστη με πρόσβαση στα ιδιωτικά στοιχεία. Το όνομα έχει επιλεγεί για να τονίσει το γεγονός ότι αυτός ο κώδικας, πακεταρισμένος σε ένα αντικείμενο αλγόριθμου αναζήτησης (Search Algorithm Object), δεν έχει πρόσβαση σε κανέναν άλλο πόρο. Η

εξασφάλιση της ευρωστίας του Sandbox είναι ένα κρίσιμο μέρος στο σχέδιο του συστήματος. Για να υπολογίσει την ταξινόμηση ενός εγγράφου, μόνο ο αλγόριθμος αναζήτησης και το ιδιαίτερο έγγραφο είναι τοποθετημένα στο Sandbox, και μόνο η εκτίμηση του εγγράφου λαμβάνεται έξω μετά από την ολοκλήρωση του υπολογισμού.

Το block του παρόχου δεδομένων (Data Provider ή DP) φιλοξενεί τις πληροφορίες των εγγράφων και διαχειρίζεται την υποβολή τους στα άλλα δύο συστατικά μέσω μιας κατάλληλης διεπαφής. Οι πληροφορίες του εγγράφου διαιρούνται σε δύο μέρη, ένα κοινό και ένα ιδιωτικό. Το κοινό μέρος ορίστηκε ανωτέρω, ενώ το ιδιωτικό μέρος περιέχει τα πλήρη έγγραφα. Το μπλοκ του παρόχου δεδομένων είναι το μόνο συστατικό του συστήματος που είναι ξεχωριστό στην περίπτωση κάθε παρόχου δεδομένων. Μια ειδική υλοποίηση της προδιαγραφής του απαιτείται για κάθε εφαρμογή. Αυτό δεν μπορεί να αποφευχθεί, δεδομένου ότι το μπλοκ του παρόχου δεδομένων πρέπει να μετατρέψει τα στοιχεία από το ιδιόκτητο σχήμα του DP στο τυποποιημένο σχήμα που χρησιμοποιείται από το σύστημα. Η διεπαφή σύνδεσης (Connecting API), που είναι ένα μέρος αυτού του συστατικού, δεν είναι απαραίτητο να υλοποιηθεί κατά περίπτωση. Εντούτοις, αποτελεί μέρος του μπλοκ του DP για δύο λόγους: Κατ' αρχάς, επειδή πρέπει να έχει πρόσβαση στις πηγές δεδομένων και, δεύτερο, επειδή είναι το κρίσιμο για την ασφάλεια συστατικό του συστήματος, αποτελώντας το φράγμα που απομονώνει το Sandbox και ελέγχει την πληροφορία που μπορεί αυτό να προσπελάσει. Για τον τελευταίο λόγο, το Connecting API πρέπει να δημοσιεύεται ως ανοιχτό λογισμικό πριν από την εμπορική εφαρμογή του συστήματος. Ο προμηθευτής στοιχείων μπορεί έπειτα να το επιθεωρήσει πριν το ενσωματώσει στο δικό του σύστημα. Αυτό θα επιτρέψει ευκολότερη υιοθέτηση του συστήματος από τους προμηθευτές στοιχείων.

Η πλατφόρμα κινητών πρακτόρων (Mobile Agent Platform) λειτουργεί ως διεπαφή με τον χρήστη, δέχεται τους αλγορίθμους αναζήτησης που αποστέλλονται από αυτόν. Η εκτέλεση τους διαιρείται σε δύο μέρη, αντίστοιχα με την διαίρεση των πληροφοριών των εγγράφων σε κοινό και ιδιωτικό μέρος. Το αντικείμενο φιλτραρίσματος αναζήτησης (Search Filter Object) ενεργεί ουσιαστικά ως ένας προεπεξεργαστής, που επεξεργάζεται τις πληροφορίες του κοινού τμήματος του κάθε έγγραφου και απορρίπτει αυτά που είναι εμφανώς μικρού ενδιαφέροντος. Αυτό το συστατικό δεν έχει πρόσβαση στις ιδιωτικές πληροφορίες. Το αντικείμενο αλγορίθμου αναζήτησης, που περνά στο Sandbox από την MA, λειτουργεί στο υποσύνολο των εγγράφων που περνούν από το φίλτρο αναζήτησης και μπορεί να επιστρέψει μόνο έναν ελεγχόμενο τύπο εξόδου. Στο σχήμα, αυτή η διαδικασία εμφανίζεται σαν ένα Search Filter Object που φιλτράρει τα ιδιωτικά στοιχεία βάσει των κοινών περιλήψεών τους. Είναι σημαντικό να γίνεται καλή χρήση του φίλτρου αναζήτησης για να περιοριστεί ο αριθμός ιδιωτικών εγγράφων που πρέπει να εξεταστούν – αυτό μπορεί να οδηγήσει στη σημαντική οικονομία πόρων.

7.2.2 Το Βασικό Σενάριο Χρήσης

Για να εκτελέσει μια αναζήτηση, ένας χρήστης στέλνει τον επιλεγμένο αλγόριθμο αναζήτησης και, προαιρετικά, ένα φίλτρο αναζήτησης στο δικτυακό τόπο του προμηθευτή. Το φίλτρο αναζήτησης μπορεί είτε να επιλέξει έγγραφα κατευθείαν χωρίς άλλη επεξεργασία, είτε να τα παραδώσει στον αλγόριθμο αναζήτησης για λεπτομερέστερη ανάλυση, είτε φυσικά να τα απορρίψει. Καθώς το φίλτρο λειτουργεί μόνο επί των δημόσιων πληροφοριών των εγγράφων όπως καθορίζεται ανωτέρω, καμία ειδική προφύλαξη δεν πρέπει να ληφθεί για την εκτέλεσή του. Όταν το φιλτράρισμα ολοκληρώνεται, το σύστημα έχει λάβει έναν κατάλογο από έγγραφα που πρέπει να αξιολογηθούν με τον αλγόριθμο αναζήτησης, εκείνα δηλαδή που έχουν παραπεμφθεί από το φίλτρο στον αλγόριθμο.

Η φάση ασφαλούς επεξεργασίας αρχίζει σε αυτό το σημείο. Ο κατάλογος από τα έγγραφα προς αξιολόγηση και ο αλγόριθμος αναζήτησης δίνονται στο Connecting API. Για να εκτιμηθεί κάθε έγγραφο το Connecting API δημιουργεί ένα διαφορετικό στιγμιότυπο (instance) του Sandbox. Το αντικείμενο αλγορίθμου αναζήτησης τοποθετείται στο Sandbox μαζί με το πλήρες έγγραφο, και επιτρέπεται να εκτελεστεί μέχρι και έναν μέγιστο χρόνο – το όριο αυτό είναι απαραίτητο για να υπερασπίσουμε το σύστημα ενάντια στις επιθέσεις άρνησης εξυπηρέτησης (denial of service). Μόλις η εκτέλεση ολοκληρώνεται, η μόνη επικοινωνία μεταξύ Sandbox και Connecting API είναι η επιστροφή των λίγων bytes της αξιολόγησης του εγγράφου.

7.2.3 Υλοποίηση

Υπάρχουν κάποιες βασικές τεχνολογικές επιλογές που πρέπει να γίνουν, προκειμένου να πραγματοποιηθεί το σχέδιο που παρουσιάζεται ανωτέρω. Οι δυο θεμελιώδεις παράγοντες συστήματος που διαμορφώνουν αυτές τις επιλογές είναι:

- **Sandboxing.** Η επιλεγμένη τεχνολογία εφαρμογής πρέπει να παρέχει μια ασφαλή μέθοδο δημιουργίας και διαχείρισης του Sandbox.
- **Μετανάστευση κώδικα.** Το σύστημα είναι βασισμένο στην εκτέλεση του αλγορίθμου αναζήτησης του χρήστη στην περιοχή του προμηθευτή περιεχομένου. Ο κώδικας του χρήστη πρέπει να μεταφερθεί σε αυτήν την περιοχή και να τοποθετηθεί στο sandbox χωρίς ανθρώπινη επέμβαση. Πρέπει να τονιστεί ότι αυτό δεν είναι μια περίπτωση απομακρυσμένης εκτέλεσης κώδικα (remote execution), η οποία θα μπορούσε ενδεχομένως να υποστηρίζεται από τεχνολογίες Remote Procedure Call (RPC). Στην περίπτωση της εκτέλεσης απομακρυσμένου κώδικα, ο κώδικας βρίσκεται και εκτελείται σε ένα απομακρυσμένο σύστημα και οι είσοδοι στέλνονται εκεί. Αντίθετα, στο σύστημα που εξετάζουμε ο απομακρυσμένος κώδικας

μεταφέρεται και εκτελείται τοπικά, ενώ τα δεδομένα δεν αφήνουν ποτέ την τοπική περιοχή.

Αυτοί οι παράγοντες μας οδήγησαν για να επιλέξουν την Java ως γλώσσα υλοποίησης του συστήματος. Η Java υποστηρίζει έναν επιμελημένο μηχανισμό για Sandboxing, που προορίζεται πρώτιστα για να υποστηρίξει την ασφαλή εκτέλεση των applets. Λόγω των διάδοσης της χρήσης τους, αυτός ο μηχανισμός είναι πολύ καλά δοκιμασμένος. Η Java υποστηρίζει πολιτικές ασφάλειας που καθορίζουν δικαιώματα πρόσβασης στα αρχεία, sockets, δίκτυο, ασφάλεια, διαδικασίες χρόνου εκτέλεσης, διαδικασίες Java reflection και serialisation [Gong, 1999], [Sun, 2001]. Για να δημιουργήσει κανείς ένα sandbox, πρέπει μόνο να φτιάξει ένα στιγμιότυπο της εικονικής μηχανής (virtual machine) της Java (JVM) με βάση μια πολιτική ασφάλειας που του αρνείται όλα τα δικαιώματα πρόσβασης στον εξωτερικό κόσμο. Οποιοσδήποτε κώδικας της Java που εκτελείται σε αυτό το JVM είναι εγγυημένο ότι δεν έχει πρόσβαση στους πόρους του συστήματος που προαναφερθήκαν.

7.2.3.1 Η πλατφόρμα πρακτόρων

Η java κάνει τη μετανάστευση κώδικα έναν σχετικά εύκολο έργο. Ο μεταγλωττιστής της Java αποθηκεύει το τον ενδιάμεσο κώδικα κάθε κλάσης, ονομαζόμενο bytecode, σε ένα χωριστό αρχείο. Το περιβάλλον χρόνου εκτέλεσης της Java χρησιμοποιεί αυτό το bytecode για να παράγει στο χρόνο της εκτέλεσης τον εκτελέσιμο κώδικα της κλάσης, με την εκκίνηση της JVM. Είναι επίσης δυνατό να φορτώσει κανείς μια κλάση και να την χειριστεί στο χρόνο εκτέλεσης, δηλαδή χωρίς η κλάση να είναι γνωστή στον χρόνο μεταγλώττισης του προγράμματος, με χρήση της βιβλιοθήκης Reflection της Java. Για το να κάνει κανείς αυτό, αρκεί το αρχείο bytecode της κλάσης να τοποθετηθεί σε μια προκαθορισμένη θέση στο σύστημα αρχείων (γνωστή ως classpath). Η κλάση φορτώνεται έπειτα με μια κλήση στη στατική μέθοδο `java.lang.Class.forName()`, η οποία παίρνει το όνομα της κλάσης ως παράμετρο.

Οποιοσδήποτε άλλες κλάσεις χρησιμοποιεί η κλάση που φορτώνεται, φορτώνεται επίσης αυτόματα.

Κατά συνέπεια, το πρόβλημα μετανάστευσης κώδικα μειώνεται στη μεταφορά των απαραίτητων αρχείων κατηγορίας από την περιοχή του χρήστη στη περιοχή του προμηθευτή δεδομένων, και την αποθήκευση τους στο σύστημα αρχείων. Η ίδια η μεταφορά αρχείων είναι τετριμμένη και μπορεί εύκολα να ολοκληρωθεί χρησιμοποιώντας διάφορα πρωτόκολλα όπως το FTP ή το HTTP, εντούτοις, η διαχείριση της διαδικασίας δεν είναι τετριμμένη. Για να απλοποιηθεί αυτό, αποφασίσαμε να χρησιμοποιήσουμε μια έτοιμη πλατφόρμα κινητών πρακτόρων ([FIPA, 2000], [Wagner, 1998], [Nwana, 1999], [Wagner, 1999], [Jennings, 1998], [Huber, 2002]). Οι πλατφόρμες κινητών πρακτόρων είναι κατασκευασμένες ακριβώς για να εξυπηρετήσουν τη μετανάστευση κώδικα μεταξύ υπολογιστών και την απομακρυσμένη εκτέλεση τους. Στην Java υλοποίηση μιας πειραματικής εφαρμογής του συστήματος, υιοθετήσαμε το Grasshopper [IKV, 2003] σαν την πλατφόρμα πρακτόρων μας. Το Grasshopper προσφέρει μια ποικιλία από επιλογές για τη μεταφορά κώδικα, συμπεριλαμβανομένου του TCP και του IIOP, με την επιλογή της SSL για την ασφάλεια μεταφοράς.

Το Grasshopper υλοποιεί τη λειτουργία του MA στο Σχήμα 25. Τα αιτήματα αναζήτησης φθάνουν στην πλατφόρμα με τη μορφή ενός κινητού πράκτορα που φέρει έναν αλγόριθμο αναζήτησης και προαιρετικά ένα φίλτρο αναζήτησης, όπου και τα δύο είναι κλάσεις Java και μεταφέρονται σε μορφή bytecode. Οι δυο κλάσεις απαιτούνται να υλοποιούν τα Java Interfaces SearchAlgorithm και SearchFilter αντίστοιχα, παρουσιαζόμενες στο Σχήμα 26. Τα τμήματα του συστήματος αναζήτησης χρησιμοποιούν αυτά τα Interfaces για να χειριστούν τις κλάσεις, χωρίς να χρειάζεται να γνωρίζουν την ακριβή υλοποίηση τους. Η χρησιμοποίηση Interfaces είναι καλύτερη από τη χρήση αφηρημένων κλάσεων,

δεδομένου ότι έτσι κανένας περιορισμός δεν επιβάλλεται στην ιεραρχία κλάσεων που υιοθετείται από το χρήστη.

```

interface DocumentSummary {
    public long getId();
    public String[] getKeywords();
    public String getTitle();
    public String getSubtitle();
    public String[] getTopics();
    public String getAbstract();
    public String getAuthor();
    public String getPublisher();
    public long getSize();
    public Date getDate();
}
interface FullDocument extends DocumentSummary {
    public String getFulltext();
}
// final, so it cannot be subclassed to cheat!
final class SearchResults {
    static final byte N = 1;
    public byte[] results = new byte[N];
    // it is so simple, and should be!
    // Let the user interpret meaning
    // of the bytes.
}
interface SearchFilter {
    // This method filters an array of
    // summaries, and returns an array
    // of ids of documents to be examined
    // in full within the sandbox
    public long[] filter(
        DocumentSummary[] summaries);
}
// classes implementing this interface
// are passed to the ConnectingAPI
interface SearchAlgorithm {
    public SearchResults assessFullDoc(
        FullDocument fullDoc);
}
// this specifies the Connecting API
// actual implementations are subclasses
abstract class AlgorithmSandbox
implements Runnable {
    // expected results size, for optimisation
    protected static final int initSize = 10;
    protected static DataProviderInterface
    dpInterface;
    public SearchResults[] performFullDocSearch(
        SearchAlgorithm algorithm,
        SearchFilter filter) {
        ...
    }
    public abstract SearchResults[]
    performFullDocSearch(
        SearchAlgorithm[] algorithms);
}
interface DataProviderInterface {
    public DocumentSummary[] getDocumentSummaries(
        SearchFilter theFilter);
    public FullDocument[] getFullDocuments(
        SearchAlgorithm theAlgorithm);
    // retrieves only docs specified in ids list
    public FullDocument[] getFullDocuments(
        SearchAlgorithm theAlgorithm,
        String[] docIds);
}

```

Σχήμα 26: Απλοποιημένος κώδικας Java των βασικών τμημάτων του συστήματος

Η πλατφόρμα πρακτόρων είναι η πύλη του PIVOTS προς τον κόσμο. Ένας στατικός πράκτορας, που καλούμε Reception, διαχειρίζεται την αλληλεπίδραση με τους εισερχόμενους κινητούς πράκτορες που φέρνουν τις αιτήσεις αναζήτησης. Ο κινητός πράκτορας παραδίδει τις δύο κλάσεις στην Reception, η οποία πραγματοποιεί το φιλτράρισμα όπως περιγράφεται ανωτέρω. Μέχρι αυτό το σημείο, κανένα προστατευόμενο δεδομένο δεν έχει χρησιμοποιηθεί, έτσι δεν υπάρχει καμία ανάγκη για περιορισμούς ασφάλειας.

7.2.3.2 Το Connecting API και η Διεπαφή του Προμηθευτή Περιεχομένου

Μόλις το φιλτράρισμα είναι ολοκληρωθεί, η Reception παραδίδει έναν κατάλογο με τους κωδικούς των εγγράφων προς εκτίμηση στο Connecting API, μαζί με ένα στιγμιότυπο του SearchAlgorithm. Το Connecting API καλεί την πηγή δεδομένων για να ανακτήσει αυτά τα έγγραφα, χρησιμοποιώντας το DataProviderInterface (Σχήμα 26). Αξίζει να αναφερθεί ότι το στιγμιότυπο του SearchAlgorithm περνά στη διεπαφή του προμηθευτή δεδομένων μαζί με το αίτημα, αν και αυτό μπορεί να φανεί περιττό σε πρώτη ανάγνωση. Αυτό γίνεται προκειμένου να επιτρέψει την χρήση συγκεκριμένων για κάθε προμηθευτή υποδομών αναζήτησης, όπως μια ιδιόκτητη μηχανή αναζήτησης. Αυτό δεν αποτελεί κενό στην ασφάλεια του συστήματος, καθώς ο αλγόριθμος αναζήτησης δεν εκτελείται σε αυτή τη φάση. Ο προμηθευτής δεδομένων έχει όμως τη δυνατότητα να επιθεωρήσει την κλάση του αλγορίθμου αναζήτησης και να χρησιμοποιήσει την πληροφορία αυτή για λόγους προ-επεξεργασίας. Έτσι οι προμηθευτές δεδομένων μπορούν να καθορίσουν δικές τους ιεραρχίες κλάσεων ή interfaces αλγορίθμων αναζήτησης. Σύμφωνα με τη θέση του αλγορίθμου αναζήτησης σε αυτήν την ιεραρχία, ο προμηθευτής στοιχείων μπορεί να αποφασίσει να φιλτράρει ορισμένα έγγραφα, μειώνοντας κατά συνέπεια την προσπάθεια για την εκτίμησή τους. Παραδείγματος χάριν, στο σενάριο με το βιβλιοπωλείο, χρησιμοποιήσαμε την ακόλουθη διεπαφή:


```
interface KeywordSearch {  
  
    public String[] getKeywords();  
  
}
```

Όταν ένας αλγόριθμος αναζήτησης υλοποιεί αυτήν την διεπαφή, ο προμηθευτής δεδομένων ξέρει ότι αναμένεται να τρέξει μια αναζήτηση βασισμένη σε λέξεις-κλειδιά στη βάση δεδομένων βιβλίων, και να επιστρέψει μόνο τα έγγραφα που περιέχονται στον κατάλογο που παρέχει το Connecting API, και περιέχουν επίσης όλες τις λέξεις κλειδιά που ορίζονται από τη διεπαφή KeywordSearch. Αυτό μπορεί δραστικά να μειώσει το ποσό εγγράφων που πρέπει να αξιολογηθούν. Η ίδια τεχνική χρησιμοποιείται κατά την ανάκτηση των περιλήψεων εγγράφων για φιλτράρισμα.

Η διεπαφή του προμηθευτή περιεχομένου πρέπει να υλοποιηθεί ξεχωριστά για κάθε προμηθευτή, για δύο λόγους. Κατ' αρχάς, πρέπει να έχει πρόσβαση στις πηγές στοιχείων του προμηθευτή χρησιμοποιώντας το ιδιόκτητο δίκτυο, λογισμικό, και υλικό του προμηθευτή. Δεύτερον, πρέπει να μετατρέψει τα ανακτημένα στοιχεία στο αναμενόμενο σχήμα από το PIVOTS. Κατά συνέπεια, ένας προμηθευτής είναι απαραίτητο να υλοποιήσει δύο παραπάνω Interfaces, τις DocumentSummary και FullDocument. Αυτές οι διεπαφές περιέχουν τις μεθόδους για την ανάκτηση των βασικών πληροφοριών εγγράφων που απαιτούνται από το PIVOTS και συνήθως επιστρέφουν απλά strings χαρακτήρων. Η πραγματική δομή των δεδομένων ή των πραγματικών κλάσεων που οι προμηθευτές δεδομένων χρησιμοποιούν για να υλοποιήσουν αυτές τις διεπαφές είναι διαφανείς στο σύστημα.

Επιστρέφοντας στο σενάριο αναζήτησης, τελικά η διεπαφή του προμηθευτή περιεχομένου παραδίδει όλο ή μέρος του ζητούμενου συνόλου από έγγραφα πίσω στο Connecting API, ανάλογα με το εάν έχει πραγματοποιηθεί προεπεξεργασία ή όχι. Για κάθε ένα από τα επιστρεφόμενα έγγραφα, το Connecting API δημιουργεί ένα στιγμιότυπο του Sandbox, τοποθετεί σε αυτό το

πλήρες έγγραφο και την κλάση του αλγορίθμου αναζήτησης , και εξάγει το αποτέλεσμα της αξιολόγησης μόλις η επεξεργασία τελειώσει.

Πρέπει να τονιστεί εδώ ότι, προκειμένου να διατηρηθεί σε πλήρη απομόνωση το Sandbox, είναι απαραίτητο κάθε στιγμιότυπο sandbox να εκτελείται ως χωριστή διαδικασία (process). Αυτό δημιουργεί μια ουσιαστική δυσχέρεια στις επιδόσεις και είναι ο κύριος λόγος που έχουμε λάβει διάφορα μέτρα για να ελαχιστοποιήσουμε τον αριθμό από έγγραφα που θα αξιολογηθούν. Εντούτοις, είναι αναπόφευκτο καθώς επαναχρησιμοποιώντας το ίδιο sandbox ή προσπαθώντας να δημιουργήσουμε ένα sandbox στο επίπεδο νημάτων (threads), υπάρχει σημαντικός κίνδυνος να επιτραπεί σε κακόβουλους χρήστες να αποκτήσουν μεγάλα μέρη ενός εγγράφου. Η γενική αδυναμία αυτών των αποδοτικότερων προσεγγίσεων είναι ότι ένας χρήστης μπορεί να αποθηκεύσει ένα συγκεκριμένο έγγραφο-στόχο μέσα στο Sandbox και κάθε φορά που το SearchAlgorithm εκτελείται για ένα διαφορετικό έγγραφο να το αγνοεί και να επιστρέφει ένα μερίδιο του στόχου ως αποτέλεσμά του. Αυτό θα ήταν σαφώς μια σοβαρή παραβίαση στις άμυνες του προμηθευτή περιεχομένου. Το μέγεθος του επιστρεφόμενου αποτελέσματος επιλέγεται έτσι ώστε είναι ασήμαντο έναντι του μεγέθους του συνολικού εγγράφου – εντούτοις, αυτή η αδυναμία θα μπορούσε να επιτρέψει σε έναν χρήστη να ψάξει n διαφορετικά έγγραφα και να πάρει σαν αποτέλεσμα ένα τμήμα ενός μόνο εγγράφου, με μέγεθος n φορές το μέγεθος του αποτελέσματος. Με μια αρκετά υψηλή τιμή του n , αυτό θα επέτρεπε σε έναν χρήστη να πάρει πίσω ένα πλήρες έγγραφο με ακριβώς μια αναζήτηση. Ακόμα κι αν διαφορετικά στιγμιότυπα του SearchAlgorithm χρησιμοποιηθούν για κάθε έγγραφο, ένας χρήστης θα ήταν σε θέση ακόμα να μεταφέρει δεδομένα μεταξύ των στιγμιότυπων με χρήση στατικών μεταβλητών-μελών. Αντίθετα από τις μεταβλητές στιγμιότυπου, οι στατικές μεταβλητές δεν διαγράφονται από τη μνήμη όταν καταστρέφεται ένα αντικείμενο, επειδή είναι κοινές σε ολόκληρη την κλάση παρά από ένα μόνο αντικείμενο αυτής. Ο μόνος τρόπος να

καταστραφούν οι στατικές μεταβλητές είναι να ξεφορτώσουμε την κλάση όπου ανήκουν από την JVM. Σύμφωνα με τις προδιαγραφές της JVM[Lindholm, 2001], καμία εγγύηση δεν γίνεται ως προς το πότε μια κλάση θα ξεφορτωθεί, ακόμα και όταν δεν υπάρχει καμία ισχύουσα αναφορά σε αυτή στο JVM. Η λειτουργία αυτή αφήνεται στην ευχέρεια της κάθε υλοποίησης της JVM. Συνεπώς, είναι αδύνατο να εγγυηθεί κανείς ότι μια στατική μεταβλητή θα καταστραφεί, ακριβώς με το να σιγουρευτεί ότι όλες οι ισχύουσες αναφορές στην κλάση καταστρέφονται.

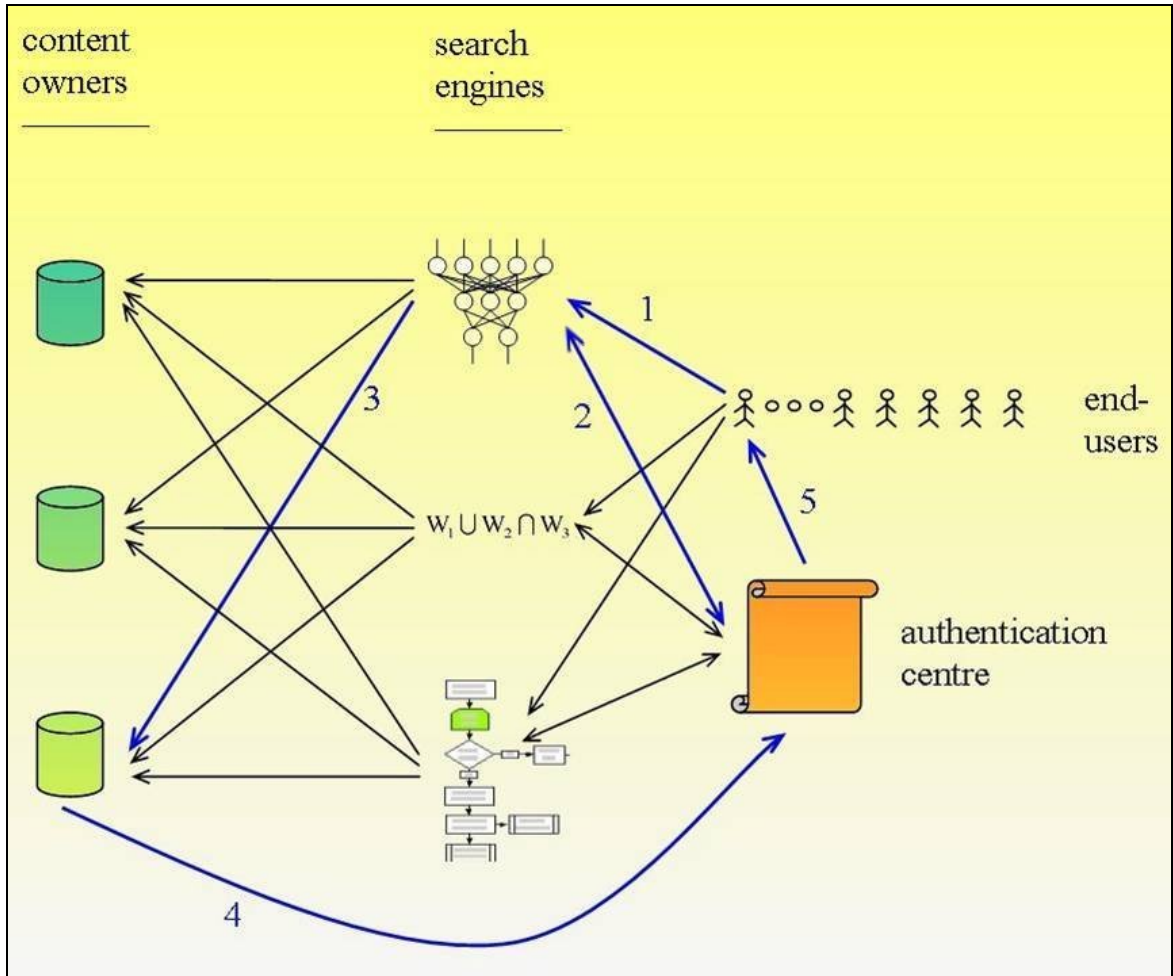
Κάποιος θα μπορούσε, φυσικά, να εφαρμόσει μια κατασκευασμένη επί τούτου υλοποίηση της JVM σύμφωνα με την προδιαγραφή της που είναι δημόσια διαθέσιμη, και να προσφέρει έναν μηχανισμό για να εγγυηθεί την εκφόρτωση των αχρησιμοποίητων κλάσεων σε συγκεκριμένα χρονικά σημεία. Αλλά, αυτό απαιτεί ουσιαστική προσπάθεια, και θα ήταν μια χρήσιμη λύση μόνο σε μεγάλης κλίμακας εφαρμογές του συστήματος. Τέλος, η επαναχρησιμοποίηση της ίδιας διαδικασίας (process) θα ήταν ακίνδυνα δυνατή εάν το JVM καταστραφεί και αναδημιουργηθεί μεταξύ εκτελέσεων του αλγορίθμου αναζήτησης. Ο κώδικας της java δεν θα είχε οποιοδήποτε τρόπο να αποθηκεύσει τις στατικές πληροφορίες έξω από JVM. Αν και η λειτουργία για να καταστραφεί ένα JVM συμπεριλαμβάνεται στην προδιαγραφή του, σαφώς δηλώνεται ότι δεν είναι υλοποιημένη μέχρι την έκδοση 1.4.1 του JVM, έτσι μόνο μια JVM μπορεί να δημιουργηθεί σε οποιαδήποτε διαδικασία σε όλη τη διάρκεια ζωής της. Εντούτοις, όταν αυτή η λειτουργία υλοποιηθεί, θα προσφέρει μια ουσιαστική ώθηση επιδόσεων. Ωστόσο αυτή τη στιγμή, η μόνη απολύτως ασφαλής μέθοδος είναι να εκτιμηθεί κάθε έγγραφο σε χωριστή υπο-διαδικασία (subprocess). Η εκκίνηση κάθε subprocess γίνεται με χρήση της οικογένειας Runtime.exec μεθόδων της Java. Η μετάδοση στοιχείων μεταξύ των subprocess διενεργείται μέσω του τερματικού και της κονσόλας τους, τα οποία ανακατευθύνονται στην

μητρική process και μπορούν να ανακτηθούν με τη μέθοδο `Process.getOutputStream ()` και τη `Process.getInputStream ()`.

7.3 ΕΦΑΡΜΟΓΗ ΤΗΣ ΜΕΘΟΔΟΥ ΩΣ ΑΣΦΑΛΗΣ

ΥΠΗΡΕΣΙΑ

Σε οποιαδήποτε πρακτικά εφαρμόσιμη περίπτωση, είναι αδικαιολόγητο να υποθέσουμε ότι κάθε χρήστης είναι ικανός προγραμματιστής με κατοχή μιας αποτελεσματικής, προσωπικά σχεδιασμένης μεθόδου αναζήτησης. Η επιχειρησιακή περίπτωση που είναι σημαντική να υποστηρίξουμε είναι αυτή που παρουσιάστηκε στο Σχήμα 27. Οι ιδιοκτήτες περιεχομένου, οι μηχανές αναζήτησης, και οι τελικοί χρήστες είναι οι κύριοι μέτοχοι εδώ. Όλοι οι ιδιοκτήτες περιεχομένου είναι προμηθευτές δεδομένων που φιλοξενούν συμβατές υλοποιήσεις του συστήματος PIVOTS. Κάθε μηχανή αναζήτησης παρέχει στους τελικούς χρήστες τη δική της ιδιαίτερα προηγμένη τεχνολογία αναζήτησης, που μπορεί να ικανοποιήσει λεπτομερείς και ιδιαίτερα συγκεκριμένες απαιτήσεις αναζήτησης. Συνεπώς οι μηχανές αναζήτησης θεωρούνται χρήστες από την σκοπιά των παρόχων περιεχομένου. Απαλλάσσουν τους τελικούς χρήστες από τον προγραμματισμό και τους αντιπροσωπεύουν τους παρόχους περιεχομένου. Οι πραγματικοί τελικοί χρήστες του συστήματος είναι έτσι έμμεσοι χρήστες από την σκοπιά των ιδιοκτητών του περιεχομένου.



Σχήμα 27: Σχηματική αναπαράσταση της λειτουργίας ασφαλούς υπηρεσίας αναζήτησης, βασισμένης στην προτεινόμενη μέθοδο

Από την άποψη του επιχειρησιακού μοντέλου που υποστηρίζεται, αυτό το σχέδιο είναι σαφώς πλεονεκτικότερο από την τρέχουσα κατάσταση όπου οι ιδιοκτήτες περιεχομένου σχεδόν πάντα προσφέρουν πρωτόγονες δυνατότητες αναζήτησης στις συλλογές δεδομένων τους. Κάτω από το προτεινόμενο σχέδιο, οι πάροχοι περιεχομένου δεν χρειάζεται να επενδύσουν στην ανάπτυξη μιας περίπλοκης υποδομής αναζήτησης. Οποιοσδήποτε τεχνικά ικανός διαμεσολαβητής μπορεί να προσφέρει προηγμένες λύσεις αναζήτησης και αξιολόγησης δεδομένων, χωρίς να χρειάζεται κάποια παραχώρηση ή ειδική συμφωνία αμοιβαίας εμπιστοσύνης ανάμεσα σε αυτόν και τον πάροχο. Οι τελικοί χρήστες μπορούν να επιλέξουν να κάνουν οποιοδήποτε συνδυασμό πηγής

δεδομένων προς διερεύνηση και μηχανής αναζήτησης. Φυσικά, οι τελικοί χρήστες μπορούν επίσης να στηριχθούν επάνω στα προγράμματά τους για να εκτελέσουν τις αναζητήσεις εάν το επιθυμούν. Επιστρέφοντας στο παράδειγμα βιβλιοπωλείων, μπορούμε τώρα να περιγράψουμε ένα ρεαλιστικό σενάριο. Ο τελικός χρήστης συνδέεται ηλεκτρονικά μέσω του WWW με μια μηχανή αναζήτησης και δίνει ένα σύνολο λεπτομερών απαιτήσεων. Σύντομα λαμβάνει πίσω τα αποτελέσματα της αναζήτησης του. Οι τεχνικές λεπτομέρειες της αναζήτησης, αν και άγνωστες στον τελικό χρήστη, είναι οι ίδιες με πριν. Οι τεχνικές λεπτομέρειες του αλγορίθμου αναζήτησης είναι επίσης κρυμμένες από τον πάροχο περιεχομένου.

Σε επόμενη παράγραφο θα περιγράψουμε ενδιαφέρουσες και ποικιλόμορφες περιοχές εφαρμογής που προκύπτουν από παραλλαγές αυτού του γενικά εφαρμόσιμου επιχειρησιακού μοντέλου.

Έχουμε μέχρι τώρα δώσει μια αρκετά ακριβή προδιαγραφή του συστήματος – λεπτομέρειες στο επίπεδο αρχιτεκτονικής του βασικού συστήματος και τη μέθοδο να γενικευθεί σε υπηρεσίες του πραγματικού κόσμου – και μπορούμε να εξετάσουμε διεξοδικά το ζήτημα της ασφάλειας. Έχοντας τη δυνατότητα να ορίσει τις παραμέτρους αναζήτησης για κάθε αναζήτηση, ένας κακόβουλος χρήστης θα μπορούσε να δημιουργήσει έναν αλγόριθμο για να ανακτήσει διαδοχικά μικρά τμήματα των ιδιωτικών δεδομένων, που κωδικοποιούνται ως το αποτέλεσμα αναζήτησης που υποτίθεται ότι είναι μια αξιολόγηση της σχετικότητας με τα κριτήρια. Αυτό το πρόβλημα δεν μπορεί να αποφευχθεί εντελώς, αλλά μπορεί να συρρικνωθεί έτσι ώστε να γίνει ασήμαντο. Τα ακόλουθα τμήματα περιγράφουν ποικίλες μεθόδους για να προστατευθούν τα ιδιωτικά δεδομένα. Κάθε μια από αυτές μπορεί από μόνη της να επιτύχει απόλυτη ασφάλεια, εάν είναι κατάλληλη για μια συγκεκριμένη εφαρμογή. Για μερικές εφαρμογές, μπορεί να είναι καλύτερο να χρησιμοποιηθεί ένας συνδυασμός μεθόδων ώστε να ατονήσουν κάποιοι από τους περιορισμούς τους,

ή ακόμα για να επιτύχουμε μεγαλύτερη ασφάλεια λόγω της οργάνωσης σε αλληπάλληλα και επικαλυπτόμενα συστήματα ασφαλείας.

7.3.1 Θέματα Ασφάλειας

Μια απλή προστασία όταν υπάρχει ένας μόνο χρήστης είναι να περιορίσουμε τον αριθμό αναζητήσεων και αξιολογήσεων που πάνω σε κάθε ιδιωτικό έγγραφο. Κατά συνέπεια, μόνο μερικά bytes μπορούν να κλαπούν στο σύνολο. Υπάρχουν, εντούτοις, δύο σημαντικότεροι τύποι επιθέσεων που πρέπει να εξεταστούν στην πράξη.

Στην πρώτη περίπτωση, κάποιοι τελικοί χρήστες δημιουργούν μια κακόβουλη μηχανή αναζήτησης και συνεργάζονται στη χρησιμοποίησή της για να υποκλέψουν περιεχόμενο. Το ίδιο αποτέλεσμα θα μπορούσε να επιτευχθεί από ένα μεμονωμένο πρόσωπο με χρησιμοποίηση πολλών διαφορετικών ταυτοτήτων χρήστη. Κάθε χρήστης μπορεί μόνο να έχει πρόσβαση σε μερικά bytes, αλλά πολλοί μαζί μπορούν να συντονίσουν τις δραστηριότητές τους για να ανακτήσουν τα πλήρη αντίγραφα των ιδιωτικών στοιχείων. Αυτό είναι το θεμελιώδες ζήτημα που εξετάζεται στα ακόλουθα τμήματα.

Η δεύτερη επίθεση υποθέτει ότι η προσπάθεια υποκλοπής γίνεται από την ίδια τη μηχανή αναζήτησης, ενώ οι τελικοί χρήστες σκοπεύουν να κάνουν τίμια χρήση του συστήματος. Με αρκετούς τελικούς χρήστες που συμμετέχουν, οι ποσότητες αποτελεσμάτων που διέρχονται από κάθε μηχανή είναι εξ ολοκλήρου ανεξέλεγκτες. Ο επιχειρησιακός στόχος είναι χιλιάδες τελικοί χρήστες να υποβάλλουν ερωτήσεις μέσω των επιτυχεστέρων μηχανών αναζήτησης, έτσι αυτές οι μηχανές δεν μπορούν να περιορίζονται από οποιοδήποτε είδος ορίων στον αριθμό των προσβάσεων σε έγγραφα.

Προκειμένου να ελεγχθεί ποιος έχει πρόσβαση σε ποιες πληροφορίες, μερικά σχέδια προστασίας στηρίζονται σε έναν έμπιστο τρίτο, που λειτουργεί ως μεσολαβητής διαπίστευσης. Ο βασικός ρόλος αυτού του κέντρου διαπίστευσης

(Authentication Centre) είναι να επιτρέψει στους τελικούς χρήστες να πιστοποιούνται με μοναδικό τρόπο. Ένας κακόβουλος χρήστης πιθανόν να προσπαθήσει να χρησιμοποιήσει πολλές ταυτότητες χρηστών συστήματος, ώστε να υπερνικηθεί ο έλεγχος του κατόχου δεδομένων για το πόση ανατροφοδότηση παράγεται. Το κέντρο διαπίστευσης επικυρώνει στους ιδιοκτήτες περιεχομένου ότι οι οποιεσδήποτε υποβληθείσες ερωτήσεις έχουν προέλθει από έναν εγγεγραμμένο πραγματικό χρήστη που μπορεί να ταυτοποιηθεί με βεβαιότητα. Αν και αυτή η λειτουργία δεν είναι υποχρεωτική για όλα τα συστήματα ασφάλειας που παρουσιάζονται κατωτέρω, η επικύρωση από το κέντρο βοηθά επίσης να εξαλειφθεί το πρόβλημα από τις κακόβουλες μηχανές αναζήτησης που περιγράφεται στην προηγούμενη παράγραφο.

Αυτή η λύση φαίνεται στο Σχήμα 27. Ουσιαστικά, η μηχανή αναζήτησης δεν έχει καθόλου πρόσβαση στην ανατροφοδότηση που παράγει για τους χρήστες της. Τα βέλη παρουσιάζουν όλες τις πιθανές επιλογές μηνυμάτων, ενώ τα ελαφρώς παχύτερα βέλη παρουσιάζουν μόνο μια συγκεκριμένη πορεία που μπορεί να ακολουθηθεί. Η διαδικασία λειτουργεί ως εξής:

1. Ένας τελικός χρήστης συνδέεται σε μια μηχανή αναζήτησης, στέλνοντας ένα μήνυμα διαπίστευσης που περιλαμβάνει χρονοσήμανση, και καθορίζοντας τα επιθυμητά κριτήρια της αναζήτησης με τρόπο που εξαρτάται από την συγκεκριμένη μηχανή.
2. Η μηχανή αναζήτησης ζητά ένα πιστοποιητικό από το κέντρο διαπίστευσης, καταθέτοντας το μήνυμα διαπίστευσης του τελικού χρήστη. Το κέντρο διαπίστευσης ελέγχει την κατάσταση του τελικού χρήστη και τη χρονική σήμανση πριν εκδώσει ένα πιστοποιητικό βραχείας διάρκειας για να επιτρέψει την αναζήτηση.
3. Η μηχανή αναζήτησης υποβάλλει το πιστοποιητικό από κοινού με τον αλγόριθμο αναζήτησής σε ιδιοκτήτες περιεχομένου που επιλέγει είτε η ίδια, είτε έχει προεπιλέξει ο τελικός χρήστης, ανάλογα με την εφαρμογή.

Ο αλγόριθμος είναι παραμετροποιημένος για να προσαρμοστεί στα κριτήρια του τελικού χρήστη, ενώ το συνοδευτικό πιστοποιητικό ταυτοποιεί τον τελικό χρήστη στον ιδιοκτήτη.

4. Όταν ένας ιδιοκτήτης περιεχομένου εκτελέσει την αναζήτηση, συλλαμβάνει το αποτέλεσμα αλλά δεν το διαβιβάζει στην μηχανή αναζήτησης όπως συμβαίνει στην απλή υλοποίηση. Αντί αυτού, μεταβιβάζει το αποτέλεσμα κατευθείαν στον τελικό χρήστη.
5. Εναλλακτικά, δεν υπάρχει ανάγκη για όλες τις πηγές δεδομένων να ξέρουν πώς να φθάσουν σε όλους τους τελικούς χρήστες. Μπορούν απλά να διαβιβάσουν όλα τα αποτελέσματα στο κέντρο διαπίστευσης στο προηγούμενο βήμα, που μπορεί να χρησιμοποιήσει το προφίλ του χρήστη που ούτως ή άλλως διατηρεί για σκοπούς διαπίστευσης για να προωθήσει τα αποτελέσματα.

Σημειώνεται ότι όλες αυτές οι επικοινωνίες πρέπει να είναι ασφαλείς, αλλά αυτό είναι κάτι που εξασφαλίζεται πολύ εύκολα μέσω του πρωτοκόλλου SSL που είναι διαθέσιμο για χρήση μέσω Java[Sun, 2006].

Με την παράκαμψη της μηχανής αναζήτησης από το δρομολόγιο των αποτελεσμάτων, αποκόπτεται κάθε δυνατότητα υποκλοπής δεδομένων από αυτή την πλευρά. Με αυτό το σενάριο, μόνο οι τελικοί χρήστες μπορούν να προσπαθήσουν να κάνουν υποκλοπή των εμπιστευτικών δεδομένων.

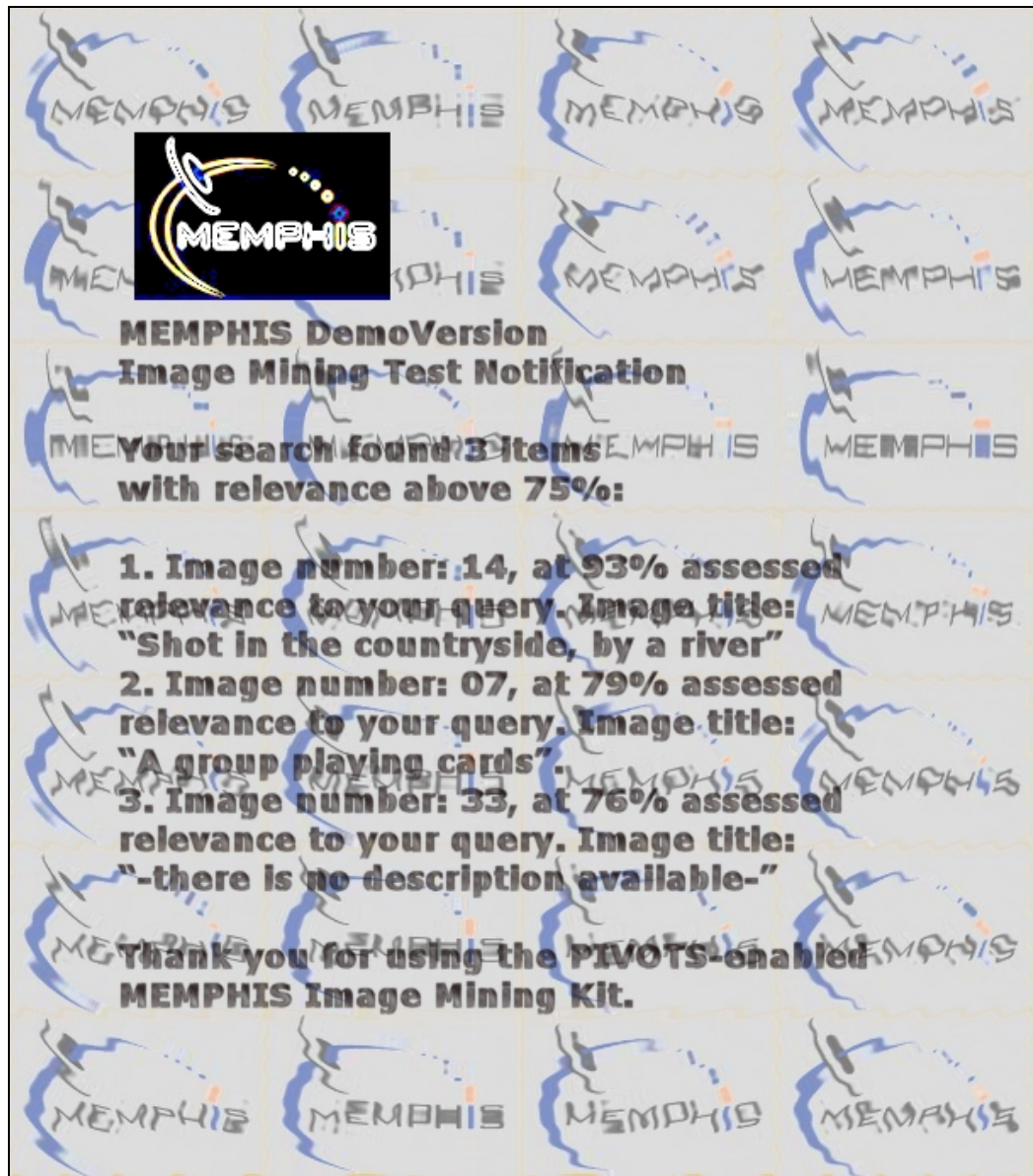
7.3.1.1 Αποτελέσματα μη αναγνώσιμα από μηχανή

Μια κρίσιμη αδυναμία οποιασδήποτε προσπάθειας επίθεσης στο σύστημα είναι ότι αυτό απαιτεί μια μηχανική εξαγωγή των πληροφοριών από την ανατροφοδότηση που παράγεται. Δεδομένου ότι κάθε αποτέλεσμα είναι της τάξης μεγέθους ενός ακέραιου αριθμού ενώ ένα έγγραφο έχει μέγεθος τυπικά εκατοντάδες χιλιάδες ή εκατομμύρια bytes, η σύνθεση κάθε τμήματος που υποκλέπεται για να ανακατασκευαστεί το έγγραφο είναι μια εργασία

που πρέπει να γίνει από λογισμικό. Η εκτέλεση από άνθρωπο θα ήταν υπερβολικά χρονοβόρος και θα είχε υπέρογκο κόστος. Θα ήταν έτσι ευκολότερο να αγοράσει κανείς τις πληροφορίες μια φορά και να δημιουργήσει αντίγραφα, παρά να κάνει αυτό το συνδυασμό byte προς byte.

Αν λοιπόν οι ιδιοκτήτες περιεχομένου, που παράγουν και δρομολογούν τα αποτελέσματα αναζήτησης, είναι σε θέση να τα κωδικοποιήσουν σε μια μορφή που να είναι κατανοητή από ανθρώπους αλλά μη αναγνώσιμη από μηχανή, τότε μπορούν να τα προστατεύσουν ικανοποιητικά. Η προτεινόμενη λύση μας για την εφαρμογή συστήματος ασφαλούς αναζήτησης, σχεδιασμένη σαν μια επέκταση του προγράμματος MEMPHIS [MEMPHIS, 2000] [MEMPHIS, 2003], είναι να μεταδώσει το αποτέλεσμα ως εικόνα. Σε αυτήν την περίπτωση, η μηχανή αναζήτησης καθορίζει ένα πρότυπο (template) στο οποίο το υπολογισμένο αποτέλεσμα πρέπει να παρεμβληθεί. Για το παράδειγμα βιβλίων ή επίσης για τα πολυμέσα, αυτό το πρότυπο είναι ένα προκαθορισμένο κείμενο που λέει ότι εκτίμηση σχετικότητας σε έναν τελικό χρήστη για το στοιχείο που προσδιορίζεται από το την κοινή περίληψη του έχει υπολογιστεί σε μια δεδομένη τιμή. Μόνο η ίδια η τιμή έχει παραχθεί με εξέταση των ιδιωτικών στοιχείων, φυσικά. Όταν το μήνυμα αποτελέσματος συντεθεί, μετασχηματίζεται σε μορφή εικόνας. Αυτή η εικόνα έπειτα διαστρεβλώνεται με έναν τρόπο που θα αποτρέψει την οπτική αναγνώριση του κειμένου που περιέχει από υπολογιστή [Yahoo, 2003] [Ladroy, 2002]. Ένας τέτοιος τρόπος προστασίας πρέπει ωστόσο να οδηγεί σε μια ευχάριστη και φιλική προς το χρήστη επικοινωνία – για παράδειγμα, μια δυσανάγνωστη εικόνα με θόρυβο που στέλνεται στο χρήστη με ηλεκτρονικό ταχυδρομείο δεν είναι μια αποδεκτή μορφή επικοινωνίας. Προφανώς, όσο πιο καλαίσθητη και ευανάγνωστη είναι η εικόνα τόσο αυξάνεται και η πιθανότητα ηλεκτρονικής ανάγνωσης της. Ωστόσο η δοκιμή στα πλαίσια του MEMPHIS έδειξε ότι είναι δυνατό να βρεθεί η χρυσή τομή

[Yannopoulos, 2002a]. Ένα παράδειγμα εφαρμογής της μεθόδου φαίνεται στο Σχήμα 28.



Σχήμα 28: Παράδειγμα εφαρμογής της μεθόδου παραμόρφωσης κειμένου για αποτροπή ανάγνωσης του από μηχανή

7.3.1.2 Άλλες επιλογές προστασίας

Σε πραγματιστικό επίπεδο, η αποτροπή της υποκλοπής ιδιωτικών στοιχείων είναι ζήτημα απλής αριθμητικής. Το μόνο κανάλι πρόσβασης του χρήστη σε οποιοδήποτε έγγραφο, όπως το πλήρες κείμενο ενός βιβλίου,

είναι η σειρά από τα αποτελέσματα που παράγονται από τον όποιο αλγόριθμο υποβάλλεται στον προμηθευτή περιεχομένου. Κατά συνέπεια, εάν ο χρήστης έχει την άδεια να υποβάλλει έναν αλγόριθμο για εκτέλεση μέχρι M φορές σε κάθε χρονική περίοδο T , και λαμβάνει αποτέλεσμα μήκους N bits κάθε φορά, τότε μέχρι $M * N$ bits πληροφορίας μπορούν να κλαπούν σε μια περίοδο T . Αποτελέσματα μήκους ενός ή δύο bytes είναι επαρκή για τις περισσότερες εφαρμογές, έτσι $N \leq 16$, ενώ ένα όριο τριών αναγνώσεων του ίδιου εγγράφου ημερησίως φαίνεται ικανοποιητικό και οδηγεί σε μια φιλική προς το χρήστη προδιαγραφή του M ως 21 και του T ως μια εβδομάδα, για να επιτρέψει τις εκρήξεις αναζητήσεων που ακολουθούνται από μια περίοδο αδράνειας. Κατασκευάζοντας έναν αλγόριθμο αναζήτησης που σε κάθε εκτέλεση του επιστρέφει και ένα διαφορετικό byte του εγγράφου, ένας χρήστης μπορεί έτσι να υποκλέψει μέχρι 42 bytes την εβδομάδα, ή περίπου 2191 bytes ετησίως. Για την κατηγορία εφαρμογών αναζήτησης βιβλίων ή πολυμέσων, το μέγεθος κάθε εγγράφου μετράται σε εκατοντάδες χιλιάδες ή εκατομμύρια bytes. Έτσι η απειλή υποκλοπής με αυτή τη μέθοδο είναι πρακτικά ασήμαντη.

Σαν συμπλήρωμα στον ανωτέρω περιορισμό, ο χρήστης μπορεί να είναι περιορισμένος από μια ισόβια ποσόστωση στο συνολικό μέγεθος των αποτελεσμάτων από οποιοδήποτε δεδομένο έγγραφο. Αυτή η ποσόστωση είναι εξαρτώμενη από την εφαρμογή, αλλά θα μπορούσε να είναι της τάξης μεγέθους του 1% του συνολικού μεγέθους του εγγράφου. Το χρονικό όριο που επιβάλλεται θα βοηθήσει να αυξηθεί το συνολικό όριο για τις περισσότερες εφαρμογές: σε μερικές περιπτώσεις, ένας χρήστης μπορεί να είναι ικανοποιημένος με το να κλέψει μια σελίδα ενός βιβλίου – έτσι, το γενικό όριο θα έπρεπε να κατεβεί σε μισή σελίδα – αλλά, όντας σε θέση να λάβει αυτήν την σελίδα μόνο μετά από καθυστέρηση ενός έτους, ο χρήστης

θα μπορούσε πολύ ευκολότερα να επισκεφθεί ένα φυσικό βιβλιοπωλείο και να αποστηθίσει το ισοδύναμο απόσπασμα του κειμένου.

Όταν οι χρήστες μπορούν να συνεργαστούν, αυτή η μέθοδος καταρρέει. Έτσι, είναι εφαρμόσιμη μόνο σε εξειδικευμένες εφαρμογές με πολύ μικρό αριθμό χρηστών, όπως αυτές που περιγράφουμε στην αντίστοιχη ενότητα παρακάτω. Επίσης μπορεί να είναι χρήσιμη σε συνδυασμό με τις άλλες μεθόδους προστασίας που περιγράφονται στη συνέχεια, παραδείγματος χάριν, για να προσφέρει την εγγύηση ότι ακόμα και ένα σφάλμα λογισμικού θα οδηγήσει σε διαρροή ασήμαντης ποσότητας πληροφορίας μέχρι να επισκευασθεί.

Η αμεσότερη δυνατή προστασία είναι ίσως η διαβίβαση του αποτελέσματος σε μια μόνο για ανάγνωση συσκευή. Παραδείγματος χάριν, ένα συνοπτικό αποτέλεσμα μπορεί να στέλνεται σε ένα κινητό τηλέφωνο ως μήνυμα SMS. Η μεταφορά των μηνυμάτων από ένα κινητό τηλέφωνο σε ένα PC είναι μάλλον δυσκίνητη διαδικασία. Αποστολή μηνυμάτων SMS συνεπάγεται φυσικά ένα ορισμένο κόστος, αλλά το κατά πόσο αυτό αποτελεί πρόβλημα εξαρτάται από την εφαρμογή. Το πραγματικό μειονέκτημα αυτής της προσέγγισης είναι ότι απαιτεί τη χρήση υλικού με ειδικά χαρακτηριστικά γνωρίσματα, δηλαδή είτε ακριβή, είτε αργή επικοινωνία με το περιβάλλον. Όμως, εάν τέτοιο υλικό είναι διαθέσιμο ούτως ή άλλως, λόγω άλλων απαιτήσεων, τότε η λύση αυτή προσφέρεται.

Τέλος, είναι δυνατό να χρησιμοποιηθεί το απλό πρότυπο της παραγράφου 7.2 χωρίς τροποποίηση, με μια σημαντική επιχειρησιακή διαφοροποίηση: την εφαρμογή μιας χρέωσης για κάθε byte αποτελέσματος, και μάλιστα μιας αρκετά υψηλής χρέωσης. Χαμηλότερες χρεώσεις θα μπορούσαν να χρησιμοποιηθούν σε συνδυασμό με την προστασία βάσει ορίων, όπως περιγράφεται ανωτέρω. Η υπηρεσία αναζήτησης δεν πρόκειται να γίνει έτσι ακριβή. Ένας τελικός χρήστης μπορεί να πληρώσει χωρίς ενδοιασμό λίγα

λεπτά για ένα χρήσιμο αποτέλεσμα, θεωρώντας την υπηρεσία φτηνή, αλλά η υποκλοπή πληροφορίας με αυτή τη χρέωση γίνεται τότε ιδιαίτερα ασύμφορη: ένα βιβλίο μπορεί να στοιχήσει χιλιάδες ευρώ με αυτόν τον τρόπο.

7.3.1.3 Ασφάλεια του χρήστη

Ο κύριος στόχος μας σε αυτή την εργασία είναι να εξασφαλίσουμε ότι ένας κακόβουλος χρήστης δεν έχει τη δυνατότητα να προκαλέσει ζημία σε έναν προμηθευτή περιεχομένου. Πρέπει επίσης να σημειώσουμε, εντούτοις, ότι ένας κακόβουλος προμηθευτής περιεχομένου θα μπορούσε να προκαλέσει ζημία σε έναν μη προστατευμένο χρήστη διαστρεβλώνοντας το αποτέλεσμα αναζήτησης που παράγεται μέσα στο sandbox. Παραδείγματος χάριν, ένας κακόβουλος προμηθευτής στοιχείων για να πωλήσει τα βιβλία του στους τελικούς χρήστες, θα μπορούσε να αποφασίσει να αντικαταστήσει περιστασιακά τα πραγματικά αποτελέσματα με καλύτερα. Τέτοιες επιθέσεις είναι εύκολο να αντιμετωπιστούν, εντούτοις.

Εάν το παράδειγμα της παραγράφου 7.2 χρησιμοποιείται, με τους χρήστες να υποβάλλουν ανεξάρτητα τον κώδικα τους, τότε οι χρήστες μπορούν προστατευθούν με τη χρησιμοποίηση αυθαίρετων και μεταβλητών κωδικοποιήσεων του αποτελέσματος.

Εάν μηχανές αναζήτησης τρίτων χρησιμοποιούνται, η χρήση περίπλοκων σχημάτων κωδικοποίησης γίνεται αρκετά περίπλοκη. Σαν εναλλακτική λύση, η προστασία μπορεί να εξασφαλιστεί με την υποστήριξη του κέντρου διαπίστευσης. Λαμβάνοντας από τους προμηθευτές τα αποτελέσματα της αναζήτησης, το κέντρο διαπίστευσης μπορεί να εφαρμόσει ένα μετασχηματισμό πάνω τους. Έτσι τα αποτελέσματα που επιστρέφει ο πάροχος δεν είναι στη μορφή που εμφανίζονται στον τελικό χρήστη. Αν ο πάροχος προσπαθήσει να επηρεάσει τα αποτελέσματα, θα πρέπει να

αποκρυπτογραφήσει τον χρησιμοποιούμενο μετασχηματισμό. Ένας περίπλοκος μετασχηματισμός μπορεί να παράσχει το επιθυμητό επίπεδο ασφαλείας.

7.3.2 Χρηματοδότηση του Κέντρου Διαπίστευσης

Ο ρόλος του κέντρου διαπίστευσης είναι αυτός ενός γενικής χρήσης έμπιστου μεσάζοντα. Καθώς ένα τέτοιο έμπιστο συμβαλλόμενο μέρος είναι ούτως ή άλλως απαραίτητο για τη βασική λειτουργία της διαπίστευσης τελικών χρηστών, αυτό μπορεί να αξιοποιηθεί περαιτέρω στην αντιμετώπιση τεχνικών ζητημάτων που προκύπτουν λόγω έλλειψης εμπιστοσύνης μεταξύ των μερών, όπως περιγράφεται στα προηγούμενα τμήματα. Οι τεχνολογίες που συσσωρεύονται σε αυτόν τον κόμβο της αρχιτεκτονικής (π.χ., από την επεξεργασία εικόνας ως τις διεπαφές για εξειδικευμένες συσκευές όπως κινητά τηλέφωνα) είναι ήδη υπάρχουσες και κανένα κόστος ανάπτυξης δεν προκύπτει με το να τοποθετηθούν εδώ. Οι δαπάνες που υπάρχουν σχετίζονται με την νομική καθιέρωση του διαμεσολαβητικού ρόλου, και οι δαπάνες εγκατάστασης και λειτουργίας των σχετικών συστημάτων. Δεδομένου ότι η επεξεργασία περιορίζεται σε σχετικά λίγες, συγκεκριμένες διαδικασίες ανά αίτημα και ανά στοιχείο της ανατροφοδότησης, οι γενικές δαπάνες θα είναι αρκετά περιορισμένες.

Κεφάλαια για την υποστήριξη του κέντρου θα πρέπει να διατεθούν από τα υπόλοιπα μέρη. Οι ιδιοκτήτες περιεχομένου έχουν κάθε κίνητρο να χρηματοδοτήσουν την χρήση του κέντρου διαπίστευσης, αφού τους προσφέρει το απαραίτητο επίπεδο ασφάλειας ώστε να μπορέσουν να διαθέσουν προηγμένες υπηρεσίες αναζήτησης στους τελικούς χρήστες τους. Οι μηχανές επίσης έχουν κίνητρο να το χρηματοδοτήσουν, επειδή αυτή είναι η τεχνολογία που επιτρέπει τη λειτουργία τους. Πιθανώς, οι μηχανές αναζήτησης θα έχουν κάτι να κερδίσουν από την προσφερόμενη υπηρεσία στους τελικούς χρήστες, είτε συνδρομές, είτε από διαφημίσεις, ή

οποιοδήποτε άλλο μοντέλο λειτουργίας. Η χρηματοδότηση τότε του κέντρου διαπίστευσης μπορεί να θεωρηθεί απλά σαν ένα επιπλέον λειτουργικό κόστος που όμως πολλαπλασιάζει τις επιχειρησιακές δυνατότητες της μηχανής. Τέλος, και οι ίδιοι οι τελικοί χρήστες μπορούν να επιβαρυνθούν μέρος του κόστους χρηματοδότησης, αφού έτσι μπορούν να έχουν πρόσβαση σε μια πολύ βελτιωμένη υπηρεσία αναζήτησης. Όπως εξηγούμε στην παράγραφο 5.2, υπάρχουν πραγματικές εφαρμογές όπου ο τελικός χρήστης είναι πρόθυμος να καταβάλει μια μικρή αμοιβή για κάθε αποτέλεσμα αναζήτησης που έλαβε στα πλαίσια μιας υπηρεσίας με σημαντικές βελτιώσεις σε σχέση με τις σημερινές δωρεάν επιλογές.

Τέλος, οι κυβερνητικές υπηρεσίες έχουν σημαντικό κίνητρο να χρηματοδοτήσουν ή να προσφέρουν την υπηρεσία ενός κέντρου διαπίστευσης σε όλα τα άλλα συμβαλλόμενα μέρη, προκειμένου να ενισχυθεί το ηλεκτρονικό εμπόριο ή για να υποστηρίξουν τις επιστημονικές ή ειδικές εφαρμογές που περιγράφονται στις παραγράφους 7.4.2 και 7.4.3.

7.4 ΕΦΑΡΜΟΓΕΣ

Το σύστημα που περιγράφουμε έχει ορισμένους περιορισμούς στην προδιαγραφή του, υπαγορευμένους για λόγους ασφαλείας, που ισχύουν για κάθε εφαρμογή. Όπως έχουμε επισημάνει ήδη, μόνο περιορισμένου, προκαθορισμένου μεγέθους αποτέλεσμα μπορεί να επιστραφεί. Αποτελέσματα από διαφορετικά έγγραφα μπορούν να συνδυαστούν κατά έναν πολύ περιορισμένο τρόπο, δηλαδή κάθε έγγραφο πρέπει να εξεταστεί σε απομόνωση και ο συνδυασμός να γίνει στη συνέχεια, όπου δεν υπάρχει πλέον η πρόσβαση στα έγγραφα. Αναγνώσιμη από μηχανή ανατροφοδότηση στους τελικούς χρήστες πιθανώς να μη μπορεί να επιτραπεί. Τέλος, οι πάροχοι περιεχομένου δεν μπορούν να ελέγξουν τι μπαίνει περιλαμβάνει το αποτέλεσμα που παράγεται, έτσι το σύστημα είναι εφαρμόσιμο μόνο σε

έγγραφα που πρέπει να κρατηθούν μυστικά στο σύνολο τους, αλλά οποιοδήποτε τμήμα τους μεγέθους συγκρίσιμου με το αποτέλεσμα μπορεί να δημοσιευθεί χωρίς πρόβλημα.

Ωστόσο, στις εφαρμογές για τις οποίες είναι κατάλληλο, το σύστημα αυτό μπορεί να εφαρμοστεί με μικρή προσπάθεια και αρκετά ωφέλη. Προφανώς υπάρχουν προγραμματιστικές εργασίες που εξαρτώνται από την εφαρμογή. Για παράδειγμα, οι διεπαφές με τα συστήματα των παρόχων δεδομένων πρέπει να αναπτυχθούν και οι μηχανές αναζήτησης πρέπει να αναπτύξουν υποστήριξη για την υπηρεσία. Σε γενικές όμως γραμμές το σύστημα είναι σχεδιασμένο ώστε να αυξήσει το πεδίο εφαρμογής των υφιστάμενων τεχνολογιών αναζήτησης, δίνοντας τους πρόσβαση σε δεδομένα που τώρα δεν μπορούν να προσπελάσουν.

Στη συνέχεια περιγράφονται τρεις τελείως διαφορετικές δυναμικές περιοχές εφαρμογής της προτεινόμενης μεθόδου. Στην παρουσίαση τους περιλαμβάνεται και μια υψηλού επιπέδου περιγραφή των τεχνολογιών αναζήτησης στις οποίες είναι βασισμένες, προκειμένου να δοθεί μια πλήρης άποψη της προτεινόμενης λύσης.

7.4.1 Περιήγηση Βιβλιοπωλείων

Αυτή η ιδέα έχει ήδη συζητηθεί με τη μορφή παραδείγματος σε όλη την παραπάνω ανάπτυξη. Έτσι δεν θα επαναλάβουμε την περιγραφή της εδώ, αλλά θα περιγράψουμε τα αποτελέσματα των ερευνών αγοράς που πραγματοποιήθηκαν στο πλαίσιο του έργου MEMPHIS [MEMPHIS, 2000] [MEMPHIS, 2003]. Το MEMPHIS είναι ένα σύνθετο έργο που περιλαμβάνει ένα εύρος στόχων που εκτείνεται από τεχνικές επεξεργασίας φυσικής γλώσσας και πολυγλωσσικές υπηρεσίες, μέχρι προηγμένη αναγνώριση θέματος και σχετικότητας για βιβλία, ειδήσεις, εικόνες, και άλλα μέσα, και

τεχνικές παρουσίασης των αποτελεσμάτων σε συσκευές περιορισμένων δυνατοτήτων, όπως κινητά τηλέφωνα.

Το αποτέλεσμα που είναι σχετικό με την έρευνα που παρουσιάζουμε εδώ, είναι αυτό της έρευνας αγοράς που πραγματοποιήθηκε για να θέσει τις προτεραιότητες του έργου. Η έρευνα έδειξε ότι οι τελικοί χρήστες έδειξαν μεγάλο ενδιαφέρον σε ειδοποιήσεις και ειδήσεις προσαρμοσμένες στα ενδιαφέροντα τους, στις δυνατότητες της συσκευής τους και στις προτιμήσεις παρουσίασης τους. Υπήρξε πολύ θετική αντιμετώπιση σε υπηρεσίες που μπορούν να τους εξοικονομήσουν χρόνο και προσπάθεια αναζήτησης υλικού σε βιβλιοπωλεία ή καταστήματα ψηφιακών μέσων. Για να καταστεί δυνατή μια τέτοια υπηρεσία, χρειάζεται η ανάπτυξη λογισμικού που αντιστοιχίζει τις προτιμήσεις των χρηστών με τα χαρακτηριστικά βιβλίων και άλλων μέσων, με μεγάλο βαθμό ακρίβειας. Η αρχική κατεύθυνση που ακολουθήθηκε στο MEMPHIS ήταν η εφαρμογή τεχνικών κατηγοριοποίησης με χρήση τεχνολογιών φυσικής γλώσσας στις περιλήψεις των βιβλίων, και στη συνέχεια η αντιστοίχιση των κατηγοριών με τις προτιμήσεις των χρηστών. Το σύστημα που προτείνουμε προσφέρει τη δυνατότητα να γίνει αυτή η αντιστοίχιση σε πολύ πιο συμπαγή βάση.

Προφανώς, οι χρήστες θα προτιμούσαν η υπηρεσία να είναι δωρεάν. Ωστόσο η έρευνα αγοράς έδειξε ότι οι χρήστες θα ήταν διατεθειμένοι να πληρώσουν για ένα μηχανισμό έρευνας και ειδοποίησης, υποθέτοντας ότι τα αποτελέσματα θα έχουν ικανοποιητική ακρίβεια. Εξάλλου η αγορά βιβλίων μέσω ηλεκτρονικού εμπορίου είναι μια λύση οικονομικά συμφέρουσα, αλλά χρειάζεται τεχνολογική υποστήριξη για την αναγνώριση βιβλίων που ενδιαφέρουν. Σπάνια ένας πελάτης θα μπει στη διαδικασία να εξετάσει βιβλία σε ένα φυσικό βιβλιοπωλείο, και στη συνέχεια να τα αγοράσει ηλεκτρονικά.

Αυτά τα ευρήματα δικαιολογούν δύο από τις προσεγγίσεις που προτείναμε για την αντιμετώπιση των θεμάτων ασφαλείας, δηλαδή την αποστολή των

αποτελεσμάτων σε συσκευές μόνο ανάγνωσης όπως fax ή SMS και τη χρέωση της υπηρεσίας αναζήτησης.

7.4.2 Ελεγχόμενη Πρόσβαση σε Απόρρητες Εικόνες για Υπηρεσίες Ασφαλείας

Μια άλλη εφαρμογή που έχουμε εξετάσει περιλαμβάνει την αξιολόγηση του περιεχομένου διαβαθμισμένων βάσεων δεδομένων εικόνας, όπως φωτογραφίες ταυτότητας. Μια τέτοια εφαρμογή θα ήταν πολύτιμη σε περιπτώσεις όπως αυτή της διεθνούς αστυνομικής συνεργασίας, αλλά θα μπορούσε να γενικευτεί, παραδείγματος χάριν, στην ελεγχόμενη διανομή υλικού μεταξύ διαφημιστικών αντιπροσωπειών. Οι αστυνομίες διαφορετικών χωρών θα ήθελαν να καταστήσουν τα διαβαθμισμένων στοιχεία τους προσβάσιμα από άλλες αστυνομίες. Οι πλήρεις συλλογές δεδομένων δεν θα μπορούσαν να γίνουν ανοιχτά διαθέσιμες σε διερεύνηση από εμπειρογνώμονες για λόγους προστασίας προσωπικών δεδομένων, αλλά αν μια ξένη αστυνομική υπηρεσία μπορούσε να εντοπίσει ένα μικρό αριθμό στοιχείων που την ενδιέφεραν, η κοινοποίησή τους θα μπορούσε να εξεταστεί ή, τουλάχιστον, να δοθούν κάποιες σχετικές πληροφορίες. Τέτοιες βάσεις δεδομένων θα μπορούσαν να είναι ανοιχτά διαθέσιμες υπό την έννοια ότι αναζητήσεις θα μπορούσαν να αποσταλούν και να εκτελεστούν οποιαδήποτε στιγμή, χωρίς την ανάγκη ανθρώπινης επίβλεψης. Εντούτοις, η λειτουργία της θα μπορούσε να γίνει μόνο υπό πολύ στενά όρια επιπέδων εξυπηρέτησης. Το PIVOTS θα μπορούσε να παράσχει την υποδομή για το σχεδιασμό και την λειτουργία μιας τέτοιας υπηρεσίας, με τήρηση όλων των περιορισμών ασφαλείας και επιπέδου χρήσης.

7.4.3 Πειραματική Δοκιμή Μεθόδων Μηχανικής Μάθησης για Ιατρικές ή Άλλες Εφαρμογές

Μια σοβαρή δυσχέρεια που εμποδίζει συχνά τη βελτίωση των τεχνικών μηχανικής μάθησης σε οποιαδήποτε εφαρμογή είναι η ανεπαρκής διαθεσιμότητα δεδομένων εκπαίδευσης [Bishop, 1995]. Συχνά, αυτή η έλλειψη επιδεινώνεται από κοινωνικές ή οικονομικές δυσκολίες στη διάθεση των δεδομένων. Παραδείγματος χάριν, υπεύθυνοι για την ανάπτυξη εργαλείων ανάλυσης αγοράς μπορεί να χρειάζονται ιστορικά στοιχεία αγοράς που είναι καταγεγραμμένα με μεγάλη λεπτομέρεια, αλλά είναι διαθέσιμα αντί σημαντικού αντιτίμου. Ενδεχομένως, θα αναγκαστούν να αγοράσουν ένα σχετικά μικρό υποσύνολο των συνολικών δεδομένων. Ομοίως, τα ιατρικά αρχεία για τα ιστορικά των ασθενών δεν μπορούν, γενικά, να κοινοποιηθούν για ερευνητικούς σκοπούς. Οι ερευνητές έχουν πρόσβαση μόνο σε μικρά σύνολα δεδομένων με αρχεία ασθενών που έδωσαν ρητά την συγκατάθεση τους για αυτόν το σκοπό.

Τα σύνολα δεδομένων που χρησιμοποιούνται στις μεθόδους μηχανικής μάθησης πρέπει να είναι χωρισμένα σε τρία μη επικαλυπτόμενα σύνολα: τα σύνολα εκπαίδευσης, δοκιμής, και επικύρωσης. Βασικά, οι αλγόριθμοι που εφαρμόζονται στηρίζονται στα δεδομένα εκπαίδευσης για να δομήσουν μια λύση για το πρόβλημα που εξετάζεται, και στα δεδομένα δοκιμής για βραχυπρόθεσμες αξιολογήσεις της απόδοσή τους σε μια επαναληπτική διαδικασία βελτίωσης. Αυτή η διαδικασία μπορεί, για διάφορους λόγους, να οδηγήσει σε παραπλανητικά αποτελέσματα. Το σύνολο επικύρωσης χρησιμοποιείται έτσι ως τελική δοκιμή που καθορίζει εάν μια αποδεκτή λύση έχει βρεθεί ή όχι. Αν η λύση αλλαχθεί μετά τη δοκιμή με τα στοιχεία επικύρωσης υπάρχει κίνδυνος διαστρέβλωσης του αποτελέσματος, έτσι τα στοιχεία επικύρωσης πρέπει να χρησιμοποιηθούν μία μόνο φορά. Εφαρμόζεται μόνο για να συγκρίνουμε τα παραγόμενα αποτελέσματα με ένα

προκαθορισμένο σύνολο από επιθυμητά αποτελέσματα, ως τελική αξιολόγηση της ποιότητας απόδοσης της μεθόδου που έχει κατασκευαστεί.

Η έλλειψη δεδομένων δημιουργεί έτσι ποικίλα προβλήματα. Η απομόνωση στοιχείων για λόγους επικύρωσης υπογραμμίζει τη μη διαθεσιμότητα τους στα στάδια της ανάπτυξης και δοκιμής. Έτσι συχνά τα σύνολα επικύρωσης ελαχιστοποιούνται σε μέγεθος, με αποτέλεσμα η τελική αποτίμηση της μεθόδου να είναι ανακριβής. Αν παραστεί η ανάγκη τροποποίησης της μεθόδου μετά την επικύρωση, τότε η νέα επικύρωση θα πρέπει να γίνει με διαφορετικό σύνολο δεδομένων που όμως δεν είναι διαθέσιμα. Έτσι χρησιμοποιείται το ίδιο σύνολο για την επικύρωση, με κίνδυνο παραμόρφωσης της παραγόμενης λύσης. Κάποιες παραμορφώσεις είναι εύκολα ανιχνεύσιμες, αλλά η φύση τους είναι τέτοια που γενικά είναι δύσκολο να εντοπιστούν [Bishop, 1995].

Η μέθοδος αναζήτησης που περιγράψαμε μπορεί να αξιοποιηθεί σε αυτό το πλαίσιο διακρίνοντας ότι τα τρία σύνολα δεδομένων χρησιμοποιούνται με διαφορετικό τρόπο. Τα σύνολα δοκιμής και εκπαίδευση χρησιμοποιούνται έντονα και επαναληπτικά. Συχνά, τα στοιχεία εκπαίδευσης στην πραγματικότητα ενσωματώνονται μερικώς ή πλήρως στην ίδια τη λύση. Όμως το σύνολο επικύρωσης χρησιμοποιείται μόνο μια φορά για την παραγωγή στατιστικών μέτρων απόδοσης.

Έτσι η χρήση της μεθόδου μπορεί να υποστηρίξει αυτή τη διαδικασία, δίνοντας πρόσβαση σε δεδομένα για λόγους επικύρωσης. Οι ιδιοκτήτες των δεδομένων μπορούν εύκολα να καταστήσουν τα σύνολα ιδιωτικών δεδομένων διαθέσιμα μέσα από την αρχιτεκτονική του συστήματος ασφαλούς αναζήτησης. Δεν διατρέχουν κανέναν κίνδυνο αποκάλυψης των πραγματικών στοιχείων, αλλά επιτρέπουν στις μεθόδους που έχουν αναπτυχθεί να επικυρωθούν με εκτενείς δοκιμές. Ανάλογα με την εφαρμογή, οι ιδιοκτήτες μπορεί να έχουν διάφορα κίνητρα για να συνδράμουν τη

διαδικασία. Για παράδειγμα, οι πάροχοι οικονομικών δεδομένων θα μπορούσαν, αναγνωρίζοντας ότι τα τιμολόγια τους είναι πολύ υψηλά για ερευνητικούς σκοπούς, να προσφέρουν διαφορετικά τιμολόγια για την εκτέλεση υπολογισμών σε αυτά χωρίς την αποκάλυψη των ίδιων των δεδομένων. Στην περίπτωση των ιατρικών ιστορικών, το κίνητρο είναι απλούστερο. Το απόρρητο των προσωπικών δεδομένων των ασθενών δεν μπορεί να παραβιαστεί για οποιοδήποτε λόγο, αλλά το προτεινόμενο παράδειγμα επιτρέπει στην έρευνα να εκμεταλλευτεί το πλήρες σύνολο των διαθέσιμων ιατρικών στοιχείων για σκοπούς επικύρωσης, χωρίς την αποκάλυψη τους σε τρίτους. Βέβαια οι μέθοδοι που επικυρώνονται θα έχουν προηγουμένως αναπτυχθεί και δοκιμαστεί βάσει δεδομένων που διατέθηκαν με τη συγκατάθεση των ασθενών για ελεύθερη μελέτη τους.

Εξετάζοντας το Σχήμα 27 ακόμα μια φορά, μπορούμε να διακρίνουμε την ευελιξία της αρχιτεκτονικής. Οι τελικοί χρήστες και οι μηχανές αναζήτησης δεν είναι ξεχωριστοί σε αυτή την περίπτωση, αφού και οι δύο ρόλοι εκτελούνται από τους ερευνητές που αναπτύσσουν το λογισμικό. Εντούτοις, οι ιδιοκτήτες περιεχομένου είναι ακόμα ανεξάρτητοι και δεν συνδέονται με τους ερευνητές με κάποια σχέση εμπιστοσύνης. Και οι δύο τύποι συμβαλλόμενων μερών είναι διαθέσιμοι σε μεγάλους αριθμούς. Μια μεγάλης κλίμακας εγκατάσταση του συστήματος θα σήμαινε έτσι ότι κάθε βιομηχανικός, ιατρικός ή ακαδημαϊκός ερευνητής θα μπορούσε να επικυρώσει και να αποτιμήσει την απόδοση πειραματικού λογισμικού επάνω σε λεπτομερή ιατρικά αρχεία οποιουδήποτε συμμετέχοντος νοσοκομείου, ιδρύματος ιατρικής έρευνας ή ακόμα και κυβερνητικού οργανισμού.

Τα οφέλη που προσφέρονται έτσι στους ερευνητές είναι διπλά. Με την εξάλειψη της ανάγκης για ένα ανοιχτό σύνολο επικύρωσης, μπορούν να εκπαιδεύσουν τα συστήματά τους με περισσότερα στοιχεία εκπαίδευσης. Αυτό θα είναι ίσως από μόνο του επαρκές για να βελτιώσει τα αποτελέσματα.

Η απόδοση της εκπαίδευσης κατάρτισης υποβιβάζεται συχνά σε μεγάλο βαθμό από την έλλειψη δεδομένων. Αφετέρου, η διαδικασία επικύρωσης μπορεί πλέον να εκτελεστεί σε ένα πλούτο δεδομένων. Επομένως η επικύρωση θα δώσει μια πολύ αξιόπιστη αποτίμηση της απόδοσης της μεθόδου.

Πρέπει βέβαια να σημειωθεί ότι ευαίσθητες πληροφορίες με πολύ μικρό μέγεθος θα μπορούσαν να κλαπούν, τα ιατρικά αρχεία ενός συγκεκριμένου ασθενή. Οποιοσδήποτε τέτοιες κρίσιμες πληροφορίες, όπως τα ονόματα ασθενών ή οι αριθμοί κοινωνικής ασφάλισης, μπορούν απλά να αφαιρούνται από κάθε έγγραφο πριν αυτό εισαχθεί στο Sandbox. Τα υπόλοιπα στοιχεία, τα οποία δεν μπορούν να δημοσιευθούν σύμφωνα με την τρέχουσα νομοθεσία, μπορούν ωστόσο να γίνουν διαθέσιμα για την έρευνα μέσω του προτεινόμενου συστήματος προστασίας των προσωπικών δεδομένων.

8 ΣΥΜΠΕΡΑΣΜΑΤΑ

Στην παρούσα διατριβή παρουσιάζονται μέθοδοι και τεχνικές λύσεις για την αναζήτηση και ανάκτηση πληροφορίας από ετερογενή συστήματα με ομοιόμορφο και ανεξάρτητο από τις πηγές της πληροφορίας τρόπο. Η μεθοδολογία Ozonε αποτελεί μια προσέγγιση στην ολοκλήρωση ετερογενών συστημάτων με οντολογίες, η οποία αντιμετωπίζει τα προβλήματα αναπαράστασης πληροφορίας, αρχιτεκτονικής και διάφορα λειτουργικά προβλήματα (όρια μεταξύ συστημάτων, διεπαφές, επεκτασιμότητα, συντηρησιμότητα, επιδόσεις), και περιγράφει βασικές υποστηρικτικές διαδικασίες και εργαλεία για την διαχείριση του έργου κατασκευής ενός ολοκληρωμένου συστήματος. Η μέθοδος ασφαλούς αναζήτησης απευθύνεται στο πιο εξειδικευμένο πρόβλημα αναζήτησης πληροφορίας όταν η χωρίς περιορισμούς πρόσβαση στην πληροφορία που εξετάζεται δεν είναι δυνατή. Οι δύο μεθοδολογίες δοκιμάστηκαν στα πλαίσια πειραματικών προγραμμάτων και τα αποτελέσματα έδωσαν μια πρώτη απόδειξη (proof of concept) της εφικτότητας πρακτικής εφαρμογής τους.

Έτσι η εργασία τόσο στο Ozonε όσο και στον τομέα της ασφαλούς αναζήτησης είναι προσανατολισμένη στον πειραματικό έλεγχο των ιδεών και μεθοδολογιών στις οποίες βασίζονται. Έχοντας καλύψει αυτό το βήμα,

ακολουθεί η ανάγκη μετεξέλιξης τους ώστε να πλησιάσουν περισσότερο τα βιομηχανικά πρότυπα και να ενσωματώσουν τις τελευταίες τάσεις και ιδέες στη βιομηχανία λογισμικού, ώστε να καταστούν κατάλληλα για βιομηχανική αξιοποίηση.

Όσο αφορά τη μέθοδο ασφαλούς αναζήτησης που περιγράψαμε, ο βασικός άξονας εξέλιξης θα ήταν η ένταξη του στο ευρύτερο πλαίσιο του Ozone. Παρόλο που και οι δύο μεθοδολογίες βασίζονται σε αντικειμενοστραφή υποδομή και οι υλοποιήσεις και των δύο έχουν βασιστεί σε τεχνολογίες Java Standard και Enterprise Edition (J2EE), δεν είναι διασυνδεδεμένες μεταξύ τους. Οι δύο μεθοδολογίες είναι συμπληρωματικές και όχι ανταγωνιστικές μεταξύ τους, καθώς αντιμετωπίζουν διαφορετικά προβλήματα που μπορούν κάλλιστα να συνυπάρχουν στο ίδιο ολοκληρωμένο σύστημα. Έτσι, θα ήταν επωφελής μια ενοποίηση των δύο συστημάτων έτσι ώστε η ασφαλής αναζήτηση να ενταχθεί στην αρχιτεκτονική του Ozone σαν μια επιπλέον υπηρεσία προς τις εφαρμογές, που να μπορεί να χρησιμοποιηθεί κατά περίπτωση όταν η φύση της εφαρμογής το απαιτεί και οι πηγές δεδομένων το επιτρέπουν.

Όσο αφορά το Ozone, η παραπέρα εξέλιξη πρέπει να γίνει λαμβάνοντας υπόψη το χαρακτήρα και τη δυναμική του, και τον τρόπο που τοποθετείται στο γενικότερο τομέα της ολοκλήρωσης πηγών δεδομένων, ένα ευρύτατο πεδίο που συνδυάζει πολλά περίπλοκα επιμέρους προβλήματα. Το συγκεκριμένο πρόβλημα που αντιμετωπίζει το Ozone είναι η εφικτότητα και η δυνατότητα συντήρησης των λύσεων ολοκλήρωσης πληροφορίας που βασίζονται σε οντολογίες. Το πρόβλημα αυτό είναι κρίσιμο για την πρακτική εφαρμογή τέτοιων λύσεων, ωστόσο δεν έχει λάβει την ανάλογη προσοχή από την επιστημονική και την ευρύτερη ερευνητική κοινότητα. Έτσι στην έρευνα που περιγράφεται στην παρούσα διατριβή επιχειρήθηκε η δημιουργία μιας γέφυρας ανάμεσα στις επιτυχείς ερευνητικές προσεγγίσεις στην οντολογική ολοκλήρωση πληροφορίας και τις δυνατότητες, ανάγκες και περιορισμούς της βιομηχανίας πληροφορικής.

Το Ozone σχεδιάστηκε σαν ένα εύχρηστο σύστημα που εκθέτει μόνο την απολύτως αναγκαία οντολογική λειτουργικότητα προκειμένου να καταστεί εφικτή η αναζήτηση και ανάκτηση πληροφορίας, καλύπτοντας τα πιο περίπλοκα οντολογικά χαρακτηριστικά και διεργασίες λογικής. Παρά το ότι το σύστημα επικεντρώνει στην ολοκλήρωση σχεσιακών βάσεων, είναι ανοικτό σε κάθε είδους πηγή δεδομένων. Ακόμα και μια άλλη πλατφόρμα ολοκλήρωσης πηγών δεδομένων θα μπορούσε να συνδεθεί στο Ozone σαν μια μοναδική πηγή.

Έτσι το Ozone δεν αποτελεί μια γενικευμένη λύση στο πρόβλημα της ολοκλήρωσης πληροφορίας. Αντιθέτως, προσφέρει λύσεις μόνο όσο αφορά την αναζήτηση και εξαγωγή δεδομένων από πηγές. Δεν είναι συγκρίσιμο με μια γλώσσα πολλαπλών βάσεων δεδομένων (multi-database language) αφού δεν υποστηρίζει την τροποποίηση ή εισαγωγή στοιχείων, και δεν υποστηρίζει ερωτήματα που συνδέουν περισσότερες από μία πηγές δεδομένων. Αυτές οι παραχωρήσεις σε λειτουργικότητα επιτρέπουν στο Ozone να υποστηρίξει τη δημιουργία, συντήρηση και επέκταση ομοσπονδιών πηγών δεδομένων σε μικρότερους χρόνους, με λιγότερη προσπάθεια και πολυπλοκότητα και, εν κατακλείδι, με μικρότερο κόστος.

Υπό αυτό το πρίσμα, δεν έχει νόημα να επιχειρηθεί μια επέκταση του Ozone ώστε να γίνει μια καθολική και γενική πλατφόρμα ολοκλήρωσης. Η φυσική προέκταση του είναι προς την κάλυψη συγκεκριμένων προβλημάτων και αδυναμιών του συστήματος. Υπάρχει μια σειρά τέτοιων θεμάτων προς αντιμετώπιση. Αρχικά απαιτείται η προσαρμογή της υποδομής του συστήματος στα πλέον διαδεδομένα πρότυπα. Το πρώτο βήμα σε αυτή την κατεύθυνση είναι η μεταφορά της προδιαγραφής των interfaces του συστήματος από τη Java σε web services. Τα web services έχουν γίνει το de facto πρότυπο της βιομηχανίας στην επικοινωνία μεταξύ κατανεμημένων συστημάτων και αποτελούν την πλέον κατάλληλη τεχνολογία, τόσο από τεχνικής όσο και από στρατηγικής πλευράς, για την υποδομή του Ozone. Η χρήση τεχνολογιών WSDL, XML και XSD για τον

ορισμό των interfaces θα δώσει επιπλέον πολύ μεγαλύτερη ελαστικότητα στην αρχιτεκτονική, αφού θα είναι δυνατή η διασπορά του ολοκληρωμένου συστήματος κατά μήκος οποιουδήποτε δικτύου υπολογιστών, περιλαμβανομένου και του Internet. Αυτή η εργασία είναι σχετικά απλή, ωστόσο άπτεται των λειτουργικών προβλημάτων του συστήματος, όπως η διαχείριση λαθών και τα θέματα επιδόσεων, η αντιμετώπιση των οποίων είναι μια σημαντική πρόκληση σε μια πλατφόρμα με τόσο ελαστική αρχιτεκτονική και γενικευμένη δυνατότητα χρήσης.

Δυνατότητα βελτίωσης της αρχιτεκτονικής υπάρχει και ως προς την πλευρά των οντολογιών. Το interface προς τις οντολογίες είναι καλά ορισμένο και κατανοητό, ωστόσο η σχεδίαση του έχει γίνει με γνώμονα τις ανάγκες του Ozone και δεν διευκολύνει την υλοποίηση του από την πλευρά της οντολογίας. Η κατασκευή του interface απαιτεί κατανόηση της οντολογίας την οποία θα συνδέσει, γνώση αντικειμενοστραφούς προγραμματισμού, αλλά και κατανόηση του μοντέλου αναπαράστασης γνώσης που χρησιμοποιεί το Ozone. Μια κακή υλοποίηση μπορεί να επηρεάσει την απόδοση του συστήματος αλλά ακόμα και τη λειτουργική του πληρότητα, σε περίπτωση κάποιου λάθους λογικής.

Εδώ υπάρχει η δυνατότητα αξιοποίησης του OKBC [Chaudhri, 1998] για την απλοποίηση αυτού του έργου. Το OKBC δίνει τη δυνατότητα ανάπτυξης αντικειμενοστραφούς κώδικα που επικοινωνεί με μια οντολογία με βάση οντολογικούς όρους, δηλαδή το οντολογικό μοντέλο γνώσης και τον οντολογικό συμβολισμό. Έτσι μια γενική υλοποίηση του interface προς τις οντολογίες σε OKBC όπως αυτό περιγράφεται στην παράγραφο 6.3, θα μπορούσε να χρησιμεύσει ως διαμεσολαβητικό επίπεδο ανάμεσα στο Ozone και στις οντολογίες, στις οποίες θα προσέφερε ένα interface υλοποιημένο σε OKBC. Οι οντολογίες θα μπορούσαν να επικοινωνήσουν με αυτό το interface, υποθέτοντας ότι η οντολογική πλατφόρμα υποστηρίζει OKBC, υλοποιώντας το στην εγγενή γλώσσα και αναπαράσταση της οντολογίας. Έτσι η υλοποίηση του

interface δεν θα ενέπλεκε γνώσεις αντικειμενοστραφούς προγραμματισμού και θα απαιτούσε μόνο μια επιφανειακή αντίληψη του μοντέλου γνώσης του Ozone. Η εργασία προσάρτησης μιας οντολογίας στο Ozone στην περίπτωση αυτή θα περιοριζόταν σημαντικά. Ωστόσο, η γενική υλοποίηση σε OKBC του interface δεν είναι καθόλου απλή, δεδομένης της πολυπλοκότητας και διαφορετικότητας των διαφορετικών οντολογικών πλατφορμών, και θα αποτελούσε μια ενδιαφέρουσα ερευνητική κατεύθυνση.

Από μεθοδολογικής πλευράς, το Ozone παρουσιάζει ένα σημαντικό κενό στη διαχείριση συστημάτων μονάδων. Η διαχείριση μονάδων είναι ένα ιδιαίτερα περίπλοκο πρόβλημα σε τομείς εφαρμογών όπως το ηλεκτρονικό εμπόριο, όπου είναι αδύνατο να περιοριστεί κανείς σε φυσικά μεγέθη και συστήματα όπως το SI και το CGS. Τα συστήματα μέτρησης εμπορικών μεγεθών, όπως για παράδειγμα τα μεγέθη πουκαμίσων, παρουσιάζουν μεγάλη πολυπλοκότητα και δεν είναι από τη φύση τους μαθηματικοποιημένα. Στο Ozone αντιμετωπίστηκε το πρόβλημα της φορμαλιστικής περιγραφής μέσω της εισαγωγής κατάλληλων τύπων ιδιοτήτων και κατάλληλων περιορισμών στα σύνολα τιμών τους. Για παράδειγμα, τα ευρωπαϊκά μεγέθη πουκαμίσων χρησιμοποιούν μόνο άρτιους αριθμούς, ενώ τα μεγέθη παπουτσιών χρησιμοποιούν ακέραιους στο ευρωπαϊκό σύστημα, αλλά ημιακέραιους στο αμερικάνικο. Το Ozone υποστηρίζει τέτοιες ιδιόζουσες κλίμακες μέτρησης, όμως δεν αντιμετωπίζει το πρόβλημα της μετατροπής από μία κλίμακα σε άλλη. Το ζήτημα της μετατροπής, αλλά και γενικότερα η σε βάθος υποστήριξη συστημάτων μονάδων μέτρησης, απαιτεί περίπλοκους κανόνες και στα πλαίσια του Ozone ο καλύτερος τρόπος αντιμετώπισης του είναι πιθανότατα η χρήση εξειδικευμένων οντολογιών που θα κωδικοποιούν την απαραίτητη γνώση και θα προσφέρουν τις αντίστοιχες υπηρεσίες μετατροπής, περιγραφής και ελέγχου στην πλατφόρμα του Ozone. Αυτή η εργασία αποτελεί μια πολύ ενδιαφέρουσα εξέλιξη του συστήματος.

Τέλος, όπως έχει ήδη αναφερθεί η πειραματική εξέταση του Ozone σε περισσότερες περιοχές εφαρμογής είναι ιδιαίτερα σημαντική. Η παρούσα εργασία βασίστηκε στη δοκιμή του μόνο στα πλαίσια εφαρμογών ηλεκτρονικού εμπορίου. Η δοκιμή σε άλλες περιοχές εφαρμογής θα δώσει δύο επιθυμητά αποτελέσματα. Πρώτα, θα δώσει τη δυνατότητα επαύξησης, βελτίωσης και τυποποίησης του interface του Ozone προς την πλευρά των εφαρμογών. Δεύτερο και σημαντικότερο αποτέλεσμα είναι ο καλύτερος έλεγχος της συνολικής προδιαγραφής του Ozone, και η επαλήθευση της πληρότητας της. Στα πλαίσια του ηλεκτρονικού εμπορίου το Ozone έδειξε ότι έχει τη δυνατότητα να καλύψει τις ανάγκες των εφαρμογών για πρόσβαση σε μια ομοσπονδία ετερογενών πηγών δεδομένων, ωστόσο είναι πιθανό άλλες περιοχές εφαρμογής να φέρουν στην επιφάνεια επιπλέον ανάγκες.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [Aguado, 1998] Aguado, G.; Banon, A.; Bateman, J.; Bernardos, S.; Fernandez, M.; Gomez-Perez, A. Nieto, E.; Olalla, A.; Plaza, R.; Sached, A. *ONTOGENERATION: Reusing domain and linguistic ontologies for Spanish text generation*. Workshop on Applications of Ontologies and Problem-Solving Methods. European Conference on Artificial Intelligence (ECAI'98). Brighton (United Kingdom). 1998.
- [Alexander, 1986] Alexander, J. H., Freiling, M. J., Shulman, S. J., Staley, J. L., Rehfus, S., and Messick, S. L. 1986. Knowledge Level Engineering: Ontological Analysis, Proceedings of AAAI-86. Proceedings of the 5th National Conference on Artificial Intelligence, Los Altos: Morgan Kaufmann Publishers Philadelphia: 963-968.
- [Amann, 1999] Amann, B.; Fundulaki, I. Integrating Ontologies and Thesauri to Build RDF Schemas. 3rd European

- Conference on Research and Advanced Technology for Digital Libraries. ECAL'00. Paris (France). September, 1999.
- [Angele, 1998] Angele, J.; Fensel, D.; Studer, R. *Developing knowledge-based systems with mike*. Journal of Automated Software Engineering, 1998.
- [Arango, 1991] Arango, G, Williams, G., and Iscoe, N. 1991. Domain Modeling for Software Engineering. Proceedings of ICSE 1991: The International Conference on Software Engineering. ACM Press, Austin, Texas.
- [Armstrong, 1999] Armstrong R, Gannon D, Geist D, Keahey K, Kohn S, McInnes L, Parker S, and Smolinski B (1999). *Toward a Common Component Architecture for HighPerformance Scientific Computing*. In Proceedings of the 8th IEEE International Symposium on High Performance Distributed Computing, 1999.
- [Arpírez, 1998] Arpírez, J. C.; Gomez-Perez, A.; Lozano, A. Pinto, H. S. *(ONTO)²Agent: An ontology-based WWW broker to select ontologies*. Workshop on Applications of Ontologies and Problem-Solving Methods. European Conference on Artificial Intelligence (ECAI'98). Brighton (United Kingdom). 1998. Pp. 16-24.
- [Artale, 1996] Artale, A., Franconi, E., Guarino, N., and Pazzi, L. 1996. Part-Whole Relations in Object-Centered Systems: an Overview. Data and Knowledge Engineering, 20(3): 347-383.

- [Ashenhurst, 1996] Ashenhurst, R. L. 1996. Ontological Aspects of Information Modelling. *Minds and Machines*, 6: 287-394.
- [Aslan, 1999] Aslan G, McLeod D: Semantic Heterogeneity Resolution in Federated Databases by Metadata Implantation and Stepwise Evolution. In *The VLDB Journal* (1999) 8. Springer-Verlag, 1999.
- [Bateman, 1994] J. A. Bateman. KPML: The KOMET-Penman (Multilingual) Development Environment. Technical Report, GMD/IPSI, Darmstadt (Germany), 1994.
- [Bateman, 1995] Bateman, J. A. 1995. On the Relationship Between Ontology Construction and Natural Language: a Socio-Semiotic View. *International Journal of Human-Computer Studies*, 43: 929-944.
- [Batini, 1986] Batini C, Lenzerini M, Navathe S (1986) A comparative analysis of methodologies for database schema integration. *ACM Comput Surv* 18(4): 323–364 *Comput* 19(12): 10–18
- [Bayardo, 1997] Bayardo R J, Bohrer W, et. al., InfoSleuth: Agent-Based Semantic Integration of Information in Open and Dynamic Environments. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Vol. 26,2. ACM Press, New York, 1997.
- [Benjamins, 1998] Benjamins, V. R. and Fensel, D. 1998. The Ontological Engineering Initiative (KA)². In N. Guarino (ed.) *Formal Ontology in Information Systems*. IOS Press.
- [Benjamins, 1999] Benjamins, R.; Fensel, D.; Decker S.; Gomez-Perez A. *(KA)² Building Ontologies for the Internet: A mid term*

- report*. To be published in the International Journal of Human Computer Studies. 1999.
- [Bergamaschi, 1998] Bergamaschi, S., Castano, S., De Capitani di Vimercati, S., Montanari, S., and Vincini, M. 1998. An Intelligent Approach to Information Integration. In N. Guarino (ed.) *Formal Ontology in Information Systems*. IOS Press.
- [Bernaras, 1996] Bernaras, A.; Laresgoiti, I.; Corera, J. *Building and Reusing Ontologies for Electrical Network Applications*. Proceedings of the European Conference on Artificial Intelligence (ECAI'96). ECAI 96. 1996.
- [Bishop, 1995] C. M. Bishop, *Neural networks for pattern recognition*. Oxford: Clarendon Press, 1995.
- [Blazquez, 1998] Blazquez, M.; Fernandez-Lopez, M.; García-Pinar, J.M.; Gomez-Perez, A. *Building Ontologies at the Knowledge Level using the Ontology Design Environment*. Knowledge Acquisition of Knowledge-Based Systems Workshop (KAW). 1998.
- [Boag, 2006] Boag S, Chamberlin D, Fernandez M, Florescu D, Robie J, Simeon J. *XQuery 1.0: An XML Query Language*. W3C Candidate Recommendation. June 2006.
<http://www.w3.org/TR/xquery/>.
- [Borgo, 1996] Borgo, S.; Guarino, N.; Masolo, C. *Stratified Ontologies: the case of physical objects*. In proceedings of the Workshop on Ontological Engineering. Held in conjunction with ECAI96. Pages 5-15. Budapest. 1996.

- [Borst, 1997] Borst, W. N., Akkermans, J. M., and Top, J. L. 1997. Engineering Ontologies. *International Journal of Human-Computer Studies*, 46: 365-406.
- [Bray, 1998] Bray, T.; Paoli, J.; Sperberg, C.; Maler, E.; Yergeau, F. *Extensible Markup Language (XML) 1.0 (Fourth Edition)*. W3C Recommendation. August, 2006. <http://www.w3.org/TR/REC-xml> .
- [Brickley, 1999] Brickley, D.; Guha, R.V. *RDF Vocabulary Description Language 1.0: RDF Schema*. W3C Recommendation. February, 2004. <http://www.w3.org/TR/PR-rdf-schema> .
- [Brin, 1998] S. Brin and L. Page, "The Anatomy of a Large-Scale Hypertextual Web Search Engine", *Computer Networks and ISDN Systems*, vol. 30, no. 1-7, 1998, pp. 107-117.
- [Buneman, 1997] Peter Buneman. *Semistructured data*. In Proceedings of ACM PODS '97, pp. 117 – 121. Tucson, May 1997.
- [Burg, 1997] Burg, J. F. M. 1997. *Linguistic Instruments in Requirements Engineering*. IOS Press.
- [Casati, 1998] Casati, R., Smith, B., and Varzi, A. 1998. *Ontological Tools for Geographic Representation*. In N. Guarino (ed.) *Formal Ontology in Information Systems*. IOS Press.
- [Ceri, 1997] Ceri, S. and Fraternali, P. 1997. *Designing Database Applications with Objects and Rules: The IDEA Methodology*. Addison Wesley.
- [Chandrasekaran, 1992] Chandrasekaran, B.; Johnson, T. R.; Smith, J. W. *Task-structure analysis for knowledge modeling*. *Communications of the ACM*, 35(9). 1992. Pp. 124-137.

- [Chandrasekaran, 1999] Chandrasekaran, B.; Johnson, T. R.; Benjamins, V. R.
Ontologies: what are they? why do we need them?. IEEE
Intelligent Systems and Their Applications. 14(1).
Special Issue on Ontologies. 1999. Pp. 20-26.
- [Chaudhri, 1997] Chaudhri V. K.; Farquhar A.; Fikes R.; Karp P. D.; Rice J.
P. *The Generic Frame Protocol 2.0* , Technical Report,
Stanford University.1997.
- [Chaudhri, 1998] Chaudhri V. K.; Farquhar A.; Fikes R.; Karp P. D.; Rice J.
P. *OKBC: A Programmatic Foundation for Knowledge
Base Interoperability*. In Proceedings of the Fifteenth
National Conference on Artificial Intelligence (AAAI-98),
Madison, Wisconsin, pp 600-607. AAAI Press/The MIT
Press.
- [Chen, 1993] Chen J, Bukhres O, and Elmagarmid A K (1993). "IPL: A
multi-database transaction specification language," in
Proceedings of the 1993 International Conference on
Distributed Computing, 1993.
- [Chikofsky, 1990] Chikofsky, E.J.; Cross II, J.H. *Reverse Engineering and
design recovery: A taxonomy*. Software Magazine.
January 1990. PP:13-17.
- [Clark, 1999] Clark, J. *XSL Transformations (XSLT) Version 1.0*. W3C
Recommendation. November, 1999.
<http://www.w3.org/TR/xslt>.
- [Corcho, 2003] Corcho, O.; Gomez-Perez, A.; Leger, A.; Rey, C.;
Toumani, F. *An ontology-based mediation architecture
for e-commerce applications*. In Advances in Soft
Computing, Intelligent Information Processing and Web

- Mining. Proceedings of the International IIS: IIPWM '03 Conference. Springer-Verlag. pp: 477-488. June 2003
- [Czejdo, 1987] Czejdo B, Rusinkiewicz M, Embley D (1987) An approach to schema integration and query formulation in federated database systems. In: Proceedings of the 3rd IEEE Conference on Data Engineering, 1987, Los Angeles, CA. IEEE Comp Society, pp 477-484
- [Dahlgren, 1995] Dahlgren, K., "A Linguistic Ontology, International Journal of Human-Computer Studies, Vol.43, 1995, pp.809-818.
- [Davis, 1990] Davis, Ernest. 1990. Representations of Commonsense Knowledge. Los Altos: Morgan Kaufman.
- [Dayal & Hwang, 1984] Dayal U, Hwang H (1984) View definition and generalization for database integration in a multi-database system. IEEE Trans Software Eng 10(6): 628-644
- [Decker, 1999] Decker, S.; Erdmann, M.; Fensel, D.; Studer, R. *Ontobroker: Ontology Based Access to Distributed and Semi-Structured Information*. In R. Meersman (eds.): Semantic Issues in Multimedia Systems. Proceedings of DS-8. Kluwer Academic Publisher, Boston, 1999, 351-369.
- [Decker, 2000] S. Decker, D. Fensel, F. van Harmelen, I. Horrocks, S. Melnik, M. Klein, and J. Broekstra. Knowledge representation on the web. In Proceedings of the 2000 International Workshop on Description Logics (DL2000), Aachen, Germany, 2000.

- [Domingue, 1998] Domingue, J. *Tadzebao and Webonto: Discussing, Browsing and Editing Ontologies on the Web*. In Proceedings of the Eleventh Knowledge Acquisition Workshop, KAW98, Banff, 1998.
- [Ellis, 2001] Ellis J, Ho L. *JDBC 3.0 Specification*. Sun Microsystems, October 2001. <http://jcp.org/en/jsr/detail?id=54>.
- [Elmasri & Navathe, 1984] Elmasri R, Navathe S (1984) Object integration in logical database design. In: Proceedings of IEEE Computer Society 1st International Conference on Data Engineering, 1984, Los Angeles, CA. IEEE Comp Society, 1984.
- [Erdmann, 1999] Erdmann, M.; Studer, R. *Ontologies as Conceptual Models for XML Documents*. Proceedings of the 12th Knowledge Acquisition for Knowledge-Based Systems Workshop (KAW'99), Banff, Canada, October 1999.
- [Euzenat, 1996] Euzenat, J. Corporative memory through cooperative creation of knowledge bases and hyper-documents. Proceedings 10th KAW, Banff (Canada). 1996.
- [Farquhar, 1996] Farquhar A., Fikes R., Rice J., *The Ontolingua Server: A Tool for Collaborative Ontology Construction*, Proceedings of the 10th Knowledge Acquisition for Knowledge-Based Systems Workshop, Banff, Alberta, Canada, PP. 44.1-44.19, 1996.
- [Fellbaum, 1998] Fellbaum, C. *WordNet: An Electronic Lexical Database*. Bradford Books, 1998. ISBN 0-262-06197-X.
- [Fensel, 1998] Fensel D, Stefan Decker, Michael Erdmann, and Rudi Studer. Ontobroker: The Very High Idea. In Proceedings

- of the 11th International Flairs Conference (FLAIRS-98), Sanibal Island, Florida, May 1998.
- [Fernandez Lopez, 1999] Fernandez-Lopez, M.; Gomez-Perez, A.; Pazos-Sierra, A.; Pazos-Sierra, J. *Building a Chemical Ontology Using Methontology and the Ontology Design Environment*. IEEE Intelligent Systems & their applications. January/February 1999. PP. 37-46.
- [Fernandez, 1997] Fernandez, M.; Gomez-Perez, A.; Juristo, N. *METHONTOLOGY: From Ontological Art Towards Ontological Engineering*. Symposium on Ontological Engineering of AAAI. Stanford (California). March 1997.
- [FIPA, 2000] Agent Communication Language, FIPA 2000 Specification. Available at <http://www.FIPA.org> (current Mar. 2003).
- [Frank, 1997] Frank, A.U. 1997. Spatial Ontology: A Geographical Point of View. In *Spatial and Temporal Reasoning*. (Stock, O., ed.), Dordrecht, The Netherlands, Kluwer Academic Publishers, pp: 135-153.
- [Gaines, 1997] Gaines, B. 1997. Editorial: Using Explicit Ontologies in Knowledge-based System Development. *International Journal of Human-Computer Systems*, 46: 181.
- [Gangemi, 1998] Gangemi, A., Pisanelli, D., and Steve, G. 1998. Ontology Alignment: Experiences With Medical Terminologies. In N. Guarino (ed.) *Formal Ontology in Information Systems*. IOS Press.
- [Genesereth, 1992] Genesereth, M. R.; Fikes, R. E. *Knowledge Interchange Format. Version 3.0. Reference Manual*. Computer

- Science Department. Stanford University. Stanford, California. 1992.
- [Gomez Perez, 1995] Gomez-Perez, A.; Juristo, N.; Pazos, J. *Evaluation and assessment of knowledge sharing technology*. In N. J. Mars (editor), *Towards Very Large Knowledge Bases*. KBKS 95. IOS Press, Amsterdam, 1995. pp., 289-296.
- [Gomez Perez, 1996] Gomez Perez, A.; Fernandez-Lopez, M.; de Vicente, A. *Towards a method to conceptualize domain ontologies*. Workshop on Ontological Engineering. European Conference of Artificial Intelligence (ECAI). 1996. Budapest (Hungary) Pp. 41-52.
- [Gomez Perez, 1998] Gomez Perez, A. *Knowledge Sharing and Reuse*. In J. Liebowitz (Editor) *Handbook of Expert Systems*. CRC. 1998.
- [Gomez Perez, 1999] Gomez-Perez, A; Rojas, M. D. *Ontological Reengineering and Reuse*. European Knowledge Acquisition Workshop (EKAW). 1999.
- [Gong, 1999] L. Gong, *Inside Java 2 Platform Security*, Addison-Wesley, 1999.
- [Google, 2006] <http://www.google.com/> (current Jul. 2006).
- [Grassl, 1999] Grassl, Wolfgang. 1999. *The Reality of Brands: Towards an Ontology of Marketing*, in Koepsell (ed.), 313-359.
- [Grosso, 2001] Grosso P, Gutentag E, Milowski R A, Richman J, Parnell S, Zilles S, Berglund A, Adler S, Deach S, Caruso J, Graham T. *Extensible Stylesheet Language (XSL) Version 1.0*. W3C Recommendation. October 2001. <http://www.w3.org/TR/xsl/>.

- [Gruber, 1993] Gruber, T. R. 1993. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5: 199-220.
- [Gruber, 1995] Gruber, T. R. 1995. Toward Principles for the Design of Ontologies Used for Knowledge Sharing. *International Journal of Human and Computer Studies*, 43(5/6): 907-928.
- [Gruninger, 1995] Gruninger, M. and Fox, M. S. 1995. The Logic of Enterprise Modelling. In J. Brown and D. O' Sullivan (eds.), *Reengineering the Enterprise*. Chapman and Hall.
- [Grüninger, 1995] Grüninger, M.; Fox, M. S. *Methodology for the Design and Evaluation of Ontologies*. Proceedings of IJCAI95's Workshop on Basic Ontological Issues in Knowledge Sharing. 1995.
- [Guarino, 1995] Guarino, N. 1995. Formal Ontology, Conceptual Analysis and Knowledge Representation. *International Journal of Human and Computer Studies*, 43(5/6): 625-640.
- [Guarino, 1997] Guarino, N. 1997. Semantic Matching: Formal Ontological Distinctions for Information Organization, Extraction, and Integration. In M. T. Pazienza (ed.) *Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology*. Springer Verlag: 139-170.
- [Guarino, 1998] Guarino, Nicola (ed.) 1998. *Formal Ontology in Information Systems*, Amsterdam, Berlin, Oxford: IOS Press. Tokyo, Washington, DC: IOS Press (Frontiers in Artificial Intelligence and Applications), 1998.

- [Guarino, 2000] Guarino, N. and Welty, C. 2000. A Formal Ontology of Properties, in R. Dieng and O. Corby (eds.), Knowledge Engineering and Knowledge Management: Methods, Models and Tools. 12th International Conference (EKAW 2000), Berlin/New York: Springer: 97–112.
- [Guha, 1989] Guha, R. V.; Lenat, D. B. *Cyc: The world according to cyc, part 2: Agents and institutions*. Technical Report ACT-CYC-453-89, MCC. December, 1989.
- [Hammer & McLeod, 1996] Hammer J, McLeod D (1993) An approach to resolving semantic heterogeneity in a federation of autonomous, heterogeneous database systems. *Int J Intelligent Coop Inf Syst* 2(1): 51–83
- [Harmelen, 1999] Harmelen, F.; Fensel, D. *Surveying notations for machine-processable semantics of Web sources*. Proceedings of the IJCAI'99 Workshop on Ontologies and Problem-Solving Methods. 1999.
- [Hayes, 1985] Hayes, Patrick J. 1985. The Second Naive Physics Manifesto, in Hobbs and Moore (eds.), 1–36.
- [Haykin, 1999] S. Haykin, *Neural Networks: A comprehensive foundation*, 2nd edn. Upper Saddle River, NJ: Prentice-Hall, 1999.
- [Hendler, 2001] J. Hendler, "Agents and the Semantic Web," *IEEE Intelligent Systems*, vol. 16, no. 2, Mar./Apr. 2001, pp. 30-37.
- [Hoppenbrouwers, 1996] Hoppenbrouwers, J., van der Vos, B., and Hoppenbrouwers, S. 1996. NL Structures and Conceptual Modelling: the KISS case. In R. P. van de Riet, J. F. M.

- Burg and A. J. van der Vos (eds.), *Applications of Natural Language to Information Systems*. IOS Press.
- [Horrocks; 00] Horrocks, I., Fensel, D., Harmelen, F., Decker, S., Erdmann, M, Klein, M. *OIL in a Nutshell*. ECAI'00 Workshop on Applications of Ontologies and Problem-Solving Methods. Berlín (Germany). August, 2000.
- [Huber, 2002] M. J. Huber, *Jam Agents in a Nutshell*. Available at <http://members.home.net:80/marcush/IRS/> (current Dec. 2002).
- [IEEE, 1990] *IEEE Standard Glossary of Software Engineering Terminology*. IEEE Computer Society. New York (USA). 1990.
- [IEEE, 1996] *IEEE Standard for Developing Software Life Cycle Processes*. IEEE Computer Society. New York (USA). April 26, 1996.
- [IKV, 2003] http://www.ikv.de/content/Produkte/osp_e.htm (current Mar. 2003).
- [Jacobs, 1996] Jacobs N, Shea R. The role of Java in InfoSleuth: Agent-based exploitation of heterogeneous information resources. In *Proceeding of Intranet-96 Java Developers Conference*, April 1996.
- [Jardine, 1997] Jardine, Donald A. (ed.) 1997. *The ANSI/SPARC DBMS Model: Proceedings of the Second SHARE Working Conference on Data Base Management Systems*, Montreal, Canada, April 26–30, 1976, Amsterdam/New York : North-Holland.

- [Jennings, 1998] N. R. Jennings, K. Sycara, M. Wooldridge, "A Roadmap of Agent Research and Development", *Autonomous Agents and Multi-Agent Systems*, vol. 1, no. 1, 1998, pp. 7-38.
- [Johansson, 1989] Johansson, Ingvar. 1989. *Ontological Investigations. An Inquiry into the Categories of Nature, Man and Society*, New York and London: Routledge.
- [KACTUS, 1996] *The KACTUS Booklet version 1.0*. Esprit Project 8145. September, 1996.
- [Kahng & McLeod, 1996] Kahng J, McLeod D (1996) Dynamic classificational ontologies for discovery in cooperative federated databases. In: *Proceedings of the 1st International Conference on Cooperative Information Systems*, June 1996, Brussels, Belgium. IEEE-CS Press
- [Karp, 1995] Karp, R.; Myers, K.; Gruber. T. The generic frame protocol. *Proceedings of the 1995 International Joint Conference on Artificial Intelligence*. 1995.
<ftp://ftp.ai.sri.com/pub/papers/karp-gfp95.ps.Z>
- [Karp, 1999] Karp, R.; Chaudhri, V.; Thomere, J. *XOL: An XML-Based Ontology Exchange Language*. Technical Report. July, 1999. <http://www.ai.sri.com/~pkarp/xd>.
- [Kasper, 2003] Walter Kasper, Katarina Stanoevska-Slabeva, Yiannis Stavroulas, Angelos Yannopoulos. *Cross-lingual Information Transformation and Summarization for Portable End-Devices with the MEMPHIS Toolbox*. In *Proceedings of the 3rd Workshop on Applications and*

-
- Services in Wireless Networks, Berne, Switzerland, July 2-4 2003. ISBN: 3-9522719-0-X.
- [Kenny, 2002] S. Kenny and L. Korba, "Applying digital rights management systems to privacy rights management", *Computers & Security*, vol. 21, no. 7, 2002, pp. 648-664.
- [Kent, 1998] Kent, R. Conceptual Knowledge Markup Language (version 0.2). 1998.
<http://wave.eecs.wsu.edu/CKRMI/CKML.html>.
- [Kifer, 1995] Kifer, M.; Lausen, G.; Wu, J. *Logical Foundations of Object-Oriented and Frame-Based Languages*. Journal of the ACM. 1995.
- [Knight, 1994] Knight, K.; Luck S. *Building a Large Knowledge Base for Machine Translation*. Proceedings of the American Association of Artificial Intelligence Conference (AAAI-94). Seattle (USA). 1994.
- [Knight, 1995] Knight, K.; Chancer, I.; Haines, M.; Hatzivassiloglou. V.; Hovy, E. H.; Iida M.; Luk, S.K.; Whitney, R.A.; Yamada, K. *Filling Knowledge Gaps in a Broad-Coverage MT System*. Proceedings of the 14th IJCAI Conference. Montreal (Canada). 1995.
- [Koepsell, 2000] Koepsell, David R. 2000. *The Ontology of Cyberspace: Law, Philosophy, and the Future of Intellectual Property*, Chicago: Open Court.
- [Koh & Chen, 1993] Koh JL, Chen ALP (1993) Integration of heterogeneous object schemas. In: Elmasri R, Kouramojian V, Thalheim B (eds) Proceedings of the 12th International Conference

- on Entity Relationship Approach, 1993, Lecture notes in CS, Vol 823, Springer, pp 297–314
- [Krishnamurthy, 1991] Krishnamurthy R, Litwin W, Kent W (1991) Language features for IEEE interoperability of databases with schematic discrepancies. In: Clifford J, King R (eds) Proceedings of ACM SIGMOD International Conference on Management of Data, 1991, Denver Colo. SIGMOD Record 20(2): 40–49
- [Kwok, 2000] S.H. Kwok et. al., "Digital rights management in Internet open trading protocol (IOTP)", *Proc. Int'l Conf. Electronic Commerce (ICEC 2000)*. 2000, pp. 179-185.
- [Ladroy, 2002] E. A. Ladroy, "Securing electronic text against optical character recognition", *Proc. 2nd WOSAT Int'l Conf. Image Proc (WOSAT'02)*., 2002, pp.70-91.
- [Lang, 1991] Lang, E. 1991. The LILOG Ontology from a Linguistic Point of View. In O. Herzog and C. R. Rollinger (eds.), Text Understanding in LILOG. Springer-Verlag, Berlin.
- [Lassila, 1999] [Lassila, O.; Swick, R. Resource Description Framework \(RDF\) Model and Syntax Specification. W3C Proposed Recommendation. January, 99.](http://www.w3.org/TR/PR-rdf-syntax)
[http://www.w3.org/TR/PR-rdf-syntax.](http://www.w3.org/TR/PR-rdf-syntax)
- [Lehmann, 1995] Lehmann, F. 1995. Machine-Negotiated, Ontology-Based EDI (Electronic Data Interchange). In Proceedings of CIKM-94 Workshop on Electronic Commerce, Springer Verlag.

- [Lenat, 1990] Lenat, D.B.; Guha, R.V. *Building large knowledge-based systems*. Addison-Wesley Publising Company, Inc. 1990.
- [Levy, 1998] Levy, A.; Rousset, M.C. *CARIN: A Representation Language Combining Horn rules and Description Logics*. Artificial Intelligence Journal, vol 104. September 1998.
- [Lindholm, 2001] T. Lindholm and F. Yellin, *The Java virtual machine Specification*. Addison-Wesley, 2001.
- [Litwin, 1990] Litwin W, Mark L, Roussopoulos N (1990) Interoperability of multiple autonomous databases. ACM Comput Surv 22(3): 267–293
- [Luke, 2000] Luke S.; Heflin J. SHOE 1.01. Proposed Specification. SHOE Project. February, 2000.
<http://www.cs.umd.edu/projects/plus/SHOE/spec1.01.htm>.
- [MacGregor, 1991] MacGregor, R. *Inside the LOOM clasifier*. SIGART bulletin. #2(3):70-76. June, 1991.
- [Mahesh, 1996] Mahesh, K. 1996. Ontology Development for Machine Translation: Ideology and Methodology. New Mexico State University, Computing Research Laboratory MCCA-96-292.
- [Marcus, 1988] Marcus, S, editor. *Automating knowledge acquisition for expert systems*. Kluwer, Boston (USA). 1988.
- [McCarthy, 1980] McCarthy, J. 1980. Circumscription – A Form of Non-Monotonic Reasoning, Artificial Intelligence, 5: 13, 27–39.

- [McEntire, 1999] McEntire, R.; Karp, P.; Abernethy, N.; Olken, F.; Kent, R.; DeJongh, M.; Tarczy-Hornoch, P.; Benton, D.; Pathak, D.; Helt, G.; Lewis, S.; Kosky, A.; Neumann, E.; Hodnett, D.; Tolda, L.; Topaloglou, T. *An Evaluation of Ontology Exchange Languages for Bioinformatics*. Technical Report. August, 1999.
- [McGuinness, 1998] McGuinness, D. 1998. Ontological Issues for Knowledge-Enhanced Search. In N. Guarino (ed.) *Formal Ontology in Information Systems*. IOS Press.
- [McIlraith, 2001] S. McIlraith, T. Son, and H. Zeng. Semantic Web services. In *IEEE Intelligent Systems (Special Issue on the Semantic Web)*, March/April 2001.
- [MEMPHIS, 2000] MEMPHIS Technical Annex. Project Reference IST-2000-25045.
- [MEMPHIS, 2003] <http://www.ist-memphis.org/> (current Jul. 2003).
- [Mena, 1998] Mena, E., Kashyap, V., Illarramendi, A., and Sheth, A. 1998. Domain Specific Ontologies for Semantic Information Brokering on the Global Information Infrastructure. In N. Guarino (ed.) *Formal Ontology in Information Systems*. IOS Press.
- [Mena, 2000] Mena E, Illarramendi A, Vipul K, Sheth A. "OBSERVER : An Approach for Query Processing in Global Information Systems Based on Interoperation Across Pre-existing Ontologies". In *Distributed and Parallel Databases*, 8. Kluwer Academic Publishers, 2000.
- [Mertz, 1996] Mertz, D. W. 1996. *Moderate Realism and Its Logic*, New Haven, CN: Yale University Press.

- [Miller, 1995] Miller, G. A. 1995. WORDNET: A Lexical Database for English. *Communications of ACM*(11): 39- 41.
- [MKBEEM, 2006] The Official MKBEEM Project Site.
<http://www.mkbeem.com/>.
- [Motta, 1999] Motta, E. *Reusable Components for Knowledge Modelling*. IOS Press. Amsterdam. 1999.
- [Musen, 1993] Musen, M. *An overview of knowledge acquisition*. In David J. M.; Krivine, J. P.; Simmons, R. editors, *Second Generation Expert Systems*. Springer Verlag. 1993.
- [Navratil, 1998] Navratil, G. 1998. An Object-Oriented Approach to a Model of a Cadaster, Department of Geoinformation, Technical University of Vienna, Vienna.
- [Neches, 1991] Neches R.; Fikes, R.E.; Finin, T.; Gruber, T.R.; Senator T.; Swartout; W.R.. *Enabling technology for knowledge sharing*. *AI Magazine*. 12(3)::36-56- 1991.
- [Nirenburg, 1991] Nirenburg, S. *Machine Translation: A Knowledge-Based Approach*. Morgan Kaufmann, 1991. ISBN: 1558601287.
- [Nwana, 1999] H. Nwana, D. Ndumu, "A Perspective on Software Agents Research", *The Knowledge Engineering Review*, Cambridge University Press, vol. 14, no. 2, 1999, pp 1-18.
- [Omelayenko, 2001] Omelayenko B., *Syntactic-Level Ontology Integration Rules for E-commerce*. In: *Proceedings of The 14th International FLAIRS Conference (FLAIRS-2001)*, AAAI Press, Key West, FL, May 21-23, 2001.

- [OMG, 2003] Object Management Group. *OMG Unified Modeling Language (UML), Version 1.5*, March 2003.
<http://www.omg.org/technology/documents/formal/uml.htm>.
- [Partridge, 1996] Partridge, C. 1996. *Business Objects: Re-Engineering for Reuse*. Butterworth-Heinemann, Oxford.
- [Pazzi, 1998] Pazzi, L. 1998. Three Points of View in the Characterization of Complex Entities. In N. Guarino (ed.) *Formal Ontology in Information Systems*. IOS Press.
- [Pitoura, 1997] Pitoura E (1997) Providing Database Interoperability through Object-Oriented Language Constructs. In: *Journal of Systems Integration*, 7, 99 – 126. Kluwer, Boston, 1997.
- [Poli, 1996] Poli, R. 1996. Ontology and Knowledge Organization. In *Proceedings of 4th Conference of the International Society of Knowledge Organization (ISKO 96)*. Washington.
- [Quine, 1969] Quine, W. V. O. 1969. *Ontological Relativity, and Other Essays*, New York: Columbia University Press.
- [Rosenblatt , 2001] B. Rosenblatt, B. Trippe, and S. Mooney, *Digital Rights Management: Business and Technology*, New York: John Wiley & Sons, 2001.
- [Samuelson, 1999] P. Samuelson, "Good News and Bad News on the Intellectual Property Front", *Communications of the ACM*, vol. 42, no. 3, 1999, pp. 335-342.

- [Sattler, 2004] Sattler, Kai-Uwe; Geist, Ingolf; Schallehn, Eike.
Concept-Based Querying in Mediator Systems. VLDB Journal, Vol. 13, Issue 3. Springer-Verlag, Hamburg, 2004.
- [Schreiber, 1999] Schreiber, G.; Akkermans, H.; Anjewierden, A.; Hoog, R.; Shadbolt, N.; Van de Velde, W.; Wielinga, B.
Knowledge engineering and management. The CommonKADS Methodology. MIT press, Massachussets. 1999.
- [Sciencedirect, 2006] <http://www.sciencedirect.com/> (current Jul. 2006).
- [Smith, 2001] Barry Smith, Christopher A. Welty: FOIS Introduction: Ontology - Towards a New Synthesis. Proceedings of the 2nd International Conference on Formal Ontology in Information Systems, FOIS 2001, Ogunquit, Maine, USA, October 17-19 2001. ACM, 2001.
- [Sowa, 1984] Sowa, John F. 1984. *Conceptual Structures*. Information Processing in Mind and Machine. Reading, MA: Addison Wesley.
- [Staab, 00] Staab, S.; Erdmann, M., Maedche, A. *An extensible approach for Modeling Ontologies in RDF(S)*. In Proceedings of the ECAI00 Workshop on Applications of Ontologies and Problem Solving Methods. Berlin. Germany. August, 2000.
- [Staab, 1999] S. Staab, C. Braun, I. Bruder, A. Düsterhöft, A. Heuer, M. Klettke, G. Neumann, B. Prager, J. Pretzel, H.-P. Schnurr, R. Studer, H. Uszkoreit, & B. Wrenger[1999b]. GETESS - Searching the Web Exploiting German Texts.

- CIA '99 - Proceedings of the 3rd international Workshop on Cooperating Information Agents. Upsala, Sweden, 1999, LNCS, Berlin, Heidelberg: Springer.
- [Stavroulas , 2000] Y. Stavroulas Design and Implementation of a Greek Text-to-Speech System Based on Concatenative Synthesis. Proceedings of the 6th International Conference on Spoken Language Processing, Beijing, China, October 16 – 20, 2000.
- [Stavroulas, 2002] Yiannis Stavroulas, Theodora Varvarigou, Yiannis Kouroupis, Katerina Tsiara. Ozone: An Insulating Layer Between Ontologies, Databases And Object-Oriented Applications. Proceedings of the The Xth European Conference on Information Systems (ECIS2002), Gdansk, Poland, June 6-8 2002. Gdansk University Press, Gdansk.
- [Stavroulas, 2002a] Yiannis Stavroulas and Theodora Varvarigou. A Technique for Semi-automatic Database Integration in Database Federations. Proceedings of the IASTED International Conference on Information Systems and Databases (ISDB 2002), Tokyo, Japan, 25-27 September 2002.
- [Steels, 1990] Steels, L. *Components of expertise*. AI Magazine, 11(2). Summer 1990. Pp. 547-575.
- [Studer, 1998] Studer, R.; Benjamins, V.R.; Fensel, D. *Knowledge Engineering: Principles and Methods*. Data & Knowledge Engineering. 25: 161-197. 1998.

- [Sun, 2001] Sun Microsystems, Security and signed applet, 2001. Available at <http://jsp2.java.sun.com/products/plugin/1.3/docs/netscape.html>.
- [Sun, 2006] <http://java.sun.com/security/> (current Mar. 2006)
- [Swartout, 1995] Swartout, W.; Gil, Y. *Expect: Explicit representations for flexible acquisition*. In Proceedings of the Ninth Knowledge Acquisition for Knowledge-Based Systems Workshop. 1995.
- [Swartout, 1997] Swartout, B.; Ramesh P.; Knight, K.; Russ, T. *Toward Distributed Use of Large-Scale Ontologies*. Symposium on Ontological Engineering of AAI. Stanford (California). Mars, 1997.
- [Terpstra, 1993] Terpstra, P.; van Heist, G.; Wielinga, B.; Shadbolt, N. *Knowledge acquisition support through generalised directive models*. In Jean-Marc David, Jean-Paul Krivine, and Reid Simmons, editors, *Second Generation Expert Systems*. Springer Verlag. Berlin Heidelberg (Germany). 1993. Pp. 428-455.
- [Thompson, 1999] Thompson, H.; Beech, D.; Maloney, M.; Mendelsohn, N. XML Schema Part 1: Structures. W3C working draft. December, 1999. <http://www.w3.org/TR/xmlschema-1/>
- [Tsiara, 2002] Katerina Tsiara, Yiannis Kouroupis, Theodora Varvarigou, Yiannis Stavroulas. *Addressing the Challenges of International e-Commerce Using Agent Technologies*. In Proceedings of the 8th International Conference on Distributed Multimedia Systems, San Francisco, USA, 26-28 September 2002, ISBN: 189170611X.

- [Ullrich, 2001] Ullrich H , Puroo S, Storey V. An Ontology for Classifying the Semantics of Relationships in Database Design. In: M. Bouzeghoub. (Eds.): NLDB 2000, LNCS 1959, pp. 91-102, 2001. Springer-Verlag, 2001.
- [Uschold, 1995] Uschold, M. King, M. *Towards a Methodology for Building Ontologies*. Workshop on Basic Ontological Issues in Knowledge Sharing. 1995.
- [Uschold, 1996] Uschold, M. and Gruninger, M. 1996. Ontologies: Principles, Methods and Applications. *The Knowledge Engineering Review*, 11(2): 93-136.
- [Valente, 1999] Valente, A.; Russ, T.; McGregor, R.; Swartout, W. *Building and (Re)Using an Ontology of Air Campaign Planning*. IEEE Intelligent Systems & their applications. January/February 1999.
- [Van de Riet, 1998] Van de Riet, R., Burg, H., and Dehne, F. 1998. Linguistic Issues in Information Systems Design. In N. Guarino (ed.) *Formal Ontology in Information Systems*. IOS Press.
- [Van Heijst, 1997] Van Heijst, G., Schreiber, A. T., and Wielinga, B. J. 1997. Using Explicit Ontologies in KBS Development. *International Journal of Human and Computer Studies*, 46: 183-292.
- [Wagner, 1998] G. Wagner, *Foundations of Knowledge Systems with Applications to Databases and Agents*, New York: Kluwer Academic Publishers, 1998.
- [Wagner, 1999] G. Wagner, "Towards Agent-Oriented Information Systems", *Technical Report*, Freie Universität Berlin,

1999. Available at <http://www.inf.fu-berlin.de/~wagnerg/index.html>
- [Walmsley, 2004] Walmsley P, Fallside D. *XML Schema Part 0: Primer Second Edition*. W3C Recommendation. October, 2004. <http://www.w3.org/TR/xmlschema-0/>.
- [Wand & Weber, 1990] Wand, Y., and Weber, R.. An Ontological Model of an Information System. *IEEE Transactions on Software Engineering*, Vol. 16, 1990, pp. 1282-1292.
- [Wand, 1989] Wand, Y. 1989. A Proposal for a Formal Model of Objects. In W. Kim and F. H. Lochovsky (eds.), *Object-Oriented Concepts, Databases, and Applications*. Addison Wesley, Reading, MA: 537-559.
- [Welty, 1999] Welty, C., Lehmann, F., Gruninger, G., and Uschold, M. 1999. *Ontology: Expert Systems All Over Again?* Invited panel at AAAI-99: The National Conference on Artificial Intelligence. Austin, Texas.
- [Wiederhold, 1994] Wiederhold G (1994) Interoperation, mediation and ontologies. In: *Workshop on Heterogeneous Knowledge-Bases, 1994*. W3, pp 33-48.
- [Wiederhold, 1996] Wiederhold, G. (ed.) 1996. *Intelligent Integration of Information*. Kluwer Academic Publishers, Boston, MA.
- [Wielinga, 1992] Wielinga, B. J.; Schreiber, A. T.; Breuker, J. A. *KADS: a modeling approach to knowledge engineering*. *Knowledge Acquisition* (4). PP. 5-53. 1992.
- [Yahoo, 2003] http://edit.yahoo.com/config/eval_register/ (current Feb. 2003).

- [Yannopoulos, 2002] A. Yannopoulos, Y. Stavroulas, N. Papadakis, D. Halkos, T. Varvargou: A Method Which Enables the Assessment of Private Data by an Untrusted Third Party Using Arbitrary Algorithms But Prevents Disclosure of Their Content. In Hamid R. Arabnia, Youngsong Mun (Eds.): Proceedings of the International Conference on Internet Computing (IC'2002), Las Vegas, Nevada, USA, June 24-27, 2002. CSREA Press, 2002, ISBN 1-892512-38-6 - Volume 2.
- [Yannopoulos, 2002a] A. Yannopoulos, Y. Stavroulas, N. Papadakis, D. Halkos, T. Varvarigou "An architecture to search private data using arbitrary assessment algorithms without disclosing their content", WSEAS Transactions on Communications, Rethymno, pp. 213-220, 2002.
- [Yannopoulos, 2004] Angelos Yannopoulos, Yiannis Stavroulas, Theodora Varvarigou. Moving e-commerce with PIVOTS: Private Information Viewing Offering Total Safety. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 16, No. 6, June 2004.