

ΤΕΙ ΠΑΤΡΑΣ

ΣΧΟΛΗ ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΟΙΚΟΝΟΜΙΑΣ

*ΤΜΗΜΑ ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΥ ΣΧΕΔΙΑΣΜΟΥ ΚΑΙ ΠΛΗΡΟΦΟΡΙΑΚΩΝ
ΣΥΣΤΗΜΑΤΩΝ*

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

«Διερεύνηση των δυνατοτήτων του προτύπου OWL-S και της εφαρμογής του για την σημασιολογική περιγραφή υπηρεσιών Ιστού»

Αντωνίου Χαράλαμπος
Ντερτιλή Σύλβια

Επιβλέπων: Μπεληγιάννης Γεώργιος

Ευχαριστίες

Ευχαριστούμε θερμά :

- Τον επιβλέποντα καθηγητή μας, Κο Μπεληγιάννη Γρηγόριο για την υποστήριξη του, τις συμβουλές του και την άρτια επιστημονική καθοδήγηση που μας προσέφερε.
- Τις οικογένειες μας για την κατανόησή τους και τους φίλους μας για την αγάπη και υποστήριξή τους.

Περιεχόμενα

1	ΚΕΦΑΛΑΙΟ: ΥΠΗΡΕΣΙΕΣ ΙΣΤΟΥ ΚΑΙ ΣΗΜΑΣΙΟΛΟΓΙΚΟΣ ΙΣΤΟΣ	11
1.1	Εισαγωγή Στις Υπηρεσίες Ιστού	11
1.1.1	Εισαγωγή	11
1.1.2	Η Πλατφόρμα των Υπηρεσιών Δικτύου	13
1.1.3	SOAP	14
1.1.4	UDDI (Universal Description, Discovery and Integration Service).....	15
1.1.5	Παράδειγμα στο UDDI	17
1.1.6	WSDL (Web Services Definition Language).....	18
1.1.7	XLANG	23
1.1.8	XAML (Transaction Authority Markup Language).....	24
1.1.9	XKMS (XML Key Management Specification)	25
1.1.10	Άλλες Προτάσεις	26
1.2	Ο Σημασιολογικός Ιστός	27
1.2.1	Εισαγωγή – Παράδειγμα	27
1.2.2	Το Νόημα	28
1.2.3	Αναπαράσταση της Γνώσης	29
1.2.4	Οντολογίες	32
1.2.5	Πράκτορες.....	35
1.2.6	Η Εξέλιξη της Γνώσης.....	38
2	ΚΕΦΑΛΑΙΟ: ΣΗΜΑΣΙΟΛΟΓΙΚΕΣ ΥΠΗΡΕΣΙΕΣ ΙΣΤΟΥ.....	39
2.1	Σημασιολογικές Υπηρεσίες Δικτύου: Περιγραφή – Απαιτήσεις -Σύγχρονες Τεχνολογίες.....	39
2.1.1	Εισαγωγή	39
2.1.2	Δυνατότητες και Περιγραφικές Απαιτήσεις της Εύρεσης και της Σύνθεσης.....	41
2.1.3	Περιγραφή των Δυνατοτήτων	43
2.1.4	Βελτιώσεις των Δυνατοτήτων.....	45
2.1.5	Η Σχέση με τις Σημερινές Τεχνολογίες.....	49
2.1.6	Διαλειτουργικότητα και οι Σημερινές Τεχνολογίες.....	50
2.1.7	Επικλήση μιας Υπηρεσίας	53
2.1.8	Επανόρθωση και άλλες Απαιτήσεις.....	54
2.2	Εισαγωγή Σημασιολογικών Στοιχείων Στις Υπηρεσίες Δικτύου.....	56
2.2.1	Εισαγωγή	56
2.2.2	Παράδειγμα.....	57
2.2.3	Εισαγωγή στα Σημασιολογικά Στοιχεία.....	59
2.2.4	Η Θέσπιση	61

2.2.5	Η Εύρεση	63
2.2.6	Αναπαράσταση Δυνατοτήτων	64
2.2.7	Ταυτοποίηση Ικανοτήτων	66
2.2.8	Η Σχέση με τη UDDI	67
2.2.9	Η Σύνθεση	69
2.2.10	Σχετικές Εργασίες	74
2.2.11	Σύνοψη	75
3	ΚΕΦΑΛΑΙΟ: ΤΟ ΠΡΟΤΥΠΟ OWL-S	77
3.1	OWL-S: Σημασιολογική Σήμανση Για Τις Υπηρεσίες Δικτύου.	77
3.1.1	Εισαγωγή	77
3.1.2	Οι Υπηρεσίες στον Σημασιολογικό Ιστό	77
3.1.3	Παραδείγματα	78
3.1.4	Μια Ανώτερη Οντολογία Για Τις Υπηρεσίες	81
3.1.5	Προφίλ Υπηρεσιών	84
3.1.5.1	Συνθέτοντας το Προφίλ: Η Σχέση με το Μοντέλο Διαδικασίας	86
3.1.5.2	Οι Ιδιότητες του Προφίλ	88
3.1.5.3	Παράμετροι της Υπηρεσίας	92
3.1.5.4	Κατηγορία Υπηρεσίας	92
3.1.5.5	Κατηγοριοποιώντας τον Τύπο της Υπηρεσίας και το Προϊόν	93
3.1.6	Μοντελοποιώντας Υπηρεσίες ως Διαδικασίες	93
3.1.6.1	Παράμετροι και Εκφράσεις	94
3.1.6.2	Παράμετροι Επεξεργασίας και Αποτελέσματα	100
3.1.6.3	Ατομικές και Απλές Επεξεργασίες	109
3.1.6.4	Σύνθετες Επεξεργασίες	112
3.1.6.5	Καθορίζοντας τη Ροή Δεδομένων και τις Συνδέσεις των Παραμέτρων	121
3.1.7	Υποστηρίζοντας μια Υπηρεσία σε μια Σταθερή Υλοποίηση	131
3.1.7.1	Η Σχέση μεταξύ του OWL-S και της WSDL	132
3.1.7.2	Υποστήριξη στις Υπηρεσίες του OWL-S με το WSDL και το SOAP	135
3.1.7.2.1	Η Προδιαγραφή της WSDL	135
3.1.7.2.2	Η Τάξη Υποστήριξης του OWL-S	136
3.1.8	Περίληψη και Τρέχουσα Κατάσταση	138
3.1.9	Παραρτήματα	139
3.2	Περιγραφή των υπηρεσιών δικτύου χρησιμοποιώντας το OWL-S και την WSDL 144	
3.2.1	Σύνοψη	144
3.2.2	Εισαγωγή	145
3.2.3	Παράδειγμα για το OWL-S/WSDL	148
3.2.4	Υποστήριξη Υπηρεσιών με την WSDL	153

3.2.4.1	Η Δομή των WSDL Εγγράφων και τα Στοιχεία Επεκτασιμότητας για το OWL-S.....	153
3.2.4.2	Τύποι.....	157
3.2.4.3	Μηνύματα.....	158
3.2.4.4	Τύποι θύρας.....	159
3.2.4.5	Από τις διαδικασίες του OWL-S στις λειτουργίες της WSDL.....	161
3.2.4.6	Σύνδεση.....	161
3.3	Παραδείγματα Οντολογιών.....	162
3.3.1	Η Ιεραρχία του Προφίλ στο Owl.....	162
3.3.2	Προφίλ Βασισμένο στις Ιεραρχίες Τάξης.....	164
3.3.2.1	Επεξηγηματικές Επισημάνσεις για την Ιεραρχία του Προφίλ του OWL-S 1.1.....	164
3.4	Η Σχέση του OWL-S με Άλλες Τεχνολογίες.....	166
3.4.1	Εισαγωγή.....	166
3.4.2	WSDL.....	167
3.4.3	SOAP (Simple Object Access Protocol).....	168
3.4.4	UDDI (Universal Description, Discovery and Integration).....	169
3.4.5	BPEL4WS.....	170
3.4.6	ebXML.....	171
3.4.7	CDL (Choreography Description Language).....	172
3.4.8	OWL και SWRL.....	173
3.4.9	Grid Services.....	174
3.4.10	WSMO.....	175
4	ΚΕΦΑΛΑΙΟ: ΕΡΕΥΝΑ ΚΑΙ ΕΦΑΡΜΟΓΕΣ ΤΟΥ OWL-S.....	176
4.1	Ασφάλεια Και Ακεραιότητα με το Πρότυπο OWL-S.....	176
4.1.1	Εισαγωγή.....	176
4.1.2	Εισαγωγή της Πολιτικής Ασφαλείας στις Περιγραφές των Υπηρεσιών Δικτύου.....	178
4.1.3	Παράδειγμα.....	179
4.1.4	Ασφάλεια και Πιστοποίηση OWL-S.....	180
4.1.5	Αναπαράσταση και Ατιολόγηση των Πολιτικών.....	183
4.1.5.1	Πολιτικές Αναπαράστασης Χρησιμοποιώντας τη Rei.....	183
4.1.5.2	Επεκτείνοντας το OWL-S.....	187
4.1.5.3	Επιλογή Προμηθευτών Χρησιμοποιώντας τις Πολιτικές.....	188
4.1.5.4	Εξακριβώνοντας την Παρουσία των Πολιτικών κατά τη Διάρκεια της Αλληλεπίδρασης.....	189
4.1.5.5	Ενισχύοντας τη Πολιτική και τη Πιστοποίηση σε OWL-S VM.....	191
4.1.5.6	Σύνοψη.....	194
4.2	Η Χρήση των Σημασιολογικών Υπηρεσιών Δικτύου στο Περιβάλλον των Κινητών Εφαρμογών.....	194

4.2.1	Εισαγωγή	194
4.2.2	Σύνοψη για το «Πανεπιστήμιο μου» (My Campus).....	195
4.2.3	Χρησιμοποιώντας τις Υπηρεσίες Δικτύου στις Πληροφορίες Περιβάλλοντος.....	197
4.2.3.1	Κανόνες Επίκλησης Υπηρεσιών	198
4.2.3.2	Οι Σημασιολογικές Υπηρεσίες Χρησιμοποιώντας το OWL-S.....	198
4.2.4	Η Σύνθεση των Υπηρεσιών	200
4.2.4.1	Αυτόματοι Τελεστές Εξαγωγής Δεδομένων και Σχεδιασμού.....	201
4.2.4.2	Σενάριο	203
4.2.5	Συμπέρασμα	205
4.3	Το Παράδοξο του Μεσίτη για το OWLS.....	206
4.3.1	Σύνοψη.....	206
4.3.2	Εισαγωγή	207
4.3.3	OWL-S	208
4.3.4	Γενική Περιγραφή ενός Μεσίτη	211
4.3.5	Ο Μεσίτης Υποστηρίζεται από την OWL-S.....	214
4.3.6	Επεκτείνοντας το OWL-S.....	216
4.3.6.1	Η Επίσημη Σημασιολογία του exe.....	218
4.3.7	Η Εφαρμογή ενός Μεσίτη	218
4.3.7.1	Εύρεση Βασισμένη στο Μεσίτη	219
4.3.8	Μεσολάβηση Βασισμένη στο Μεσίτη	220
4.3.9	Αλληλεπίδραση Βασισμένη στο Μεσίτη	221
4.4	Visual Modeling of OWL-S Services.....	224
4.4.1	Σύνοψη.....	224
4.4.2	Αιτιολόγηση.....	224
4.4.3	Περίληψη.....	227
4.4.4	OWLSWIZ.....	228
4.4.5	Visual Composer.....	230
4.4.5.1	Εκφράζοντας Συνθήκες και Επιρροές	231
4.4.5.2	Εκφράζοντας τη Ροή Δεδομένων	233
4.4.6	Αποτίμηση.....	234
4.4.7	Σχετική Εργασία.....	235
4.4.8	Μελλοντική Εργασία.....	236
4.4.9	Επίλογος	237
5	Βιβλιογραφία	239

Πρόλογος

Ο Σημασιολογικός Ιστός είναι ένα όραμα και μια πρόταση για την μετεξέλιξη του Διαδικτύου και ειδικότερα του Παγκόσμιου Ιστού (World Wide Web). Ο στόχος του σημασιολογικού δικτύου είναι να εξελίξει το σημερινό διαδίκτυο έτσι ώστε οι πληροφορίες που υπάρχουν και διακινούνται σε αυτό να είναι επεξεργάσιμες από τους υπολογιστές.

Ο Tim Berners-Lee, που επινόησε τον Παγκόσμιο Ιστό το 1989, είχε το όραμα, που τώρα συμμερίζονται πολλοί - ενός ιστού δεδομένων που θα μπορούν να επεξεργαστούν οι ηλεκτρονικοί υπολογιστές. Ο Tim Berners-Lee ορίζει το σημασιολογικό δίκτυο ως *«ένα σύνολο από συνδεδεμένες εφαρμογές δεδομένων στο διαδίκτυο δομημένες με τέτοιο τρόπο ώστε να διαμορφώνεται ένα συνεπές λογικό δίκτυο δεδομένων»*. Ένας δεύτερος ορισμός του ίδιου είναι ο εξής *"Ο Σημασιολογικός Ιστός είναι μια επέκταση του σημερινού ιστού όπου η πληροφορία έχει καθορισμένη σημασία, καθιστώντας τη συνεργασία μεταξύ ανθρώπων και υπολογιστών πιο αποτελεσματική"*, Tim Berners-Lee, James Hendler, Ora Lassila, The Semantic Web, Scientific American, Μάιος 2001.

Χαρακτηριστικό του σημασιολογικού δικτύου είναι ότι αποτελεί ένα διαδίκτυο στο οποίο η πληροφορία είναι καλύτερα ορισμένη επιτρέποντας με αυτόν τον τρόπο την αυτοματοποίηση και επαναχρησιμοποίηση των δεδομένων. Προβλέπεται ότι ο Ιστός του μέλλοντος θα αποτελεί μια παγκόσμια βάση δεδομένων και γνώσης, με πληροφορίες οι οποίες θα είναι "κατανοητές" από τους υπολογιστές.

Τα τελευταία χρόνια πολλές ερευνητικές ομάδες στο χώρο του διαδικτύου έχουν επικεντρώσει τις προσπάθειές τους στην ανάπτυξη υπηρεσιών, οι οποίες θα δώσουν επιπρόσθετες δυνατότητες στους χρήστες για την εξυπηρέτηση των ολοένα αυξανόμενων αναγκών τους. Οι υπηρεσίες αυτές έχουν υιοθετηθεί από πολλούς κατασκευαστές λογισμικού, σαν μέσο για τον εμπλουτισμό των ιστοσελίδων – και όχι μόνο, με δυναμικά στοιχεία, τα οποία βασίζονται σε συγκεκριμένες διεπαφές. Διάφοροι οργανισμοί εμπλέκονται στον καθορισμό προδιαγραφών για τις «Υπηρεσίες Δικτύου» με αποτέλεσμα αρκετοί ορισμοί να έχουν διατυπωθεί, με κυρίαρχο αυτόν της ερευνητικής ομάδας W3C Web Services Architecture Working Group «Μια υπηρεσία ιστού (web service), είναι μια εφαρμογή λογισμικού, η οποία προσδιορίζεται από ένα URI, της οποίας οι διεπαφές και οι συνδέσεις είναι δυνατόν να καθοριστούν, να περιγραφούν και να εντοπισθούν από τεχνουργήματα XML, και

υποστηρίζουν απευθείας αλληλεπίδραση με άλλες εφαρμογές λογισμικού, χρησιμοποιώντας μηνύματα βασισμένα σε XML, διαμέσου πρωτοκόλλων βασισμένων στο διαδίκτυο.» Σύμφωνα με τον παραπάνω ορισμό, η «Υπηρεσία Διαδικτύου» καθορίζεται σαν ένας πόρος, όχι μόνο μέσα στον παγκόσμιο ιστό, αλλά και σε επίπεδο ενός τοπικού εσωτερικού δικτύου (intranet), ο οποίος παρέχεται από κάποιον οργανισμό.

Ο σημασιολογικός ιστός έχει σαν στόχο να προσφέρει καλύτερη ποιότητα υπηρεσιών, ακριβέστερη αναζήτηση πάνω στον ήδη υπάρχον παγκόσμιο ιστό και καλύτερα αποτελέσματα. Η βάση για να επιτευχθούν όλα αυτά είναι η περιγραφή της ήδη υπάρχουσας πληροφορίας με επιπλέον πληροφορία κατανοητή τόσο από ανθρώπους, όσο και από υπολογιστικές μηχανές. Η περιγραφή αυτή μπορεί να γίνει με τη βοήθεια οντολογιών (ontologies), οι οποίες θα παρέχουν τις απαραίτητες πληροφορίες και τους συσχετισμούς μεταξύ των εννοιών και των αντικειμένων. Ας δώσουμε όμως έναν ορισμό για αυτές:

«Η τοποθέτηση και κατηγοριοποίηση των ειδών, πραγμάτων και εννοιών σε τύπους και κατηγορίες με ένα καλά ορισμένο τρόπο λέγεται οντολογία. Η οντολογία μπορεί να έχει όνομα ώστε να είναι κατανοητή από ανθρώπους, καθιστώντας έτσι εφικτή και την κατηγοριοποίηση συγκεκριμένων οντολογιών.»

Συνήθως, οι οντολογίες δεν περιέχουν μόνο πληροφορία για τα είδη ή τις έννοιες που περιγράφουν, αλλά παρέχουν επιπρόσθετες πληροφορίες για συσχετίσεις τους με άλλα είδη ή έννοιες. Σκοπός είναι να υπάρχει ένας καλά ορισμένος τρόπος για την περιγραφή των οντολογιών αυτών ή με άλλα λόγια, σκοπός είναι να υπάρχει μια καθιερωμένη γλώσσα για τον ορισμό και την περιγραφή των οντολογιών.

Η βάση τέθηκε από την RDF (Resource Description Framework) – μία γλώσσα για τον ορισμό των οντολογιών. Η γλώσσα αυτή σχεδιάστηκε για να κάνει δηλώσεις για συγκεκριμένους πόρους. Ένας πόρος μπορεί να αντιπροσωπεύει ένα σενάριο, μια

ιδέα, μια έννοια ή δεδομένα και ταυτοποιείται από ένα URI. Η RDF είναι η βάση για την OWL (Web Ontology Language), η οποία σχεδιάστηκε για να δώσει μια αποδοτικότερη αρχιτεκτονική πάνω στις οντολογίες. Η OWL προέκυψε από την DAML+OIL μια σχετική και αρκετά επιτυχημένη προσπάθεια της DARPA. Η γλώσσα αυτή χρησιμοποιεί κλάσεις και ιδιότητες της RDF, αλλά ταυτόχρονα τις επεκτείνει και ορίζει και νέες.

Η OWL χωρίζεται σε τρεις «υπογλώσσες» ή εκδόσεις: την OWL Lite, την OWL DL και την OWL Full. Η τελευταία είναι ολόκληρη η γλώσσα. Οι άλλες δύο είναι υποσύνολα ή περιορισμοί αυτής. Ο λόγος που υπάρχουν οι τρεις αυτές εκδόσεις είναι το γεγονός ότι τα δύο υποσύνολα της OWL Full παρέχουν λιγότερη δύναμη, αλλά περιορίζουν σημαντικά το υπολογιστικό κόστος. Η OWL DL υποστηρίζει μια μορφή αυτού που λέμε περιγραφή της λογικής (Description Logic). Η OWL DL προσδίδει προσεκτικά επιλεγμένους περιορισμούς πάνω στο είδος των αντικειμένων που περιγράφει με σκοπό τη μείωση των απαραίτητων υπολογιστικών πόρων. Με αυτόν

το τρόπο εγγυάται τον υπολογισμό των αποτελεσμάτων με χρήση των υπάρχοντων υπολογιστικών δυνατοτήτων. Η OWL Lite είναι ουσιαστικά η OWL DL, αλλά με περισσότερους περιορισμούς.

Όσον αφορά στις υπηρεσίες διαδικτύου, η OWL έρχεται να ενισχύσει τους προμηθευτές των υπηρεσιών με ένα σύνολο από δεδομένα και μετασχηματισμούς για την πληρέστερη περιγραφή των δυνατοτήτων των υπηρεσιών με ένα τρόπο άμεσα κατανοητό και μεταγλωττίσιμο από υπολογιστικές μηχανές. Παρέχει αυτόματη αναζήτηση των υπηρεσιών, ενισχύοντας έτσι το ρόλο της υπηρεσίας καταλόγου (Service Registry), καθώς και δυναμική αλληλεπίδραση μεταξύ των υπηρεσιών και σύνθεσή τους. Ουσιαστικά, η δύναμή της έγκειται στο σημασιολογικό ορισμό των εννοιών και των σεναρίων γύρω από την υπηρεσία διαδικτύου, κάτι που η WSDL αδυνατεί να υποστηρίξει και υστερεί στον τομέα αυτό. Με τον τρόπο αυτό, η OWL προσδίδοντας μεγαλύτερη δύναμη στην υπηρεσία καταλόγου, στους προμηθευτές και γενικά στους αντιπροσώπους των υπηρεσιών διαδικτύου στοχεύει στην αύξηση της ποιότητας της υπηρεσίας, της σταθερότητας και της εμπιστοσύνης γύρω από αυτή. Κλείνοντας, η OWL είναι μια γλώσσα για οντολογίες η οποία:

- Είναι ικανή να αναφέρεται σε σενάρια, ιδέες, έννοιες, αντικείμενα και δεδομένα που είναι ορισμένα στον παγκόσμιο ιστό.
- Μπορεί να διαμοιράζεται πάνω στον ιστό.
- Είναι ικανή να συνεργάζεται με άλλες παρόμοιες γλώσσες, όπως η RDF.
- Μπορεί να συγχωνεύει πολλές οντολογίες ταυτόχρονα.
- Είναι ευρέως αποδεκτή.
- Είναι αρκετά εκφραστική.

Όλα τα παραπάνω καθιστούν την OWL την κυρίαρχη γλώσσα πάνω στον παγκόσμιο ιστό για το σκοπό που περιγράφουμε και αναμφίβολα θα συνεχίσει να εξελίσσεται.

Στην σημερινή μορφή του Διαδικτύου, οι υπολογιστές χρησιμοποιούνται απλώς για την αποθήκευση, την μετάδοση και την εμφάνιση των πληροφοριών. Μοναδικός αποδέκτης των πληροφοριών αυτών είναι ο τελικός χρήστης ο οποίος βλέπει τις σελίδες του Διαδικτύου στην οθόνη του υπολογιστή του. Αν λάβουμε υπόψη ότι στο Διαδίκτυο είναι αποθηκευμένη και διακινείται μια γιγαντιαία ποσότητα πληροφοριών, η προοπτική της επεξεργασίας αυτών των πληροφοριών από υπολογιστές ανοίγει τον δρόμο για νέες επαναστατικές εφαρμογές.

1 ΚΕΦΑΛΑΙΟ: ΥΠΗΡΕΣΙΕΣ ΙΣΤΟΥ ΚΑΙ ΣΗΜΑΣΙΟΛΟΓΙΚΟΣ ΙΣΤΟΣ

1.1 Εισαγωγή Στις Υπηρεσίες Ιστού

1.1.1 Εισαγωγή

Κοιτώντας πίσω τα τελευταία 6 χρόνια, είναι πολύ δύσκολο να φανταστούμε πώς εργαζόμασταν στο δίκτυο χωρίς τον παγκόσμιο ιστό. Ο λόγος που η ιδέα του παγκόσμιου δικτύου πέτυχε, ενώ διάφορα σχέδια που είχαν γίνει παλαιότερα για τη δημιουργία υπερκειμένων απέτυχαν, μπορεί να εντοπιστεί σε ένα ζευγάρι βασικών παραγόντων: στην απλότητα και στην εύκολη πρόσβαση. Ας το δούμε από την πλευρά ενός προμηθευτή υπηρεσιών π.χ ενός e-shop, αν μπορούν να προβάλλουν μία ιστοσελίδα στο δίκτυο, τότε μπορούν να συμμετέχουν σε μια παγκόσμια κοινότητα. Από την πλευρά του πελάτη, αν μπορεί να πληκτρολογεί τότε μπορεί να έχει πρόσβαση και σε πληροφορίες. Από την πλευρά μιας υπηρεσίας API η πλειοψηφία των εργασιών στο δίκτυο γίνεται με τρεις μεθόδους- πάρε, ανακοίνωσε και βάλε, καθώς και με μία απλή γλώσσα σήμανσης. Η πορεία των υπηρεσιών δικτύου είναι η απόδειξη ότι ο παγκόσμιος ιστός σαν σχέδιο, δεν έχει εφαρμογή μόνο στις πληροφορίες αλλά και στις υπηρεσίες.

Με τον όρο υπηρεσίες δεν εννοούμε τις μονολιθικές ακατέργαστες πληροφορίες όπως το Amazon.com, αλλά τις σύνθετες υπηρεσίες που μπορούν να χρησιμοποιηθούν σαν βάση για μεγαλύτερες υπηρεσίες. Για παράδειγμα το Passport της Microsoft προσφέρει μια λειτουργία πιστοποίησης την οποία εξάγει στο δίκτυο. Επομένως μια ηλεκτρονική εφημερίδα όπως η Washington Post μπορεί να αποφύγει τη δημιουργία μιας δικής της υπηρεσίας πιστοποίησης, αναθέτοντας τη στο Passport.

Το Dynamic Services Whitepaper της Oracle προσφέρει και άλλα παραδείγματα σύνθετων υπηρεσιών που είναι στην ουσία επαναχρησιμοποιήσιμες υπηρεσίες όπως η μετατροπή συναλλάγματος, η μετάφραση, η ναυτιλία και η επεξεργασία απαιτήσεων.

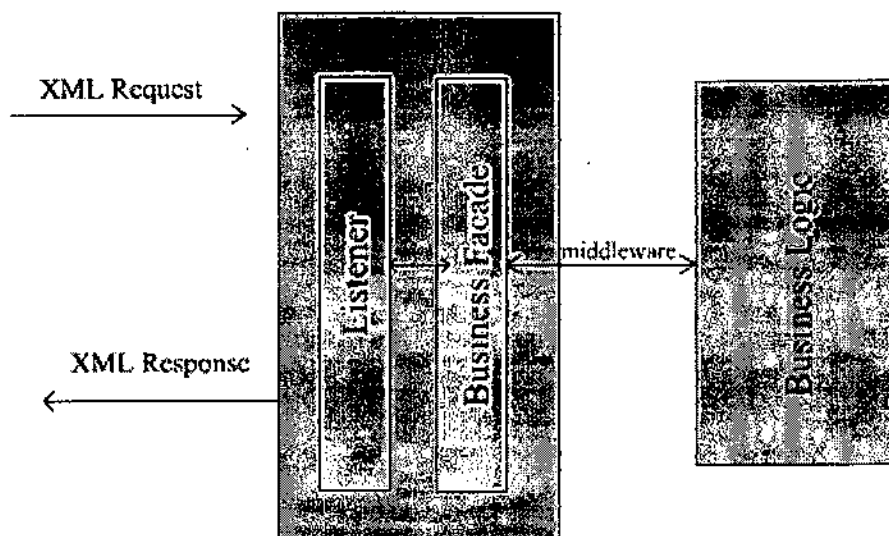
Ένα πιο επίσημο ορισμό των υπηρεσιών δικτύου μπορούμε να δανειστούμε από το διδακτικό έγγραφο της IBM πάνω στο θέμα:

Οι υπηρεσίες δικτύου είναι ένα νέο είδος των δικτυακών εφαρμογών. Είναι ανεξάρτητες, αυτοπεριγραφικές, αρθρωτές υπηρεσίες που μπορούν να ανακοινωθούν, να εντοπιστούν και να επικαλεστούν σε όλο το δίκτυο. Οι υπηρεσίες δικτύου εκτελούν λειτουργίες που μπορούν να αφορούν οτιδήποτε, από ένα απλό αίτημα μέχρι πολύπλοκες επιχειρηματικές επεξεργασίες.. Μόλις αναπτυχθεί μια υπηρεσία δικτύου, οι υπόλοιπες εφαρμογές και υπηρεσίες δικτύου μπορούν να βρουν και να καλέσουν την υπηρεσία που αναπτύχθηκε.

Το διδακτικό έγγραφο της IBM για τις υπηρεσίες δικτύου συνεχίζει λέγοντας ότι πριν λίγα χρόνια η αντίληψη που είχαμε για τις υπηρεσίες δικτύου ήταν αρκετά ανεπαρκής για να είναι ενδιαφέρουσα. Οι τάσεις που επικράτησαν όμως όπως το οικονομικότερο εύρος μετάδοσης και αποθήκευσης, το δυναμικότερο περιεχόμενο, η διεισδυτικότητα και η ποικιλία των υπολογιστικών συσκευών με διαφορετικά συστήματα πρόσβασης κάνουν την ανάγκη για ένωση πιο σημαντική, ενώ ταυτόχρονα κάνουν τα κόστη λιγότερο δυσάρεστα.

Θα μπορούσε να τεθεί το ερώτημα γιατί να ασχολούμαστε με τον παγκόσμιο ιστό ενώ υπάρχουν ενδιάμεσα προγράμματα τα οποία μας εξυπηρετούν όπως τα RMI, Jini, CORBA, DCOM, κ.α. Ωστόσο ενώ τα ενδιάμεσα προγράμματα παρέχουν υψηλά μέσα υλοποίησης, κανένα από αυτά δεν υπερέχει. Τα πλεονεκτήματα του παγκόσμιου ιστού σαν διανομέα πληροφοριών, η ευκολία στην πρόσβαση και το γεγονός ότι βρίσκεται παντού, σε σχέση με τον τεμαχισμένο κόσμο των ενδιάμεσων προγραμμάτων, όπου η διαλειτουργικότητα είναι δύσκολο να επιτευχθεί. Ο παγκόσμιος ιστός συμπληρώνει αυτά τα προγράμματα, προσφέροντας μια ενιαία και ευρέως προσβάσιμη κοινή επιφάνεια καθώς και τη δυνατότητα ένωσης με υπηρεσίες όπου η πρόσβαση υλοποιείται αποτελεσματικότερα μέσω των κλασσικών ενδιάμεσων προγραμμάτων.

Αν το δούμε από αρχιτεκτονική πλευρά, η υπηρεσία δικτύου είναι το πρώτο επίπεδο στην προγραμματισμένη πρόσβαση σε μια υπηρεσία, που γίνεται μετά υλοποιήσιμη από άλλα είδη ενδιάμεσων προγραμμάτων. Η πρόσβαση περιλαμβάνει τον χειρισμό απαιτήσεων ενός χρήστη που αγνοεί την υπηρεσία και την έκθεση των λειτουργιών που υποστηρίζονται από την επιχειρησιακή λογική. Η ίδια η λογική είναι υλοποιήσιμη από μία παραδοσιακή ενδιάμεση πλατφόρμα.



Σχήμα 1- Η Γενική Αρχιτεκτονική των Υπηρεσιών Δικτύου

1.1.2 Η Πλατφόρμα των Υπηρεσιών Δικτύου

Η βασική πλατφόρμα περιλαμβάνει την XML και το HTTP. Το HTTP είναι το πρωτόκολλο το οποίο διατρέχει πρακτικά όλο το διαδίκτυο. Η XML παρέχει μια γλώσσα μεταδεδομένων, με βάση την οποία μπορούμε να δημιουργήσουμε πιο ειδικευμένες γλώσσες για να περιγράψουμε τις περίπλοκες αλληλεπιδράσεις ανάμεσα σε πελάτες και υπηρεσίες ή ανάμεσα στα στοιχεία μιας σύνθετης υπηρεσίας. Πίσω από την εξωτερική όψη ενός εξυπηρετητή δικτύου, το μήνυμα της XML μετατρέπεται σε ενδιάμεση αίτηση και ύστερα τα αποτελέσματα μετατρέπονται πάλι σε XML.

Η πρόσβαση και η επίκληση δεν είναι τα μόνα κύρια χαρακτηριστικά, υπάρχει και η πλατφόρμα των υπηρεσιών υποστήριξης - η εύρεση, οι συναλλαγές, η ασφάλεια, η πιστοποίηση κ.α.

Ο παγκόσμιος ιστός χρειάζεται να εμπλουτιστεί και με κάποια άλλα σχέδια υπηρεσιών, που διατηρούν τη απανταχού παρουσία και την απλότητα του παγκόσμιου ιστού, για να σχηματίσουν μια πιο λειτουργική πλατφόρμα. Μια πλήρως λειτουργική πλατφόρμα μπορεί να θεωρηθεί η σύνθεση της XML μαζί με το HTTP, το SOAP, την WSDL και την UDDI. Σε υψηλότερα επίπεδα θα μπορούσαμε να προσθέσουμε τεχνολογίες όπως η XAML, η XLANG, η XKMS, και η XFS, υπηρεσίες που δεν είναι παγκοσμίως αποδεκτές ως επιβεβλημένες.

Παρακάτω κάνουμε μια σύντομη περιγραφή αυτών των στοιχείων. Θα πρέπει να σημειώσουμε, ότι ενώ οι πωλητές προσπαθούν να παρουσιάσουν τις ανερχόμενες δικτυακές υπηρεσίες σαν κάτι εύκολα κατανοητό, στην ουσία είναι μια σειρά από αναπτυγμένες τεχνολογίες. Σε υψηλότερα επίπεδα ανάλυσης υπάρχουν πολλαπλές προσεγγίσεις στο ίδιο πρόβλημα.

- SOAP (απομακρυσμένη επίκληση)
- UDDI (ανταλλαγές, υπηρεσίες διευθύνσεων)
- WSDL (εκφράζει τα χαρακτηριστικά των υπηρεσιών)
- XLANG/XAML (υποστήριξη συναλλαγών για περίπλοκες δικτυακές συναλλαγές που αφορούν πολλαπλές υπηρεσίες δικτύου)
- XKMS (XML Key Management Specification) – αυξανόμενη εργασία από τη Microsoft και τη Verisign για πιστοποίηση και εγγραφή.

1.1.3 SOAP

Το πρωτόκολλο SOAP ορίζει έναν ενιαίο τρόπο μετάδοσης των κωδικοποιημένων XML δεδομένων. Ορίζει επίσης τον τρόπο εκτέλεσης απομακρυσμένων διαδικασιών κλήσης χρησιμοποιώντας το HTTP σαν βασικό πρωτόκολλο επικοινωνίας.

Το SOAP αναδείχτηκε ύστερα από τη διαπίστωση ότι όσα και αν μπορούν να προσφέρουν τα σημερινά ενδιάμεσα προγράμματα, χρειάζονται ένα WAN περιτύλιγμα. Από αρχιτεκτονική άποψη, το να στέλνουμε μηνύματα καθαρά σε XML, είναι σημαντικό για τη διασφάλιση της διαλειτουργικότητας.

Η περαιτέρω ανάπτυξη του SOAP έχει αφεθεί πλέον στα χέρια της ομάδας εργασίας XML πρωτοκόλλων του W3C, επομένως η κατάσταση στην οποία βρίσκεται αυτή τη στιγμή το είναι σταθερή μέχρι τη στιγμή που το W3C θα παραδώσει τη προδιαγραφή του.

1.1.4 UDDI (Universal Description, Discovery and Integration Service)

Η UDDI (Universal Description, Discovery and Integration Service) παρέχει ένα μηχανισμό που βοηθά στην δυναμική εύρεση άλλων υπηρεσιών δικτύου. Χρησιμοποιώντας μια κοινή επιφάνεια UDDI, οι επιχειρήσεις μπορούν δυναμικά να συνδεθούν με υπηρεσίες που παρέχονται από εξωτερικούς επιχειρησιακούς συνεργάτες. Το UDDI είναι παρόμοιο με το CORBA ή αλλιώς μπορεί να θεωρηθεί σαν μια υπηρεσία DNS για επιχειρησιακές εφαρμογές. Χρησιμοποιείται από δύο είδη πελατών: τις επιχειρήσεις που θέλουν να ανακοινώσουν μια υπηρεσία και τους πελάτες που θέλουν να αποκτήσουν υπηρεσίες συγκεκριμένου είδους και να δεσμεύονται προγραμματικά με αυτές.

Ο πίνακας παρακάτω περιγράφει συνοπτικά τι ακριβώς προσφέρει η UDDI. Η UDDI είναι ένα επίπεδο πάνω από το SOAP και θεωρεί δεδομένο ότι οι αιτήσεις και οι απαντήσεις είναι αντικείμενα της UDDI τα οποία στέλνονται σαν SOAP μηνύματα. Ένα δείγμα ακολουθίας περιλαμβάνεται παρακάτω.

ΠΛΗΡΟΦΟΡΙΕΣ	ΛΕΙΤΟΥΡΓΙΕΣ	Αναλυτική Πληροφόρηση (υποστηρίζεται από ένα χαμηλότερου επιπέδου API)
Λευκές Σελίδες: Πληροφορίες όπως το όνομα, η διεύθυνση, ο αριθμός τηλεφώνου καθώς και άλλες πληροφορίες επαφών μιας δεδομένης επιχείρησης	Έκδοση: Πως αυτός που παρέχει μιας υπηρεσία δικτύου εγγράφει τον ίδιο	Επιχειρησιακές Πληροφορίες: Περιλαμβάνονται σε ένα αντικείμενο επιχειρησιακής οντότητας (<i>BusinessEntity</i>) που με τη σειρά του περιέχει πληροφορίες για υπηρεσίες, κατηγορίες, επαφές, URLs, και ότι άλλο είναι απαραίτητο προκειμένου να αλληλεπιδράσει με μια δεδομένη επιχείρηση
Κίτρινες Σελίδες: Πληροφορίες που κατηγοριοποιούν τις επιχειρήσεις. Οι	Εύρεση: Πως μια εφαρμογή βρίσκει μια συγκεκριμένη	Πληροφορίες Υπηρεσιών: Περιγράφει μια ομάδα υπηρεσιών δικτύου. Αυτά περιλαμβάνονται και σε ένα αντικείμενο επιχειρησιακών

πληροφορίες αυτές βασίζονται σε υπάρχοντα μη ηλεκτρονικά πρότυπα	υπηρεσία δικτύου	υπηρεσιών (<i>BusinessService</i>)
Πράσινες Σελίδες: Τεχνικές πληροφορίες σχετικά με τις υπηρεσίες δικτύου, οι οποίες δίνονται από μια δεδομένη επιχείρηση.	Σύνδεση: Πως μια εφαρμογή συνδέεται και αλληλεπιδρά με μια υπηρεσία δικτύου ύστερα από την εύρεση της.	Δεσμευμένες πληροφορίες: Οι τεχνικές πληροφορίες που είναι απαραίτητες για να τεθεί σε λειτουργία μια υπηρεσία δικτύου. Περιλαμβάνει URLs, πληροφορίες σχετικά με τις ονομασίες μεθόδων, τύπους επιχειρημάτων, κ.α. Το αντικείμενο της δεσμευμένης φόρμας (<i>BindingTemplate</i>) αντιπροσωπεύει αυτά τα δεδομένα. Λεπτομερής Προδιαγραφές Υπηρεσιών: Περιλαμβάνει μεταδεδομένα που αφορούν διάφορες προδιαγραφές, οι οποίες υλοποιούνται από μια δεδομένη υπηρεσία δικτύου. Αυτά καλούνται <i>tModels</i> στην προδιαγραφή του UDDI.

Δεν υπάρχει κανένα βραχυπρόθεσμο σχέδιο που θα καθιστά τη UDDI ικανή να υποστηρίξει μια πλήρη εύρεση, π.χ. γεωγραφικά περιορισμένες έρευνες ή οδηγίες και συμβόλαια διαπραγματεύσεων που υποστηρίζονται από μεσάζοντες όπως το eLance. Η UDDI αναμένεται να είναι η βάση για υψηλότερου επιπέδου υπηρεσίες που θα υποστηρίζονται από άλλα πρότυπα. Η UDDI θα μπορεί αργότερα να υποστηρίξει πιο περίπλοκες επιχειρηματικές λογικές, συμπεριλαμβάνοντας την υποστήριξη επιχειρησιακών οργανισμών. Η UDDI υποστηρίζεται ευρέως και είναι κάτω από τη καθοδήγηση της IBM, της Microsoft και της Arriba, δεν θεωρείται όμως ακόμα πρότυπο.

1.1.5 Παράδειγμα στο UDDI

Το παρακάτω ερώτημα όταν ενσωματωθεί σε SOAP επιστρέφει πληροφορίες για τη Microsoft.

```
<find_business generic="1.0" xmlns="urn:uddi-org:api">
<name>Microsoft</name>
</find_business>
```

Το αποτέλεσμα είναι μια λεπτομερής λίστα <businessinfo> στοιχείων που έχουν πρόσφατα εγγραφεί στη Microsoft και περιλαμβάνουν πληροφορίες για την ίδια την υπηρεσία UDDI.

```
<businessList generic="1.0"
operator="Microsoft Corporation"
truncated="false"
xmlns="urn:uddi-org:api">
<businessInfos>
<businessInfo
businessKey="0076B468-EB27-42E5-AC09-9955CFF462A3">
<name>Microsoft Corporation</name>
<description xml:lang="en">
```

Empowering people through great software any time, any place and on any device is Microsoft's vision. As the worldwide leader in software for personal and business computing, we strive to produce innovative products and services that meet our customer's.

```
</description>
<serviceInfos>
<serviceInfo
businessKey="0076B468-EB27-42E5-AC09-9955CFF462A3"
serviceKey="1FFE1F71-2AF3-45FB-B788-09AF7FF151A4">
<name>Web services for smart searching</name>
</serviceInfo>
<serviceInfo
businessKey="0076B468-EB27-42E5-AC09-9955CFF462A3"
serviceKey="8BF2F51F-8ED4-43FE-B665-38D8205D1333">
```

```

    <name>Electronic Business Integration Services</name>
  </serviceInfo>
  <serviceInfo
    businessKey="0076B468-EB27-42E5-AC09-9955CFF462A3"
    serviceKey="611C5867-384E-4FFD-B49C-28F93A7B4F9B">
    <name>Volume Licensing Select Program</name>
  </serviceInfo>
  <serviceInfo
    businessKey="0076B468-EB27-42E5-AC09-9955CFF462A3"
    serviceKey="A8E4999A-21A3-47FA-802E-EE50A88B266F">
    <name>UDDI Web Sites</name>
  </serviceInfo>
</serviceInfos>
</businessInfo>
</businessInfos>
</businessList>

```

1.1.6 WSDL (Web Services Definition Language)

Η WSDL (Web Services Definition Language) παρέχει τον τρόπο στους προμηθευτές υπηρεσιών να περιγράφουν τη βασική δομή των απαιτήσεων από τις υπηρεσίες δικτύου πάνω σε διαφορετικά πρωτόκολλα ή κωδικοποιήσεις. Η WSDL χρησιμοποιείται για να περιγράψει τι μπορεί να κάνει μια υπηρεσία δικτύου, που ανήκει και πως υλοποιείται. Καθώς το αίτημα για ανεξαρτητοποίηση του SOAP/HTTP υπάρχει σε πολλές προδιαγραφές, η WSDL έχει νόημα μόνο όταν θεωρεί το συνδυασμό SOAP/HTTP/MIME σαν τον απομακρυσμένο μηχανισμό επίκλησης. Οι καταχωρήσεις της UDDI περιγράφουν διάφορες πλευρές των υπηρεσιών δικτύου, συμπεριλαμβάνοντας τις λεπτομέρειες σύνδεσης της υπηρεσίας. Η WSDL βρίσκεται στο υποσύνολο περιγραφής μιας UDDI υπηρεσίας.

Η WSDL ορίζει τις υπηρεσίες σαν συλλογή από δικτυακά τελικά σημεία ή πύλες. Στην WSDL οι αφηρημένοι ορισμοί των τελικών σημείων και των μηνυμάτων είναι διαχωρισμένοι από την πραγματική δικτυακή πλευρά ή από τις συνδέσεις στην μορφοποίηση δεδομένων. Αυτό επιτρέπει την επαναχρησιμοποίηση αφηρημένων

ορισμών των μηνυμάτων, τα οποία είναι αφηρημένες περιγραφές των δεδομένων που έχουν ανταλλαγή, καθώς και είδη πυλών, που είναι αφηρημένες περιγραφές των λειτουργιών. Το πραγματικό πρωτόκολλο και οι προδιαγραφές διαμόρφωσης δεδομένων για ένα συγκεκριμένο τύπο πύλης, συνιστούν μια επαναχρησιμοποιήσιμη σύνδεση. Μια πύλη δημιουργείται από τον συνδυασμό μιας διεύθυνσης δικτύου και μιας επαναχρησιμοποιούμενης σύνδεσης. Μια συλλογή από πύλες ορίζουν μια υπηρεσία. Συμπεραίνουμε επομένως ότι ένα έγγραφο WSDL χρησιμοποιεί τα παρακάτω στοιχεία για τον ορισμό μιας υπηρεσίας δικτύου:

- Τύποι, περιέχει ορισμούς για τους τύπους δεδομένων χρησιμοποιώντας ένα σύστημα τύπων όπως το XSD.
- Μήνυμα, ένας αφηρημένος ορισμός των δεδομένων που χρησιμοποιούνται για την επικοινωνία.
- Λειτουργία, μια αφηρημένη περιγραφή μιας ενέργειας που υποστηρίζεται από μια υπηρεσία.
- Κατηγορία θύρας, ένα αφηρημένο σύνολο λειτουργιών που υποστηρίζεται από ένα ή περισσότερα τελικά σημεία.
- Η σύνδεση, ένα σταθερό πρωτόκολλο και οι προδιαγραφές μορφοποίησης δεδομένων, για ένα συγκεκριμένο τύπο θύρας.
- Θύρα, ένα μεμονωμένο τελικό σημείο που ορίζεται σαν συνδυασμός σύνδεσης και διεύθυνσης δικτύου.
- Υπηρεσία, μια συλλογή τελικών σημείων που συνδέονται.

Γενικά η WSDL είναι μια φόρμα που αναφέρει το πώς οι υπηρεσίες πρέπει να περιγράφονται από τους πελάτες. Παρακάτω περιγράφουμε μια stock quote service advertisement και ένα αντιπροσωπευτικό δείγμα αίτησης/απάντησης για την υπηρεσία που ερευνά το συγκεκριμένο ερώτημα στη Motorola.

Service Advertisement

```
<?xml version="1.0"?>
<definitions name="StockQuote"
  targetNamespace="http://example.com/stockquote.wsdl"
  xmlns:tns="http://example.com/stockquote.wsdl"
```

```
xmlns:xsd1="http://example.com/stockquote.xsd"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns="http://schemas.xmlsoap.org/wsdl/">
```

```
<types>
```

```
<schema targetNamespace="http://example.com/stockquote.xsd"
```

```
  xmlns="http://www.w3.org/1999/XMLSchema">
```

```
<element name="TradePriceRequest">
```

```
<complexType>
```

```
<all>
```

```
<element name="tickerSymbol" type="string"/>
```

```
</all>
```

```
</complexType>
```

```
</element>
```

```
<element name="TradePrice">
```

```
<complexType>
```

```
<all>
```

```
<element name="price" type="float"/>
```

```
</all>
```

```
</complexType>
```

```
</element>
```

```
</schema>
```

```
</types>
```

```
<message name="GetLastTradePriceInput">
```

```
<part name="body" element="xsd1:TradePriceRequest"/>
```

```
</message>
```

```
<message name="GetLastTradePriceOutput">
```

```
<part name="body" element="xsd1:TradePrice"/>
```

```
</message>
```

```
<portType name="StockQuotePortType">
```

```
<operation name="GetLastTradePrice">
```

```
<input message="tns:GetLastTradePriceInput"/>
```

```
<output message="tns:GetLastTradePriceOutput"/>
```

```

</operation>
</portType>

<binding name="StockQuoteSoapBinding"
  type="tns:StockQuotePortType">
  <soap:binding style="document"
    transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="GetLastTradePrice">
    <soap:operation
      soapAction="http://example.com/GetLastTradePrice"/>
    <input>
      <soap:body use="literal"
        namespace="http://example.com/stockquote.xsd"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
    </input>
    <output>
      <soap:body use="literal"
        namespace="http://example.com/stockquote.xsd"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
    </output>
  </operation>
</binding>

<service name="StockQuoteService">
  <documentation>My first service</documentation>
  <port name="StockQuotePort" binding="tns:StockQuoteBinding">
    <soap:address location="http://example.com/stockquote"/>
  </port>
</service>

</definitions>

<binding name="StockQuoteServiceBinding"
  type="StockQuoteServiceType">

```

```

<soap:binding style="rpc"
transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="getQuote">
    <soap:operation
soapAction="http://www.getquote.com/GetQuote"/>
    <input>
      <soap:body type="InMessageRequest"
namespace="urn:live-stock-quotes"
encoding="http://schemas.xmlsoap.org/soap/encoding"/>
    </input>
    <output>
      <soap:body type="OutMessageResponse"
encoding="http://schemas.xmlsoap.org/soap/encoding"/>
    </output>
  </operation>
</binding>
<service name="StockQuoteService">
  <documentation>My first service
    </documentation>
  <port name="StockQuotePort"
binding="tns:StockQuoteBinding">
    <soap:address location="http://example.com/stockquote"/>
  </port>
</service>
</definitions>

```

Μια SOAP ενσωματωμένη επίκληση στην υπηρεσία StockQuote

```

POST /StockQuote HTTP/1.1
Host: www.stockquoteserver.com
Content-Type: text/xml;
charset="utf-8"
Content-Length: nnnn
SOAPAction: "Some-URI"

```

```
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <m:GetLastTradePrice
xmlns:m="Some-URI">
      <symbol>MOT</symbol>
    </m:GetLastTradePrice>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Μια SOAP ενσωματωμένη επίκληση στην υπηρεσία StockQuote

```
HTTP/1.1 200 OK Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
```

```
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <m:GetLastTradePriceResponse
xmlns:m="Some-URI">
      <Price>14.5</Price>
    </m:GetLastTradePriceResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

1.1.7 XLANG

Η παραδοσιακή άποψη που επικρατεί για τις συναλλαγές ανάμεσα στις βάσεις δεδομένων είναι ότι πρέπει να πραγματοποιηθεί ολόκληρη η συναλλαγή ή αλλιώς καθόλου. Παρέχοντας τέτοιου είδους εγγύηση σε μία κατανεμημένη υποδομή,

εμπλέκεται μια υψηλού κόστους διαδικασία που καλείται σύνδεση δύο φάσεων. Ένα διαφορετικό μοντέλο έχει προταθεί από τις έρευνες που έχουν πραγματοποιηθεί στο τομέα των βάσεων δεδομένων, όπου οι ενέργειες περιέχουν αντισταθμιστικούς παράγοντες που ακυρώνουν την επίδραση μιας ενέργειας. Στην πραγματικότητα η ύπαρξη αντισταθμιστικών ενεργειών είναι αρκετά κοινή. Η XLang είναι ένα σύστημα χαρακτήρων και συμβόλων που χρησιμοποιείται για την έκφραση αντισταθμιστικών ενεργειών σε κάθε αίτημα που τελικά πρέπει να μείνει ανεκτέλεστο. Η υποδομή των υπηρεσιών δικτύου μπορεί να ενισχύσει τις προδιαγραφές της XLang ώστε να μπορεί να εκτελεί πιο περίπλοκες λειτουργίες αντιστάθμισης.

1.1.8 XAML (Transaction Authority Markup Language)

Η XAML (Transaction Authority Markup Language) προσφέρει ένα πιο παραδοσιακό τρόπο σύνδεσης δύο φάσεων, χρησιμοποιώντας συναλλαγές με σημασιολογικά στοιχεία πάνω στις υπηρεσίες δικτύου. Μια δικτυακή συναλλαγή business to business για την αγορά βενζανίου ακολουθεί παρακάτω σαν παράδειγμα. Η XAML δεν περιορίζει ολοκληρωτικά την ίδια σε σύνδεση δύο φάσεων και αφήνει ανοιχτό το ενδεχόμενο ότι κάποιες ανεκτέλεστες ενέργειες θα είναι αντισταθμιστικές ενέργειες όπως στην XLang. Ενώ η σύνδεση δύο φάσεων είναι ξεκάθαρα χρήσιμη στην ενότητα της επιχείρησης, κάποιες δικτυακές συναλλαγές είναι καλά φυλαγμένες από ένα αντισταθμιστικό μοντέλο το οποίο είναι από υπολογιστική άποψη, χαμηλότερου κόστους. Μέχρι να θεωρηθούν σημαντικές οι αντισταθμιστικές ενέργειες στο μοντέλο της XAML, η XLang θα έχει επαρκείς λόγους ύπαρξης.

Σενάριο

Το παρακάτω σενάριο δείχνει μια συναλλαγή σε επιχειρησιακό επίπεδο, η οποία περιλαμβάνει ένα σύνολο από υπηρεσίες δικτύου που θα υλοποιούσε η XAML. Ας υποθέσουμε ότι έχουμε μια εταιρεία η οποία αγοράζει βενζάνιο από ένα χημικό κατασκευαστή στο δίκτυο. Προκειμένου ο αγοραστής να προχωρήσει στην αγορά το βενζανίου, απαιτεί επιπρόσθετες υπηρεσίες που θα παρέχονται από τρίτους, όπως τη ναυτιλιακή διεκπεραίωση με συγκεκριμένους όρους παράδοσης,

πιστοδοτική πληρωμή, ασφάλεια σε περίπτωση ατυχήματος, καθώς και κυβερνητική ελαστικότητα για ασφαλή μεταφορά. Ο αγοραστής δεν θα προχωρήσει στην αγορά του βενζανίου αν δεν είναι όλες αυτές οι υπηρεσίες διαθέσιμες και αν δεν ικανοποιούν τις απαιτήσεις του. Επομένως θα προβεί είτε στην αγορά όλων είτε καμιάς. Με άλλα λόγια αν δεν ικανοποιούνται οι σχετικές με την αγορά απαιτήσεις, η επιχειρησιακή συναλλαγή δεν θα ολοκληρωθεί.

Το λογισμικό που παρέχει το υψηλότερο επίπεδο της επιχειρησιακής συναλλαγής, πρέπει να συντονιστεί με τη κάθε μια από τις υπηρεσίες δικτύου που συμμετέχουν. Αυτό περιλαμβάνει α) το σύστημα απογραφής του προμηθευτή χημικών β) τη πολιτική ασφαλείας που εξασφαλίζει ότι το προϊόν έχει ναυπηγηθεί γ) μια πιστοδοτική υπηρεσία που εξασφαλίζει τη πληρωμή με βάση τους όρους του πωλητή δ) μια υπηρεσία μεταφοράς που θα εγγυάται την έγκαιρη ναυπήγηση και παράδοση του προϊόντος ε) μια ρυθμιστική υπηρεσία που θα εξασφαλίζει ελαστικότητα με τις κυβερνητικές απαιτήσεις για ασφάλεια.

1.1.9 XKMS (XML Key Management Specification)

Η XKMS (XML Key Management Specification) είναι το αποτέλεσμα της προσπάθειας που γίνεται από την Microsoft και τη Verisign για την ενοποίηση του PKI και των ψηφιακών πιστοποιήσεων (που χρησιμοποιούνται για την ασφάλεια των συναλλαγών στο διαδίκτυο), χρησιμοποιώντας εφαρμογές της XML. Η κεντρική ιδέα ήταν να ανατεθεί η διαδικασία υπογραφής σε έναν εξυπηρετητή (trust server). Η XKMS βασίζεται στην προδιαγραφή XML Signature η οποία αποτελεί αντικείμενο μελέτης του W3C.

Η XKMS αποτελείται από δύο μέρη: την προδιαγραφή XML Key Information Service (X-KISS) και από την προδιαγραφή XML Key Registration Service Specification (X-KRSS). Η προδιαγραφή X-KISS ορίζει ένα πρωτόκολλο για μια trust service που αναλύει την δημόσια πληροφορία που περιέχεται σε XML-SIG στοιχεία. Το πρωτόκολλο X-KISS επιτρέπει σε κάποιον πελάτη να αναθέσει μέρος ή ακόμα και ολόκληρο το κομμάτι των εργασιών που είναι απαραίτητο για την επεξεργασία <ds:KeyInfo> στοιχείων. Ο κύριος στόχος του πρωτοκόλλου είναι να ελαχιστοποιήσει την πολυπλοκότητα των εφαρμογών υλοποίησης, με το να τις βάζει στη θέση του πελάτη. Έτσι, θωρακισμένες από την πολυπλοκότητα και το συντακτικό του

θεμελιώδους PKI, χρησιμοποιούνται για να καθιερώσουν σχέσεις εμπιστοσύνης. Αυτό μπορεί να βασίζεται σε διαφορετικές προδιαγραφές όπως X.509/PKIX, SPKI, ή PGP X-KRSS που περιγράφουν πως καταχωρούνται οι δημόσιες πληροφορίες.

Ενώ δεν υπάρχουν απaráβατες δεσμεύσεις σε αυτές τις προτάσεις για πρωτόκολλα και μεταφορές, η υπάρχουσα προδιαγραφή XKMS βασίζεται στην XML, τη SOAP, την WSDL. Οι άλλες προτάσεις που έχουν πραγματοποιηθεί στο συγκεκριμένο τομέα περιλαμβάνουν την S2ML (Security Services Markup Language) και την AuthXML που είναι ενοποιημένες κάτω από την καλή κηδεμονία της OASIS's XML Security Services.

1.1.10 Άλλες Προτάσεις

➤ **ADS (Advertisement and Discovery of Services Protocol)**

Το ADS αποδέχεται την WSDL σαν μια XML μορφοποίηση για μια υπηρεσία, θέλει όμως να ασχοληθεί και με υπηρεσίες που δεν υποστηρίζονται από την XML, για να δημιουργήσει περιγραφές σε WSDL. Η XML τυποποιείται σε ένα φάκελο svcsadvnt.xml που βρίσκεται στη ρίζα ενός εξυπηρετητή δικτύου, ο οποίος συλλογικά διαφημίζει όλες τις υπηρεσίες που είναι διαθέσιμες στη συγκεκριμένη ιστοσελίδα. Αυτό αφαιρεί το βάρος της διαφήμισης από τη κάθε υπηρεσία και παρέχει στους μηχανισμούς εύρεσης υπηρεσιών ένα μοναδικό ιστότοπο όπου θα μπορούν να ψάχνουν για διαφημίσεις. Για τα ηλεκτρονικά καταστήματα που θέλουν να διαφημίσουν τις υπηρεσίες τους χωρίς την XML, η ADS προτείνει μια ετικέτα με όνομα <serviceDescriptionLocation> και περιεχόμενο <valid URL of document containing service advertisements>. Στην περίπτωση μηχανισμών εύρεσης υπηρεσιών που βασίζονται σε HTML, ο μηχανισμός εύρεσης προβαίνει σε συμπεράσματα σχετικά με τις ιδιότητες της υπηρεσίας.

➤ **XFS (XMethods File System)**

Η XFS επιτρέπει την ανακοίνωση και ανάγνωση φακέλων μέσω της SOAP επιφάνειας. Αυτός ο τύπος συστήματος φακέλων μπορεί να χρησιμοποιηθεί για τη συγκέντρωση του όγκου των πληροφοριών, που μπορεί να είναι προσπελάσιμες από

πολλαπλές πηγές. Η XFS παρέχει ένα εργαλείο στον πελάτη που ενσωματώνει την βασισμένη σε XFS υπηρεσία δικτύου, σε περίβλημα του Windows Explorer. Το Windows Explorer ενοποιείται τότε με το σύστημα φακέλων που βασίζεται σε XML-SOAP. Εν κατακλείδι μπορούμε να παρατηρήσουμε ότι η ιδέα του XFS είναι τεχνικά ελκυστική.[1]

1.2 Ο Σημασιολογικός Ιστός

1.2.1 Εισαγωγή – Παράδειγμα

Ένας τύπος περιεχομένων Ιστού που θα έχει μεγάλη σημασία στους Η/Υ θα δημιουργήσει μια επανάσταση νέων πιθανοτήτων.

Το σύστημα ψυχαγωγίας χτυπούσε με τον ήχο του τραγουδιού των Beattle's "We Can Work It Out" όταν καλούσε το τηλέφωνο. Όταν ο Pete απάντησε η μουσική αυτόματα χαμήλωσε, η αδερφή του η Lucy ήταν στην άλλη γραμμή από το γραφείο του γιατρού: «Η μαμά χρειάζεται να επισκεφθεί έναν ειδικό και μετά πρέπει να πραγματοποιήσει ένα αριθμό φυσιοθεραπειών, δύο φορές την εβδομάδα ή κάτι παραπλήσιο, θα βάλω τον ατζέντη μου να κανονίσει τα ραντεβού». Ο Pete άμεσα συμφώνησε να μοιραστούν την οδήγηση.

Στο γραφείο του γιατρού, η Lucy έδωσε τις κατάλληλες πληροφορίες στον ατζέντη του σημασιολογικού ιστού διαμέσου του φυλλομετρητή. Ο ατζέντης γρήγορα ανέκτησε τις πληροφορίες σχετικά με την περιγραφόμενη θεραπεία της μαμάς από τον μεσίτη του γιατρού, έλεγξε αρκετές λίστες προμηθευτών και έλεγξε ποιες υπηρεσίες, με κατάταξη αρκετά καλή ή εξαιρετική που βρίσκονται σε ακτίνα 20 χιλιομέτρων, ταιριάζουν με τον τρόπο θεραπείας της μαμάς. Μετά προσπάθησε να βρει μια ώρα που θα ταιριάζει με τις διαθέσιμες ώρες συνάντησης των προμηθευτών (που φαίνονται μέσω των προσωπικών τους ιστοσελίδων) αλλά και με το πολυάσχολο πρόγραμμα του Pete και της Lucy.

Σε λίγα λεπτά ο ατζέντης, τους παρουσίασε ένα σχέδιο. Του Pete δεν του άρεσε. Το Πανεπιστημιακό Νοσοκομείο ήταν μακριά από το σπίτι της μαμάς και θα οδηγούσε για πολλή ώρα. Έβαλε τον δικό του ατζέντη να ξανακάνει την έρευνα με αυστηρότερους περιορισμούς σχετικά με την τοποθεσία και το χρόνο. Ο ατζέντης της

Lucy έχοντας πλήρη εμπιστοσύνη στον ατζέντη του Pete και στο περιεχόμενο της παρούσας εργασίας, αυτόματα βοήθησε παρέχοντας πιστοποιητικά εισχώρησης και μεθόδους συντόμευσης στα δεδομένα που είχαν ήδη ελεγχθεί.

Σχεδόν άμεσα, το νέο σχέδιο παρουσιάστηκε: μια πιο κοντινή κλινική και σε πιο σύντομο χρονικό διάστημα, αλλά υπήρχαν δύο ακόμα προβλήματα. Πρώτα ο Pete θα έπρεπε να επαναπροσδιορίσει λίγα από τα όχι και τόσο σημαντικά ραντεβού του. Έλεγχε και διαπίστωσε ότι αυτό δεν ήταν πρόβλημα. Το άλλο ήταν σχετικά με τη λίστα της ασφαλιστικής εταιρείας η οποία δεν περιείχε τον συγκεκριμένο παροχέα στη λίστα των φυσιοθεραπευτών. «Ο τύπος της υπηρεσίας και το επίπεδο της ασφάλειας επιβεβαιώνονται με άλλα μέσα», τον καθησύχασε ο ατζέντης.

Η Lucy έδωσε τη συγκατάθεσή της ενώ την ίδια στιγμή ο Pete γκρίνιαζε λέγοντας: «Δεν με ενδιαφέρουν οι λεπτομέρειες» και έτσι όλα είχαν κανονιστεί.

1.2.2 Το Νόημα

Ο Pete και η Lucy θα μπορούσαν, με τη χρήση των μεσιτών, να πραγματοποιήσουν όλες τις διαδικασίες χάρη στο σημασιολογικό ιστό, ο οποίος θα αναπτυχθεί στο μέλλον και όχι με τη βοήθεια του παγκόσμιου ιστού. Το μεγαλύτερο κομμάτι του δικτυακού περιεχομένου σήμερα, έχει σχεδιαστεί για ανάγνωση από τους χρήστες και όχι για χρήση από υπολογιστικά προγράμματα. Οι Η/Υ μπορούν επιδέξια ν' αναλύσουν τις ιστοσελίδες σε ότι αφορά τους συνδέσμους και άλλες διεργασίες ρουτίνας, δεν έχουν όμως κανένα αξιόπιστο τρόπο επεξεργασίας της σημασιολογίας.

Ο Σημασιολογικός Ιστός θα εισάγει την έννοια της δομής στο γεμάτο νόημα περιεχόμενο των ιστοσελίδων, δημιουργώντας ένα περιβάλλον, όπου οι πράκτορες λογισμικού θα μπορούν να πηγαίνουν από τη μια σελίδα στην άλλη και θα μπορούν εύκολα να τελούν κάποιες ενέργειες για τους χρήστες. Ένας ατζέντης αυτού του είδους, θα γνωρίζει μπαίνοντας στην ιστοσελίδα της κλινικής, ότι η σελίδα περιέχει λέξεις κλειδιά όπως «θεραπεία, φάρμακο, φυσιολογία», ότι ο Dr.Hartmann θα εργάζεται στην κλινική Δευτέρα, Τετάρτη, Παρασκευή και ότι η μορφή YYYY-MM-DD παρουσιάζει τις ώρες των ραντεβού. Η σημασιολογία αυτή κωδικοποιείται στην ιστοσελίδα όταν ο διευθυντής της κλινικής δημιουργεί μια πορεία, χρησιμοποιώντας

ένα προσωπικό λογισμικό για να γράφει τις σημασιολογικές ιστοσελίδες μαζί με τις πηγές που παρουσιάζονται στην ιστοσελίδα της Ένωσης Φυσιοθεραπευτών.

Ο Σημασιολογικός Ιστός δεν είναι ένας ξεχωριστός ιστός αλλά μια προέκταση του τρέχοντος μέσα στον οποίο οι πληροφορίες δίνουν ένα καλά καθορισμένο νόημα, δίνοντας τη δυνατότητα σε Η/Υ και ανθρώπους να συνεργάζονται. Τα πρώτα βήματα σχεδίασης του Σημασιολογικού Ιστού χρησιμοποιώντας τη δομή του υπάρχοντος Παγκόσμιου Ιστού έχουν γίνει. Στο άμεσο μέλλον, αυτές οι αναβαθμίσεις θα εξελιχθούν σε μια αξιόλογη νέα λειτουργικότητα καθώς οι μηχανές γίνονται ολοένα ικανότερες να επεξεργάζονται και να κατανοούν τα δεδομένα.

Η βασική ιδιότητα του Παγκόσμιου Ιστού είναι η παγκοσμιοποίηση του. Η δυνατότητα μιας υπερσύνδεσης συνοψίζεται στο ότι «οτιδήποτε μπορεί να συνδεθεί με οτιδήποτε». Η τεχνολογία δικτύου δεν θα πρέπει να κάνει διακρίσεις ανάμεσα στην πρόχειρη απεικόνιση και στην ραφιναρισμένη απόδοση, ανάμεσα στις οικονομικές και στις ακαδημαϊκές πληροφορίες ή ανάμεσα στις παραδόσεις, γλώσσες, Μ.Μ.Ε κ.α. Οι πληροφορίες ποικίλλουν ανάμεσα σε πολλούς άξονες. Ένας απ' αυτούς είναι η διαφορά ανάμεσα στις πληροφορίες που παράγονται για ανθρώπινη κατανάλωση και άλλες που παράγονται κυρίως για μηχανές. Στο ένα μέρος της κλίμακας έχουμε τα πάντα από διαφημιστικά των 5 δευτερολέπτων έως ποίηση. Στο άλλο μέρος έχουμε βάσεις δεδομένων, προγράμματα και αποτελέσματα αισθητήρων. Ο Ιστός έχει εξελιχθεί άμεσα ως ένα μέσο αντικειμένων για ανθρώπους παρά για δεδομένα και πληροφορίες που μπορούν να επεξεργάζονται αυτόματα. Ο Σημασιολογικός Ιστός έχει σαν στόχο να δημιουργήσει αυτή την αυτόματη επεξεργασία.

Όπως και το διαδίκτυο, ο Σημασιολογικός Ιστός θ' αποκεντρωθεί όσο το δυνατόν περισσότερο. Τέτοια δικτυακά συστήματα δημιουργούν ενθουσιασμό σε κάθε επίπεδο, από μεγάλες εταιρείες μέχρι τους ατομικούς χρήστες, και παρέχουν κέρδη τα οποία είναι δύσκολο ή αδύνατο να προβλεφθούν. Η αποκέντρωση απαιτεί δεσμεύσεις: ο Ιστός θα πρέπει ν' απαγκιστρωθεί από την έννοια της συνοχής όλων των συνδέσεων, αναγγέλλοντας το μήνυμα "ERROR 404: NOT FOUND".

1.2.3 Αναπαράσταση της Γνώσης

Για να λειτουργήσει ο Σημασιολογικός Ιστός, οι Η/Υ θα πρέπει να έχουν πρόσβαση στις δομημένες συλλογές πληροφοριών και στα σύνολα των κανόνων

συμπερασμάτων, για να συνθέσουν έναν αυτόματο συλλογισμό. Οι ερευνητές της τεχνητής νοημοσύνης έχουν μελετήσει τέτοια συστήματα πολύ πριν αναπτυχθεί ο ιστός. Η αναπαράσταση της γνώσης όπως αποκαλείται αυτή η τεχνολογία, είναι σ' ένα στάδιο εφάμιλλο με αυτό που ήταν τα υπερκείμενα πριν από τη δημιουργία του Ιστού. Είναι μια ενδιαφέρουσα ιδέα και μπορούμε να δούμε αξιοσημείωτες υλοποιήσεις της. Αποτελεί την αρχή σημαντικών εφαρμογών, αλλά για να μπορέσουμε να συνειδητοποιήσουμε το πλήρες φάσμα των δυνατοτήτων της θα πρέπει να συνδεθεί μ' ένα μοναδικό παγκόσμιο σύστημα.

Τα παραδοσιακά συστήματα αναπαράστασης γνώσης, τυπικά έχουν συγκεντρωθεί απαιτώντας να μοιράζεται ο καθένας ακριβώς τον ίδιο ορισμό σε κοινές έννοιες όπως «γονιός» ή «όχημα». Ωστόσο η έννοια του κεντρικού ελέγχου γίνεται ασφυκτική, η αύξηση του μεγέθους και του πεδίου δράσης ενός τέτοιου συστήματος, καθιστά αδύνατη τη διαχείριση του.

Επιπλέον αυτά τα συστήματα συνήθως περιορίζουν προσεκτικά τις ερωτήσεις που μπορούν να ερωτηθούν, για να μπορούν να απαντηθούν αξιόπιστα ή για να μπορούν να απαντηθούν εξολοκλήρου. Το πρόβλημα αυτό θυμίζει το θεώρημα του Godel από τα μαθηματικά: κάθε σύστημα που είναι αρκετά περίπλοκο ώστε να είναι χρήσιμο, περιέχει αναπάντητες ερωτήσεις που μοιάζουν με εκδοχές του βασικού παραδόξου «η πρόταση είναι λανθασμένη». Για να αποφύγουμε τέτοια προβλήματα, τα συστήματα παραδοσιακής αναπαράστασης γνώσης είχαν το δικό τους στενό, από άποψη ιδιοσυγκρασίας, σύνολο κανόνων για να δημιουργούν συμπεράσματα για τα δεδομένα π.χ ένα γενεαλογικό σύστημα λειτουργώντας πάνω σε μια βάση δεδομένων του οικογενειακού δέντρου, μπορεί να περιέχει τον κανόνα « η γυναίκα του θείου μου είναι θεία μου». Ακόμα και αν θα μπορούσαν τα δεδομένα να μεταφερθούν από το ένα σύστημα στο άλλο, οι κανόνες υπήρχαν σε μια εντελώς διαφορετική μορφή, αν και συνήθως αυτό δεν ήταν δυνατό.

Αντιθέτως, οι ερευνητές του Σημασιολογικού Ιστού, δέχονται αυτά τα παράδοξα και τις αναπάντητες ερωτήσεις σαν μια τιμή. Δημιουργούμε τη γλώσσα για τους κανόνες όσο εκφραστική απαιτείται από τον Ιστό, για να πραγματοποιεί τη διαδικασία υποστήριξης σε ευρύ επίπεδο. Η φιλοσοφία αυτή είναι παρόμοια με αυτή του τυπικού ιστού. Αρχικά, κατά τη διάρκεια ανάπτυξης του ιστού, οι επικριτές αυτής της ενέργειας έδειξαν ότι δε θα μπορούσε ποτέ να είναι μια καλά οργανωμένη βιβλιοθήκη, χωρίς μια κεντρική βάση δεδομένων και δομή δέντρου, δε θα ήταν ποτέ σίγουρο ότι θα έβρισκε κανείς οτιδήποτε. Είχαν δίκιο, αλλά η εκφραστική δύναμη του

συστήματος έκανε μεγάλες ποσότητες πληροφοριών διαθέσιμες και οι μηχανές αναζήτησης παράγουν τώρα ολοκληρωμένους δείκτες του διαθέσιμου υλικού. Η πρόκληση του Σημασιολογικού Ιστού, είναι να παρέχει μια γλώσσα που να εκφράζει τα δεδομένα και τους κανόνες υποστήριξης των δεδομένων ταυτόχρονα και να επιτρέπει στους κανόνες από κάθε σύστημα αναπαράστασης γνώσης, να εκτίθονται στο δίκτυο.

Προσθέτοντας λογική στον ιστό, το ζήτημα των μέσων που χρησιμοποιούν τους κανόνες για να βγάζουν συμπεράσματα και για να απαντούν ερωτήσεις, προβληματίζει την κοινωνία του Σημασιολογικού Ιστού. Ένα σύνολο μαθηματικών και μηχανικών αποφάσεων δυσκολεύουν την ενέργεια αυτή. Η λογική θα πρέπει να είναι αρκετά ισχυρή ώστε να μπορεί να περιγράψει σύνθετες ιδιότητες αντικειμένων, αλλά όχι τόσο δυνατή ώστε οι μεσίτες να μπορούν να παραπλανηθούν όταν τους ζητηθεί να λάβουν υπόψη τους ένα παράδοξο. Ευτυχώς ένα μεγάλο μέρος της πληροφορίας που θέλουμε να εκφράσουμε βρίσκεται στις γραμμές «a hex-head bolt is a type of machine bolt», το οποίο έχει γραφτεί με τη δυνατότητα να αναγνωστεί στις υπάρχουσες γλώσσες με λίγο περισσότερο λεξιλόγιο.

Οι απαραίτητες τεχνολογίες για την ανάπτυξη του Σημασιολογικού Ιστού έχουν ήδη βρεθεί. Η πρώτη είναι η eXtensible Markup Language (XML) και η δεύτερη είναι η Resource Description Framework (RDF). Η XML επιτρέπει στον καθένα να δημιουργήσει τις δικές του σημάνσεις, κρυμμένες ετικέτες που είτε σχολιάζουν ιστοσελίδες είτε μέρη ενός κειμένου σε μια σελίδα. Υπάρχουν σενάρια ή προγράμματα που μπορούν να χρησιμοποιήσουν αυτές τις ετικέτες με έξυπνους τρόπους, αλλά ο σεναριογράφος θα πρέπει να ξέρει πως χρησιμοποιεί ο συγγραφέας την κάθε ετικέτα. Η XML επιτρέπει στους χρηστές να προσθέσουν αυθαίρετη δομή στα κείμενα τους, αλλά δε λένε τίποτα για το τι σημαίνουν οι δομές αυτές.

Το νόημα εκφράζεται μέσω της RDF, η οποία το κωδικοποιεί σε ένα σύνολο τριάδων, όπου κάθε τριάδα είναι το υποκείμενο, το ρήμα και το αντικείμενο μιας στοιχειώδους πρότασης. Αυτές οι τριάδες μπορούν να γραφτούν χρησιμοποιώντας τις ετικέτες της XML. Στην RDF, το έγγραφο επιβεβαιώνει ότι συγκεκριμένα αντικείμενα έχουν ιδιότητες με συγκεκριμένες τιμές. Αυτή η δομή αποδεικνύεται ένας φυσικός τρόπος για να περιγράψεις την πλειονότητα των δεδομένων που επεξεργάζονται από μηχανές. Το υποκείμενο και το αντικείμενο αναγνωρίζονται από ένα URI, όπως χρησιμοποιούνται σαν μια σύνδεση σε μια ιστοσελίδα. Τα ρήματα

αναγνωρίζονται από το URI, το οποίο δίνει τη δυνατότητα στον καθένα να καθορίζει μια καινούρια ιδέα, ένα νέο ρήμα, καθορίζοντας το URI του κάπου στον ιστό.

Οι ανθρώπινες γλώσσες αναπτύσσονται όταν χρησιμοποιούν τον ίδιο όρο για να εκφράσουν διαφορετικά πράγματα, αλλά ο αυτοματισμός δεν το κάνει. Φανταστείτε ότι θέλουμε να προσλάβουμε ένα κλόουν από ένα γραφείο για να μοιράζει μπαλόνια στους πελάτες για τα γενέθλια τους. Δυστυχώς η υπηρεσία αυτή μεταφέρει τις διευθύνσεις μου από τη δική μου βάση δεδομένων στη δική τους, χωρίς να γνωρίζει ότι ορισμένες αναφέρονται σε διευθύνσεις αποστολής λογαριασμών και ότι κάποιες από αυτές είναι ταχυδρομικά κουτιά. Οι κλόουν που προσεληφθήκαν κατέληξαν να διασκεδάζουν ταχυδρομικούς υπαλλήλους, όχι απαραίτητα ότι είναι κακό, αλλά δεν είναι και το αναμενόμενο αποτέλεσμα. Η χρήση ενός διαφορετικού URI για κάθε ιδέα, λύνει αυτό το πρόβλημα. Κάθε διεύθυνση που αναφέρεται σε ταχυδρομείο θα μπορεί να αναγνωριστεί από μια που είναι ταχυδρομική διεύθυνση.

Το σύνολο των τριάδων της RDF σχετίζονται με τις πληροφορίες ιστού και τα συνδεδεμένα αντικείμενα. Επειδή η RDF χρησιμοποιεί το URI για να κωδικοποιήσει τις πληροφορίες αυτές σε ένα έγγραφο, αλλά συνδεδεμένες με ένα μοναδικό καθορισμό όπου ο καθένας θα μπορεί να βρει στον Ιστό. Π.χ φανταστείτε ότι έχουμε είσοδο σε πολλές βάσεις δεδομένων με πληροφορίες για ανθρώπους που ζουν σε ένα συγκεκριμένο ταχυδρομικό κώδικα, θα πρέπει να γνωρίζουμε ποια πεδία σε κάθε βάση δεδομένων αναπαριστούν τα ονόματα και ποια τους ταχυδρομικούς κώδικες. Η RDF μπορεί να ορίσει ότι το πεδίο 1 στη βάση δεδομένων A αντιστοιχεί στον ταχυδρομικό κώδικα.

1.2.4 Οντολογίες

Υπάρχουν βέβαια και άλλα προβλήματα προς επίλυση, λόγω του ότι δύο βάσεις δεδομένων μπορεί να χρησιμοποιούν διαφορετικούς μηχανισμούς αναγνώρισης κάτι για το οποίο τελικά είναι το ίδιο, όπως είναι ο ταχυδρομικός κώδικας. Ένα πρόγραμμα που θέλει να συγκρίνει ή να ενώσει πληροφορίες ανάμεσα στις δύο βάσεις δεδομένων, θα πρέπει να γνωρίζει ότι αυτοί οι δύο όροι έχουν το ίδιο νόημα. Ιδανικά, το πρόγραμμα θα πρέπει να έχει ένα τρόπο να βρίσκει τις κοινές έννοιες σε οποιοδήποτε βάσεις δεδομένων συναντά.

Η λύση σε αυτό το πρόβλημα παρέχεται από τον τρίτο βασικό παράγοντα του Σημασιολογικού Ιστού, από τις συλλογές πληροφοριών που καλούνται οντολογίες. Στη φιλοσοφία, ο όρος οντολογία αναφέρεται στη θεωρία για τη φύση της ύπαρξης, τι τύποι πραγμάτων υπάρχουν. Η τεχνητή νοημοσύνη και οι ερευνητές του ιστού έχουν χρησιμοποιήσει τον όρο για τους σκοπούς της επαγγελματικής τους διαλέκτου, και για αυτούς μια οντολογία είναι ένα έγγραφο ή ένας φάκελος ο οποίος τυπικά καθορίζει τις σχέσεις ανάμεσα στους όρους. Το πιο τυπικό είδος οντολογίας για τον ιστό έχει μια ταξινόμηση και ένα σύνολο κανόνων συμπερασμάτων.

Η ταξινόμηση καθορίζει τις τάξεις των αντικειμένων και τις σχέσεις μεταξύ τους. Π.χ μια διεύθυνση μπορεί να καθορίζεται ως ένας τύπος τοποθεσίας και κωδικών πόλεων που μπορεί να καθορίζονται μόνο για να παρέχονται με την τοποθεσία. Οι κατηγορίες, οι υποκατηγορίες και οι σχέσεις μεταξύ των οντοτήτων είναι ένα πολύ χρήσιμο εργαλείο για τους χρήστες του ιστού. Μπορούμε να εκφράσουμε ένα μεγάλο αριθμό σχέσεων μεταξύ των οντοτήτων εκχωρώντας τις ιδιότητες στις τάξεις και επιτρέποντας στις υποκατηγορίες να υιοθετούν τέτοιες ιδιότητες. Αν οι κωδικοί πόλεων πρέπει να είναι του τύπου πόλεις και οι πόλεις γενικά έχουν ιστοσελίδες, μπορούμε να πούμε ότι η ιστοσελίδα συνδέεται με ένα ταχυδρομικό κωδικό ακόμα και όταν καμία βάση δεδομένων δεν συνδέει το ταχυδρομικό κωδικό άμεσα με μια ιστοσελίδα.

Οι κανόνες συμπερασμάτων στις οντολογίες δίνουν περισσότερες δυνατότητες. Μια οντολογία μπορεί να εκφράζει τον κανόνα «Αν ένας ταχυδρομικός κώδικας σχετίζεται με ένα κωδικό πολιτείας και μια διεύθυνση χρησιμοποιεί τον ταχυδρομικό κώδικα, τότε η διεύθυνση έχει και το σχετικό κώδικα πολιτείας». Ένα πρόγραμμα τότε θα συμπεράνει ότι η διεύθυνση του Πανεπιστημίου Cornell, που βρίσκεται στο Ithaca, πρέπει να βρίσκεται στην πολιτεία της Νέας Υόρκης, που βρίσκεται στις Η.Π.Α. Ο Η/Υ δεν μπορεί να καταλάβει όλες τις πληροφορίες, αλλά τώρα μπορεί να χειρίζεται τους όρους περισσότερο αποτελεσματικά, με τρόπους που είναι χρήσιμοι και ευνόητοι στον ανθρώπινο χρήστη.

Με τις οντολογίες στον ιστό, λύνονται αρκετά προβλήματα σε θέματα οντολογίας. Το νόημα των όρων ή των κωδικών XML που χρησιμοποιούνται σε μια ιστοσελίδα μπορούν να καθοριστούν από δείκτες μιας σελίδας σε μια οντολογία. Φυσικά το ίδιο πρόβλημα, όπως πριν, τώρα εμφανίζεται αν κάποιος στοχεύει σε μια οντολογία η οποία καθορίζει διευθύνσεις που περιέχουν κωδικούς περιοχής και κάποιος άλλος στοχεύει σε μια που χρησιμοποιεί ταχυδρομικούς κώδικες. Αυτού του

είδους οι συγχύσεις μπορούν να λυθούν αν οι οντολογίες παρέχουν σχέσεις ισότητας: μιας ή περισσότερων οντολογιών που μπορεί να περιέχουν πληροφορίες ότι ο δικός μου ταχυδρομικός κώδικας είναι ίδιος με τον δικό σου ταχυδρομικό κωδικό.

Το σχέδιο να στείλουμε κλόουν να διασκεδάσει τους πελάτες μας έχει μερικώς επιλυθεί, όταν οι δύο βάσεις δεδομένων στοχεύουν σε διαφορετικούς προσδιορισμούς διευθύνσεων. Το πρόγραμμα χρησιμοποιώντας ευδιάκριτα URI για διαφορετικές έννοιες των διευθύνσεων, δεν θα τις μπερδέψει και θα ανακαλύψει ότι οι έννοιες δεν σχετίζονται. Το πρόγραμμα θα μπορούσε να χρησιμοποιήσει μια υπηρεσία που παίρνει μια λίστα ταχυδρομικών διευθύνσεων και να την μετασχηματίσει σε μια λίστα φυσικών διευθύνσεων αναγνωρίζοντας και μετακινώντας παλαιότερους ταχυδρομικούς κώδικες και άλλες αταίριαστες διευθύνσεις. Η δομή και η σημασιολογία που παρέχονται από τις οντολογίες, κάνουν πιο εύκολο για τον επιχειρηματία να παρέχει μια τέτοια υπηρεσία και να κάνει τη χρήση του ξεκάθαρη.

Οι οντολογίες μπορούν να εμπλουτίσουν τη λειτουργία του Ιστού με πολλούς τρόπους. Μπορούν να χρησιμοποιηθούν με ένα απλό τρόπο για να βελτιώσουν την ευστοχία των ερευνών του Ιστού. Περισσότερο αναπτυγμένες εφαρμογές, χρησιμοποιούν οντολογίες για να συσχετίσουν την πληροφορία σε μια σελίδα με τις αντίστοιχες δομές γνώσης και τους κανόνες συμπερασμάτων. Μια σελίδα για τέτοια χρήση είναι η <http://www.cs.umd.edu/~hendler>. Αν πάει ο φυλλομετρητής δικτύου σε αυτήν την σελίδα, θα δούμε την κανονική ιστοσελίδα που ονομάζεται «Dr. James A. Hendler». Ως χρήστης, μπορεί κάποιος άμεσα να βρει την σύνδεση με ένα βιογραφικό σημείωμα και να διαβάσει ότι ο Hendler πήρε το μεταπτυχιακό του στο Πανεπιστήμιο του Brown. Ένα πρόγραμμα Η/Υ που ψάχνει να βρει τέτοιες πληροφορίες, θα πρέπει να είναι αρκετά περίπλοκο για να υποθέσει ότι αυτή η πληροφορία βρίσκεται στο βιογραφικό και για να κατανοήσει την Αγγλική γλώσσα που χρησιμοποιείται.

Για τους Η/Υ, η σελίδα συνδέεται με μια σελίδα οντολογίας που καθορίζει πληροφορίες σχετικά με το τμήμα επιστήμης Η/Υ. Π.χ. η δουλειά των διδασκόντων στα πανεπιστήμια και γενικά αυτών που έχουν διδακτορικό. Η περεταίρω σήμανση στη συγκεκριμένη σελίδα χρησιμοποιεί την έννοια της οντολογίας για να περιγράψει ότι ο Hendler πήρε το μεταπτυχιακό του απ' την οντότητα που περιγράφεται από το URI <http://www.brown.edu>, την ιστοσελίδα του Brown. Οι Η/Υ μπορούν επίσης να

βρουν ότι ο Hendler είναι μέλος μιας συγκεκριμένης έρευνας, έχει ένα συγκεκριμένο e-mail κ.α. Όλες αυτές οι πληροφορίες επεξεργάζονται από τον Η/Υ και μπορούν να χρησιμοποιηθούν, για να απαντήσει σε ερωτήματα. Για να απαντήσει ο χρήστης στα ερωτήματα αυτά θα πρέπει να εξετάσει όλο το περιεχόμενο σελίδων που παρουσιάζονται από μια μηχανή αναζήτησης.

Επίσης, αυτή η βελτίωση κάνει ευκολότερο να αναπτυχθούν προγράμματα που θα μπορούν να μαρκάρουν σύνθετες ερωτήσεις των οποίων οι απαντήσεις δεν ανήκουν σε μια μόνο ιστοσελίδα. Υποθέστε ότι επιθυμείτε να βρείτε την Κα Cook την οποία γνωρίσατε σε ένα επαγγελματικό συνέδριο πέρυσι. Δεν θυμάστε το πρώτο της όνομα, αλλά θυμάστε ότι εργαζόταν σε ένα από τους πελάτες σας και ότι ο γιος ήταν μαθητής στο σχολείο που σπουδάσατε. Ένα έξυπνο πρόγραμμα έρευνας μπορεί να εξάγει όλες τις σελίδες των ανθρώπων που ονομάζονται «Cook», να βρει αυτούς που έχουν αναφέρει ότι εργάζονται σε μια εταιρεία που βρίσκεται στην λίστα πελατών σου και να ελέγξεις αν τα παιδιά τους βρίσκονται στο κατάλληλο σχολείο.

1.2.5 Πράκτορες

Άλλο ένα σημαντικό χαρακτηριστικό θα είναι οι ψηφιακές υπογραφές, οι οποίες είναι κρυπτογραφημένα σύνολα δεδομένων τα οποία χρησιμεύουν στους Η/Υ και στους πράκτορες για να τους επιβεβαιώσουν ότι οι πληροφορίες προέρχονται από μια συγκεκριμένη έμπιστη πηγή. Θέλεις να είσαι σίγουρος ότι μια δήλωση που στάλθηκε στο λογιστικό σου πρόγραμμα ότι χρωστάς χρήματα σε ένα έμπορο δεν είναι μια παραποίηση από κάποιον χάκερ. Οι πράκτορες πρέπει να είναι σκεπτικοί με ισχυρισμούς που διαβάζουν στο Σημαιολογικό Ιστό ώσπου να ελέγξουν τις πηγές των πληροφοριών.

Υπάρχουν ήδη πολλές αυτοματοποιημένες υπηρεσίες χωρίς σημασιολογία, ωστόσο άλλα προγράμματα όπως οι πράκτορες δεν έχουν κανένα τρόπο να εντοπίσουν κάποιον που θα κάνει αυτή τη συγκεκριμένη λειτουργία. Αυτή η διαδικασία ονομάζεται εύρεση υπηρεσίας και μπορεί να πραγματοποιηθεί μόνο όταν υπάρχει μια κοινή γλώσσα να περιγράψει μια υπηρεσία, με τέτοιο τρόπο ώστε να επιτρέπει στους άλλους πράκτορες να αντιληφθούν τη λειτουργία που προσφέρεται και πως μπορούν να την εκμεταλλευτούν. Οι υπηρεσίες και οι ατζέντες μπορούν να

διαφημίσουν αυτή τη λειτουργία αποθηκεύοντας τέτοιες περιγραφές σε ευρετήρια ανάλογα με τις «Κίτρινες Σελίδες».

Είναι διαθέσιμα κάποια -χαμηλού επιπέδου- σχέδια, εύρεσης υπηρεσίας όπως το Microsoft's Universal Plug και το Play, τα οποία στοχεύουν στη σύνδεση διαφορετικού τύπου συσκευών και η Sun Microsystem's Jimi, η οποία στοχεύει στην ένωση υπηρεσιών. Αυτές οι πρωτοβουλίες, αντιμετωπίζουν το πρόβλημα σε επίπεδο δομής ή συντακτικού και βασίζονται στη προτυποποίηση ενός προαποφασισμένου συνόλου λειτουργικών περιγραφών. Η προτυποποίηση δεν μπορεί να προχωρήσει περαιτέρω γιατί δεν μπορούμε να προβλέψουμε τις μελλοντικές ανάγκες. Ο Σημασιολογικός Ιστός, αντιθέτως, είναι πιο ευέλικτος. Οι πράκτορες του καταναλωτή και του παραγωγού μπορούν να φθάσουν σε μια κοινή συνεννόηση ανταλλάσσοντας οντολογίες, οι οποίες παρέχουν το λεξιλόγιο που απαιτείται για την μεταξύ τους συνεννόηση. Οι ατζέντες μπορούν ακόμα, αυτοδύναμα, να βρουν νέες δυνατότητες υποστήριξης όταν ανακαλύψουν νέες οντολογίες. Η σημασιολογία μας διευκολύνει στο να εκμεταλλευτούμε μια υπηρεσία η οποία ταιριάζει μόνο μερικώς σε μια αίτηση. Μια τυπική διαδικασία θα περιέχει τη δημιουργία μιας «αλυσίδας τιμών» στην οποία οι μονάδες της πληροφορίας διαβιβάζονται από τον ένα ατζέντη στον άλλο, κάθε ένας θα προσθέτει μία τιμή, για να κατασκευαστεί το τελικό προϊόν που απαιτείται από το χρήστη. Κατά τη δημιουργία σύνθετων αλυσίδων τιμών αυτόματα με τη ζήτηση, ορισμένοι ατζέντες θ' αναπτύξουν τεχνολογίες τεχνητής νοημοσύνης επιπρόσθετα με τον Σημασιολογικό Ιστό. Ο Σημασιολογικός Ιστός θα παρέχει την υποδομή και το πλαίσιο εργασίας για να γίνουν αυτού του είδους οι τεχνολογίες πιο εφικτές.

Συγκεντρώνοντας όλα αυτά τα στοιχεία, θα βρούμε το αποτέλεσμα που ζητούσαν ο Pete και η Lucy στο αρχικό σενάριο. Οι ατζέντες τους θα έχουν στείλει τμηματικά την εργασία σε άλλες υπηρεσίες και οι ατζέντες τους θα έχουν βρει μια, μέσω διαφήμισης υπηρεσιών. Π.χ. θα μπορούσαν να είχαν χρησιμοποιήσει μια έμπιστη υπηρεσία από την οποία θα έπαιρναν μια λίστα προμηθευτών και θα αποφάσιζαν ποιοι από αυτούς ταιριάζουν με τη λίστα των φυσιοθεραπευτών της ασφάλειας και με τα μαθήματα θεραπείας. Τη λίστα με τους προμηθευτές θα την παρείχε μια άλλη υπηρεσία έρευνας κ.τ.λ. Αυτές οι ενέργειες, που είναι διαμορφωμένες σε αλυσίδες και με τις οποίες ένα μεγάλο ποσοστό δεδομένων διανέμεται στο δίκτυο, έχουν σταδιακά μειωθεί σε ένα μικρό αριθμό δεδομένων με

την έρευνα του Pete και της Lucy, δημιουργώντας έτσι ένα πλάνο για τα ραντεβού που ταιριάζουν με το πρόγραμμα τους και τις άλλες απαιτήσεις.

Στο επόμενο βήμα, ο Σηματολογικός Ιστός θα περάσει από τον εικονικό στον πραγματικό κόσμο. Τα URI μπορούν να αποτελέσουν δείκτες για τα πάντα, συμπεριλαμβανομένου και των φυσικών οντοτήτων, που σημαίνει ότι μπορούμε να χρησιμοποιήσουμε την RDF για να περιγράψουμε συσκευές όπως τα τηλέφωνα και οι τηλεοράσεις. Τέτοιες συσκευές μπορούν να διαφημίσουν τη λειτουργικότητα τους, τι μπορούν να κάνουν και πως να ελεγχθούν, όπως οι πράκτορες λογισμικού. Μια τέτοια σηματολογική προσέγγιση, είναι πιο ευέλικτη από σχέδια χαμηλού επιπέδου όπως το Universal Plug και το Play. Π.χ. αυτό που αποκαλούμε σήμερα αυτοματισμό απαιτεί αυστηρή διευθέτηση των συσκευών που επιθυμούμε να εργαστούν ταυτόχρονα. Οι σηματολογικές περιγραφές των δυνατοτήτων των συσκευών και των λειτουργιών τους θα μας βοηθήσει στο να πετύχουμε αυτοματισμό σε υψηλό επίπεδο, με την μικρότερη ανθρώπινη μεσολάβηση. Ένα συνηθισμένο παράδειγμα συμβαίνει όταν ο Pete απαντά στο τηλέφωνο και η ένταση του στερεοφωνικού χαμηλώνει. Από το να έχει ένα πρόγραμμα για κάθε συσκευή, θα μπορούσε να προγραμματίσει μόνιμα μια τέτοια λειτουργία, για να καλύψει κάθε τοπική συσκευή που έχει καταχωρήσει ότι έχει ένταση ήχου όπως η τηλεόραση, το DVD Player και τα media players στον φορητό Η/Υ που έφερε στο σπίτι από τη δουλειά το απόγευμα. Τα πρώτα σημαντικά βήματα έχουν γίνει σ' αυτήν την περιοχή με την προσπάθεια ανάπτυξης ενός προτύπου που θα περιγράφει τις δυνατότητες λειτουργίας των συσκευών καθώς και τις προτιμήσεις του χρήστη. Βασισμένο στην RDF, αυτό το πρότυπο ονομάζεται Composite Capability/Preference Profil (CC/PP).

Αρχικά θα επιτρέπει σε κινητά τηλέφωνα και σε άλλους πελάτες του Ιστού να περιγράψουν τα χαρακτηριστικά τους ώστε το περιεχόμενο του Ιστού να σχεδιαστεί άμεσα γι' αυτούς. Αργότερα, μόλις προσθέσουμε τη πλήρη προσαρμοστικότητα των γλωσσών για τη διαχείριση των οντολογιών και της λογικής, τα μηχανήματα θα μπορούν αυτόματα να αναζητήσουν και να προσλάβουν υπηρεσίες και άλλες συσκευές για επιπρόσθετες πληροφορίες και λειτουργικότητα.

1.2.6 Η Εξέλιξη της Γνώσης

Ο Σημασιολογικός Ιστός δεν αποτελεί μονάχα το εργαλείο για τη κατασκευή ατομικών εργασιών. Όταν είναι σωστά σχεδιασμένος μπορεί να βοηθήσει στην εξέλιξη της ανθρώπινης γλώσσας συνολικά.

Η ανθρώπινη προσπάθεια κινείται μέσα σε μια εσωτερική τάση ανάμεσα στην αποτελεσματικότητα μικρών συνόλων που λειτουργούν ελεύθερα και την ανάγκη να συνεργαστεί με την ευρύτερη κοινωνία. Ένα μικρό σύνολο μπορεί να κινηθεί πιο γρήγορα και έξυπνα.

Ο συντονισμός ενεργειών σε ένα μεγάλο σύνολο ανθρώπων είναι οδυνηρά αργός και απαιτείται μία σειρά επικοινωνιών. Οι παγκόσμιες εργασίες κινούνται ανάμεσα στα δύο αυτά άκρα, με μια τάση να ξεκινούν από τα μικρά και να καταλήγουν σε μεγαλύτερα σύνολα.

Μια αναγκαία επεξεργασία είναι η ένωση όλων των υποκατηγοριών μιας κουλτούρας, όταν μια ευρύτερα κοινή γλώσσα απαιτείται. Συχνά δύο σύνολα αναπτύσσουν ανεξάρτητα παρόμοιες έννοιες και η περιγραφή, της μεταξύ τους σχέσης αποφέρει σημαντικά οφέλη. Π.χ. ένα Φιλανδό-Αγγλικό λεξιλόγιο, οι σχέσεις επιτρέπουν την επικοινωνία και τη συνεργασία ακόμα και όταν η ομοιότητα μιας ιδέας δεν μας οδηγεί σε ομοιότητα των όρων.

Ο Σημασιολογικός Ιστός, με την ονομασία κάθε έννοιας απλά με ένα URI, επιτρέπει στον καθένα να εκφράζει και να εφευρίσκει νέες έννοιες με ελάχιστη προσπάθεια. Η ενοποιητική γλώσσα δίνει τη δυνατότητα στις ιδέες αυτές να συνδέονται σταδιακά με τον Παγκόσμιο Ιστό. Η δομή αυτή θα διευρύνει τη γνώση και τις ανθρώπινες εργασίες σε ουσιαστικές αναλύσεις από πράκτορες λογισμικού, παρέχοντας μια νέα τάξη εργαλείων μέσω των οποίων θα μπορούμε να ζήσουμε, να εργαστούμε και να μαθαίνουμε μαζί.[2]

2 ΚΕΦΑΛΑΙΟ: ΣΗΜΑΣΙΟΛΟΓΙΚΕΣ ΥΠΗΡΕΣΙΕΣ ΙΣΤΟΥ

2.1 Σημασιολογικές Υπηρεσίες Δικτύου: Περιγραφή – Απαιτήσεις -Σύγχρονες Τεχνολογίες

2.1.1 Εισαγωγή

Οι υπηρεσίες δικτύου επεκτείνουν την έννοια του δικτύου από πηγή διανεμημένων πληροφοριών σε πηγή διανεμημένων υπηρεσιών. Αυτό που ουσιαστικά πρόσθεσε το Σημασιολογικό Δίκτυο στο συνολικό περιεχόμενο του Ιστού, είναι πληροφορίες τις οποίες μπορεί να κατανοήσει και να μεταφράσει ένας υπολογιστής ώστε να είναι δυνατή η «έξυπνη» πρόσβαση σε ετερογενείς και διανεμημένες πληροφορίες. Παρομοίως οι έννοιες του σημασιολογικού δικτύου χρησιμοποιούνται για να ορίσουν νοήμων υπηρεσίες δικτύου, π.χ. υπηρεσίες που θα υποστηρίζουν την αυτόματη εύρεση, σύνθεση, επίκληση και την εσωτερική λειτουργικότητα. Αυτή η συνδυασμένη εφαρμογή των εννοιών του σημασιολογικού δικτύου και των υπηρεσιών δικτύου με σκοπό την υλοποίηση των νοήμων υπηρεσιών δικτύου, αναφέρεται συνήθως ως σημασιολογικές υπηρεσίες δικτύου.

Εξαιτίας του τεράστιου ενδεχόμενου αντίκτυπου των σημασιολογικών υπηρεσιών δικτύου σε χώρους όπως οι επιχειρησιακές εφαρμογές ενοποίησης και το ηλεκτρονικό εμπόριο, έχουν γίνει πολλές προσπάθειες, τόσο ακαδημαϊκές όσο και βιομηχανικές, με σκοπό να φτάσουν τις σημασιολογικές υπηρεσίες δικτύου στο μέγιστο των δυνατοτήτων τους. Αυτές οι κινήσεις καλύπτουν διαφορετικές πτυχές των απαιτήσεων για τη πραγματοποίηση των σημασιολογικών υπηρεσιών δικτύου. Κάποιες φορές είναι απλά συμπληρωματικές, αλλά από ότι φαίνεται συγκρούονται εξαιτίας των διαφορετικών προσεγγίσεων. Αυτές οι ενέργειες στοχεύουν στη βελτίωση της σημερινής τεχνολογίας υπηρεσιών δικτύου, γύρω από το SOAP, τη WSDL και τη UDDI, που δεν παρέχουν ουσιαστική υποστήριξη στον αυτοματισμό των υπηρεσιών.

Μία από αυτές τις ενέργειες είναι το Web Services Modeling Framework (WSMF), το οποίο στοχεύει στη δημιουργία ενός κατάλληλου εννοιολογικού μοντέλου για τη σύνθεση, την ανάπτυξη και τη περιγραφή υπηρεσιών.[3]

Ένα άλλο εγχείρημα που πραγματοποιείται είναι η DAML-S, μία προσπάθεια από την DARPA να περιγράψει την οντολογία των υπηρεσιών δικτύου, με στόχο τη δημιουργία υπηρεσιών δικτύου που θα μπορούν να κατανοήσουν οι υπολογιστές[4].

Η BPEL4WS[5] και η BPML[6]/ WSCI[7], έχουν παρόμοιες λειτουργίες και οι δύο στοχεύουν στον ορισμό μιας γλώσσας που θα περιγράφει τα μοντέλα διαδικασίας καθώς και τη κοινή επιφάνεια διαδικασίας με το σχέδιο υποστήριξης υπηρεσιών, προκειμένου να παρέχει διαλεκτικά και ενδολειτουργικά μέσα για τις υπηρεσίες δικτύου.

Σε ότι αφορά τις ενέργειες του W3C σε αυτό το τομέα, η πρωτοβουλία που ανέλαβε να ορίσει ένα σύνολο από απαιτήσεις πάνω στις περιγραφές των υπηρεσιών δικτύου, δίνει λίγη έως καθόλου βάση στη σημασιολογική υποστήριξη. Για αυτό το λόγο προσφέρει μια αδύναμη βάση στην επίτευξη του αυτοματισμού των λειτουργιών.

Ανάμεσα στις προσεγγίσεις που έχουν παρουσιαστεί μέχρι στιγμής, η WSMF είναι η μόνη που έχει μεγάλα περιθώρια δράσης, καθώς περιγράφει ένα ευέλικτο πλαίσιο εργασίας με σκοπό να πραγματοποιήσει τις σημασιολογικές υπηρεσίες δικτύου. Παρόλα αυτά η ουσιαστική υλοποίηση των θεωρητικών απαιτήσεων που παρουσιάζει είναι ακόμα υπό ανάπτυξη στο πλαίσιο των χρηματοδοτούμενων εργασιών SWWS από την Ευρωπαϊκή Ένωση¹. Η DAML-S εστιάζει στη παροχή σημασιολογικών στοιχείων για τις περιγραφές των υπηρεσιών δικτύου, παρόλο που κάποιες ελλείψεις και περιορισμοί έχουν εντοπιστεί μέσα στη προτεινόμενη οντολογία. Οι αρχικές ενέργειες εστίαζαν στη μοντελοποίηση των επιχειρηματικών διαδικασιών, η BPEL4WS και η BPML/ WSCI δεν ενσωματώνουν σημασιολογικά στοιχεία στα αρχικά τους μοντέλα, ούτε στις προσωπικές αλλά ούτε και στις δημόσιες διαδικασίες. Για αυτό το λόγο δεν μπορούν να υποστηρίξουν τη δυναμική εύρεση, σύνθεση και επίκληση[8].

Όποια και αν είναι η προσέγγιση ή ο σκοπός, κάθε πρωτοβουλία βασίζεται σε ένα συγκεκριμένο τρόπο περιγραφής υπηρεσιών. Η εύρεση, η σύνθεση, η επίκληση και η ενδολειτουργικότητα βασίζονται κυρίως στο πως περιγράφονται οι υπηρεσίες και στο πως εκτίθενται για περαιτέρω χρήση. Ο τρόπος με τον οποίο περιγράφεται μια υπηρεσία καθορίζει σε τι έκταση άλλες δομές μπορούν να προσφέρουν αυτόματη υποστήριξη.

¹ Semantic Web enabled Web Services. <http://swws.semanticweb.org/>

Η WSMF ένα πλαίσιο περιγραφής υπηρεσιών που θα εκπληρώνει τις απαιτήσεις για σημασιολογικές υπηρεσίες δικτύου. Βέβαια χρειάζεται περαιτέρω ανάλυση καθώς και ένα συγκεκριμένο υπόβαθρο από τις προτεινόμενες έννοιες περιγραφής. Για αυτό σε αυτή τη προσέγγιση θεωρούμε αυτό το πλαίσιο εργασίας ως αρχικό σημείο για να καθορίσουμε τις απαιτήσεις για μια ουσιαστική περιγραφή υπηρεσίας. Θα παρουσιάσουμε, θεωρώντας τη WSMF ως βάση, ένα σύνολο προϋποθέσεων που απαιτούνται για τη περιγραφή υπηρεσιών δικτύου, καθώς και τις κατευθυντήριες γραμμές για την επιλογή υπόβαθρων, λαμβάνοντας υπόψη χαρακτηριστικά από πρόσφατες προσπάθειες.

2.1.2 Δυνατότητες και Περιγραφικές Απαιτήσεις της Εύρεσης και της Σύνθεσης

Η αυτόματη εύρεση και σύνθεση των υπηρεσιών αποτελούν ίσως τις μεγαλύτερες προκλήσεις που αντιμετωπίζουν οι σημασιολογικές υπηρεσίες δικτύου. Η εύρεση του κατάλληλου τρόπου να συνδυαστούν αυτά τα δύο χαρακτηριστικά, αποτελεί σημείο «κλειδί» για την ενσωμάτωση του Δικτύου σε μια διανεμημένη πηγή υπολογισμού, καθώς καθιστούν ικανές τη τοποθεσία και το συνδυασμό των διανεμημένων υπηρεσιών να εκτελέσουν την απαραίτητη λειτουργία.

Η αυτόματη εύρεση περιλαμβάνει τον αυτόματο εντοπισμό υπηρεσιών δικτύου που θα παρέχουν μια συγκεκριμένη λειτουργία και θα εμμένουν στις ιδιότητες που ζητούνται. Για να παρέχει μια τέτοια αυτόματη τοποθεσία, η διαδικασία εύρεσης θα πρέπει να βασίζεται στη σημασιολογική ταυτοποίηση της περιγραφής που ζητείται και της περιγραφής της υπηρεσίας που προσφέρεται. Αυτό το πρόβλημα δεν απαιτεί μόνο έναν αλγόριθμο για τη ταυτοποίηση αυτών των περιγραφών, αλλά και μια γλώσσα που θα εκφράζει ξεκάθαρα τις δυνατότητες των υπηρεσιών.

Επιπλέον η σύνθεση των υπηρεσιών δικτύου απαιτεί κάτι παραπάνω από την απλή παρουσίαση συνδυασμών από υπηρεσίες που θα γνωρίζουν εκ των προτέρων τη ροή και τις συνδέσεις. Θα πρέπει επίσης να επιτρέπει στο συνδυασμό υπηρεσιών, να παρέχει μια δεδομένη λειτουργία στη περίπτωση που μια αίτηση δεν μπορεί να πραγματοποιηθεί όταν χρησιμοποιούνται οι διαθέσιμες υπηρεσίες χωριστά.

Η εύρεση και η σύνθεση υπηρεσιών απαιτεί μια προσέγγιση που θα βασίζεται σε σημασιολογικές περιγραφές καθώς ο αιτών θα πρέπει να εκφράζεται σε ένα

υψηλό επίπεδο και με αφηρημένο τρόπο, προκειμένου να ενεργοποιήσει τις διαδικασίες υποστήριξης.

Η εύρεση και η σύνθεση συνεργάζονται με διαφορετικούς τρόπους. Τα παρακάτω παραδείγματα περιγράφουν διαφορετικά σενάρια τα οποία περιλαμβάνουν τη τοποθεσία και το συνδυασμό υπηρεσιών:

— Ένας χρήστης θέλει να κλείσει μια πτήση από την Αθήνα στη Μαδρίτη, με σταθερή ανώτατη τιμή. Έχοντας αυτή τη πληροφορία η εύρεση θα ψάξει για μια υπηρεσία που θα αποδέχεται το τόπο προορισμού, άφιξης, την ημερομηνία και τη μέγιστη τιμή, προσφέροντας μια θέση στη κατάλληλη πτήση. Στη περίπτωση που μια τέτοια υπηρεσία δεν είναι διαθέσιμη, αλλά μπορεί να βρεθεί μια υπηρεσία που ψάχνει πληροφορίες πτήσεων εφόσον διαθέτει τα δεδομένα του ταξιδιού, καθώς και μια άλλη υπηρεσία που μπορεί να κλείσει μια πτήση εφόσον γνωρίζει τις πληροφορίες της πτήσης, σε αυτή τη περίπτωση θα πρέπει να γίνει συνδυασμός των δύο υπηρεσιών. Ο συνδυασμός των δύο υπηρεσιών μπορεί να δώσει τη λειτουργικότητα που ζητείται.

— Ένα γραφείο ταξιδιών μοντελοποιεί την επιχειρηματική του διαδικασία για να παρέχει μια υπηρεσία «φτιάξε ένα ταξίδι». Το πρακτορείο μοντελοποιεί τον έλεγχο και τη ροή των δεδομένων ανάμεσα στις διαφορετικές υπηρεσίες που εμπλέκονται (για να κλείσουν μια πτήση, ένα ξενοδοχείο, ένα αυτοκίνητο). Σε αυτή τη περίπτωση η σύνθεση μοντελοποιείται από τη πλευρά του προμηθευτή. Στη περίπτωση όμως που το πρακτορείο δεν θέλει να περιορίσει την υπηρεσία κράτησης πτήσεων σε μία μόνο εταιρεία, θα πρέπει να δίνεται μια υψηλού επιπέδου περιγραφή της απαιτούμενης υπηρεσίας στο μοντέλο διαδικασίας, ώστε να καταστήσει δυνατή τη δυναμική εύρεση (και σύνθεση αν μια μόνο υπηρεσία δεν μπορεί να ικανοποιήσει τις προϋποθέσεις) από τις καλύτερες διαθέσιμες υπηρεσίες, για να κλείσει μια πτήση με βάση τα κριτήρια του αιτούντα.

Αυτά τα δύο παραδείγματα μας δείχνουν ότι οι βασικές λειτουργίες της εύρεσης και της σύνθεσης είναι να παρέχουν την απαιτούμενη λειτουργικότητα. Αν και αυτές οι περιπτώσεις θα μπορούσαν να μετατραπούν και να επεκταθούν σε έναν μεγάλο αριθμό από παραδείγματα, όλα τους βασίζονται σε μια υψηλού επιπέδου περιγραφή της λειτουργικότητας που αναζητείται.

2.1.3 Περιγραφή των Δυνατοτήτων

Θα μπορούσαμε να δούμε την υψηλού επιπέδου απαιτούμενη περιγραφή λειτουργικότητας ως δυνατότητα της υπηρεσίας. Διαφορετικές υπηρεσίες μπορούν να παρέχουν τις ίδιες δυνατότητες, π.χ. να ψάχνουν ένα βιβλίο και μια ταινία.

Με αυτή την έννοια, οι δυνατότητες θα πρέπει φυσιολογικά να περιγράφονται χωριστά από τις περιγραφές των συγκεκριμένων υπηρεσιών για διάφορους λόγους:

— Εκφράζουν γενικές λειτουργίες: οι περισσότερες υπηρεσίες προσφέρουν την ίδια λειτουργικότητα αλλά όταν έχουν διαφορετικό ραφινάρισμα θα πρέπει να σχετίζονται με το ίδιο γενικό υψηλό επίπεδο δυνατοτήτων.

— Χρησιμοποιούν διαφορετικές ορολογίες: το ραφινάρισμα που γίνεται από μια συγκεκριμένη υπηρεσία δικτύου μπορεί να εκφραστεί μέσω διαφορετικής ορολογίας από αυτή που χρησιμοποιείται για να περιγράψει τις δυνατότητες των υπηρεσιών, γιατί όσο αυξάνεται η ελαστικότητα, η χρήση της ίδιας ορολογίας δεν είναι δυνατή.

— Μια δεδομένη υπηρεσία μπορεί να παρουσιάσει δύο διαφορετικές δυνατότητες, ενώ μπορεί να προβάλλει μια μόνο περιγραφή της υπηρεσίας.

— Υποστηρίζει τη διαδικασία εύρεσης: για να πραγματοποιηθεί η εύρεση χρειάζεται αρχικά τις περιγραφές των δυνατοτήτων. Σε δευτερεύοντα βήματα εκτελούνται το ραφινάρισμα και οι απαιτήσεις. Επομένως ο διαχωρισμός των δυνατοτήτων και των ραφιναρισμάτων της υπηρεσίας που παραπέμπουν σε αυτές, καθιερώνει ένα φυσικό σύνδεσμο με τη διαδικασία εύρεσης.

Η έννοια ανακοίνωση σημαίνει ορισμός των δυνατοτήτων καθώς και ορισμός των ραφιναρισμάτων των υπηρεσιών και του συνδέσμου ανάμεσα στα ραφινάρια και τις δυνατότητες. Αυτή η περιγραφή πρέπει να επιτρέπει την αιτιολόγηση πάνω στη πληροφορία που παρουσιάζεται από την υπηρεσία.

Για να μπορεί να υποστηρίξει τη δυναμική εύρεση και σύνθεση, μια δυνατότητα θα πρέπει να περιλαμβάνει τις παρακάτω πληροφορίες:

-Προϋποθέσεις: τα υψηλού επιπέδου δεδομένα που εισάγονται σε μια υπηρεσία συνοδεύονται από όρους που αφορούν αυτά τα δεδομένα. Αυτά τα δεδομένα είναι έννοιες μιας δεδομένης κυριότητας μιας οντολογίας. Κάθε προϋπόθεση θα περιλαμβάνει ένα μηχανισμό αναγνώρισης που θα επιτρέπει μελλοντικές αναφορές. Με την έννοια υψηλού επιπέδου δεδομένα εννοούμε ότι μπορούμε να βρούμε πιο συγκεκριμένες έννοιες μέσα στην οντολογία, π.χ. υπόδειξη

πληροφοριών πληρωμής ως προϋπόθεση αντί για πληροφορίες πιστωτικών καρτών ή τραπεζών ή και ακόμα δεδομένων. Αν μια ιδιαίτερα συγκεκριμένη έννοια δίνεται σαν προϋπόθεση, τότε μια δυνατότητα δεν θα μπορεί να εκφράσει τις γενικές λειτουργίες. Πρέπει να σημειώσουμε ότι οι προϋποθέσεις μιας δεδομένης δυνατότητας δεν είναι ανεξάρτητες η μια από την άλλη, καθώς όλες ορίζουν τη λειτουργία που εκφράζεται μέσα από μια δυνατότητα.

-Μετά-υποθέσεις: Τα υψηλού επιπέδου αποτελέσματα που προκύπτουν από την εκτέλεση μιας υπηρεσίας, τα οποία συνοδεύονται από τους όρους που αφορούν αυτά τα αποτελέσματα. Τα αποτελέσματα είναι επίσης έννοιες μιας δεδομένης κυριότητας μιας οντολογίας. Ορίζονται και στις μετα-υποθέσεις μηχανισμοί αναγνώρισης και όπως και στις προϋποθέσεις δεν μπορούν να θεωρηθούν ανεξάρτητες καθώς η αφαίρεση έστω και μιας από αυτές αλλάζει τη λειτουργία που εκφράζεται από μια δυνατότητα.

-Περιγραφή που βασίζεται σε κείμενο: όταν μπορεί να γίνει μετάφραση κατανοητή στους ανθρώπους.

-Υπηρεσίες: παραπέμπει στις υπηρεσίες που παρουσιάζουν τη περιγραφόμενη δυνατότητα.

-Μηχανισμός αναγνώρισης: για να πραγματοποιείται η παραπομπή σε κάποια δυνατότητα.

Οι προ και οι μετά υποθέσεις ορίζουν την ικανότητα μιας υπηρεσίας σε όρους της πληροφορίας που χρειάζεται, προκειμένου να χρησιμοποιηθεί η υπηρεσία καθώς και τα αποτελέσματα που προκύπτουν από την επίκληση της υπηρεσίας. Η περιγραφή δυνατοτήτων μέσω της έκφρασης των λειτουργιών τους σε όρους των απαιτούμενων υψηλού επιπέδου εισαχθέντων δεδομένων και αποτελεσμάτων, καλύπτει τις παρακάτω αναγκαίες προϋποθέσεις:

-Μοντελοποίηση μιας διαδικασίας κατά τη διάρκεια σχεδίασης. Σε αυτή τη περίπτωση η ροή των εργασιών και των δεδομένων ορίζεται εκ των προτέρων, τουλάχιστον κατά ένα μέρος. Για αυτό το λόγο η ανακοίνωση της χρήσης μιας δυνατότητας, θα πρέπει να καθιστά ικανή τη προδιαγραφή κατά τη διάρκεια σχεδίασης των δεδομένων που εισάγονται και των αποτελεσμάτων της υπηρεσίας. Αυτή η πληροφορία είναι απαραίτητη για τα μοντέλα δεδομένων και τη ροή ελέγχου. Παρόλα αυτά, αυτή η πληροφορία θα πρέπει να κρατηθεί σε αρκετά γενικό επίπεδο για να μπορεί να περιγράψει γενικές λειτουργίες της υπηρεσίας, επιτρέποντας τη δυναμική τοποθέτηση και συνδυασμό των υπηρεσιών. Παραδείγματος χάριν σε μια

επιχειρηματική διαδικασία όπου χρησιμοποιείται μια υπηρεσία για την αγορά αγαθών και μια άλλη υπηρεσία για τη μεταφορά τους, το αποτέλεσμα της υπηρεσίας που χρησιμοποιήθηκε για την αγορά θα πρέπει να χρησιμοποιηθεί από την υπηρεσία που θα τα μεταφέρει και θα πρέπει να σχεδιαστούν τα απαραίτητα δεδομένα και η ροή ελέγχου.

-Η δυναμική εύρεση: δευτερεύοντες αλγόριθμοι όπως αυτοί που παρουσιάζονται στις αναφορές [9] και [10] υποστηρίζονται, σχετίζοντας τις ανάλογες προ και μετά υποθέσεις με την κατάλληλη κυριότητα οντολογίας καθώς και από τη χρήση συγκεκριμένων ραφιναρισμάτων υπηρεσιών.

-Η δυναμική σύνθεση: για να επιτευχθεί ένας συνδυασμός υπηρεσιών θα πρέπει να εκτελεστεί μια δεδομένη λειτουργικότητα, εκφράζοντας τη σε όρους των προ και μετά υποθέσεων, όπως περιγράφεται στην αναφορά [10].

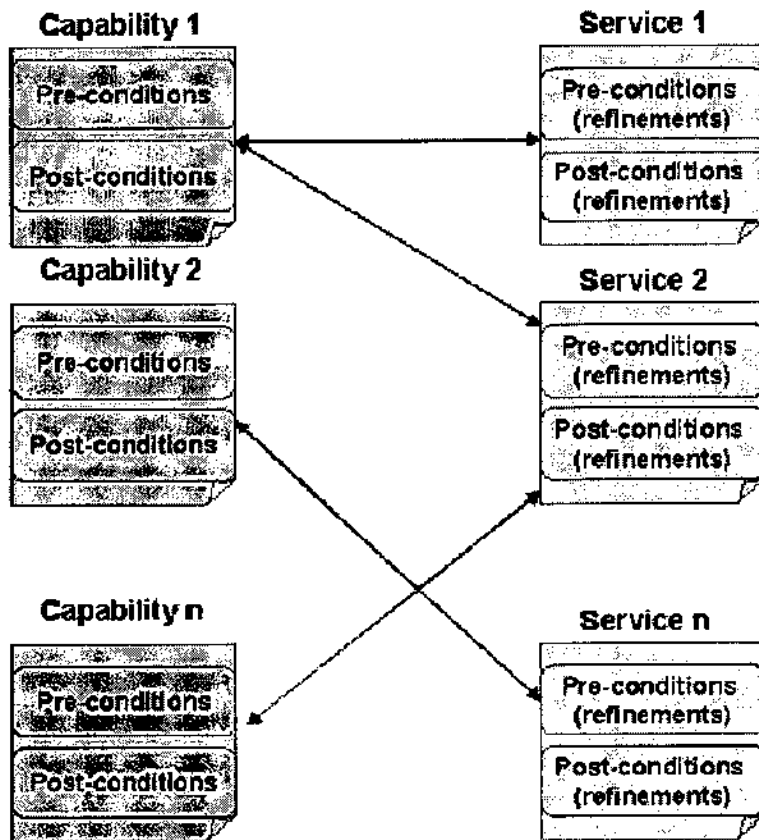
-Αντιστοίχιση από n σε m : όταν περιγράφουμε μια δυνατότητα χρησιμοποιώντας προ και μετά υποθέσεις και μη περιλαμβάνοντας τις χαμηλού επιπέδου πληροφορίες εισαχθέντων και εξαχθέντων δεδομένων, ενεργοποιείται η από n σε m αντιστοίχιση ανάμεσα στις δυνατότητες και τις υπηρεσίες. Έτσι μπορεί να γίνει περιγραφή των γενικών λειτουργιών και η ανακοίνωση των διαφορετικών γενικών λειτουργιών από την ίδια υπηρεσία. Χαμηλότερου επιπέδου εισαχθέντα και εξαχθέντα δεδομένα δεν πρέπει να περιλαμβάνονται στη περιγραφή δυνατοτήτων γιατί κάτι τέτοιο θα απέτρεπε αυτά τα χαρακτηριστικά και θα συνεπάγονταν διάφορα προβλήματα μοντελοποίησης.

Οι πληροφορίες που βασίζονται σε κείμενο χρησιμοποιούνται για να μπορεί ένας χρήστης να φυλλομετρά τις δυνατότητες. Οι δυνατότητες θα πρέπει να γίνονται κατανοητές από τους ανθρώπους και τους υπολογιστές, γιατί μπορεί ένας σχεδιαστής διαδικασιών να χρειαστεί να ψάξει τις κατάλληλες δυνατότητες για να τις συμπεριλάβει στο μοντέλο διαδικασίας κατά την ώρα του σχεδιασμού.

2.1.4 Βελτιώσεις των Δυνατοτήτων

Εφόσον μια δυνατότητα έχει περιγραφεί, οι διαφορετικές υπηρεσίες που παρουσιάζουν αυτή τη δυνατότητα μπορούν να τη βελτιώσουν. Με αυτό τον τρόπο μπορούν να εκφραστούν συγκεκριμένες απαιτήσεις, περιορισμοί και αποτελέσματα μιας ξεχωριστής υπηρεσίας. Για να αποφευχθεί ο πλεονασμός, η υπηρεσία θα

περιγράφει μόνο τις βελτιώσεις που παρουσιάζει, χωρίς να περιγράφει τη πληροφορία που ούτως η άλλως περιλαμβάνεται στη περιγραφή των δυνατοτήτων. Το Σχήμα 2 αναπαριστά τη σχέση ανάμεσα στις δυνατότητες και τις υπηρεσίες.



Σχήμα 2 - Η Σχέση Ανάμεσα στις Δυνατότητες και τις Υπηρεσίες

Όταν ορίζουμε τις βελτιώσεις μιας υπηρεσίας, μια ολοκληρωμένη υψηλού επιπέδου περιγραφή των δεδομένων που εισάγονται, ζητείται από την υπηρεσία και μας δίνεται το αποτέλεσμα της εκτέλεσής της. Γενικά τα πραγματικά δεδομένα είναι απαραίτητα για να περιγράψουμε τις χαμηλού επιπέδου λεπτομέρειες της λειτουργίας μιας υπηρεσίας. Τα δεδομένα που εισάγονται και εξάγονται σχετίζονται με τις προ και μετά υποθέσεις, καθώς συνιστούν την υλοποιημένη μορφή σε όρους δεδομένων των υψηλού επιπέδου δεδομένων.

Επομένως οι πληροφορίες που εκθέτει μια υπηρεσία, η οποία θέλει να βελτιώσει τις γενικές λειτουργίες της, είναι οι παρακάτω:

-Μηχανισμός αναγνώρισης: πληροφορίες για τις αναφορές υπηρεσιών.

-Περιγραφή σε κείμενο: πληροφορίες κατανοητές από τους ανθρώπους.

-Αναφορές δυνατοτήτων: αναφορές πάνω στη δυνατότητα ή τις δυνατότητες που παρουσιάζονται από μια υπηρεσία.

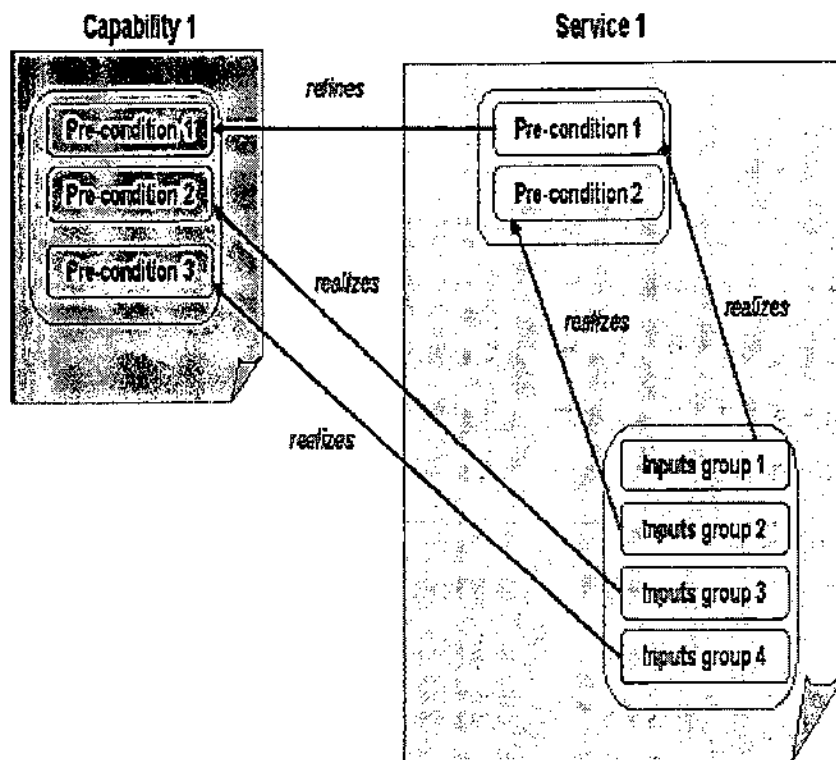
-Βελτιώσεις προϋποθέσεων: αν η βελτίωση μιας ή περισσότερων προϋποθέσεων ορίζεται σε μια δυνατότητα, θα περιλαμβάνεται μια αναφορά σε αυτές. Αυτή είναι η περίπτωση όπου μια υπηρεσία αναφέρεται σε μια δυνατότητα, ζητώντας γενικές πληροφορίες πληρωμής σαν προϋπόθεση, ενώ η υπηρεσία δέχεται πληροφορίες μόνο σχετικά με πιστωτικές κάρτες. Επιτρέπονται νέες προϋποθέσεις και περιλαμβάνονται μηχανισμοί αναγνώρισης για κάθε νέα προϋπόθεση ή βελτίωση. Γενικά οι βελτιώσεις των προϋποθέσεων αντανakλούν τη δύναμη των προϋποθέσεων.

-Εισαχθέντα δεδομένα: τα δεδομένα που εισάγονται ομαδοποιούνται και αναφέρονται στις προϋποθέσεις, είτε μέσω των γενικών δυνατοτήτων είτε μέσω των βελτιώσεων της υπηρεσίας. Για να μπορούμε να έχουμε πολυμορφία, διαφορετικά σύνολα από εισαχθέντα δεδομένα μπορούν να οριστούν για την ίδια προϋπόθεση. Παραδείγματος χάριν στη περίπτωση μιας υπηρεσίας που δέχεται διαφορετικούς τρόπους πληρωμών, τα δεδομένα που απαιτούνται εξαρτώνται από το πώς ο αιτών θέλει να πληρώσει. Γενικά είναι λογικό να ορίζεται ένα μόνο interface για την υπηρεσία. Επομένως αυτή η υπηρεσία θα έχει μία προϋπόθεση «πληροφορίες πληρωμής», με διαφορετικά σύνολα δεδομένων εισαγωγής για πληρωμή μέσω πιστωτικής, μεταφορά χρημάτων κ.τ.λ.

-Βελτιώσεις μετά-υποθέσεων: ορίζονται ακολουθώντας τον ίδιο μηχανισμό με τις προ-υποθέσεις. Οι βελτιώσεις των ήδη υπαρχόντων μετά-υποθέσεων αντανakλούν την αδυναμία των δυνατοτήτων των μετά-υποθέσεων, ενώ οι βελτιώσεις των καινούργιων αντανakλούν την ενδυνάμωση της δυνατότητας των μετά υποθέσεων.

-Δεδομένα που εξάγονται: περιγράφονται ομοίως με τα δεδομένα που εισάγονται.

Στο Σχήμα 3 φαίνονται οι βελτιώσεις των προ- υποθέσεων και πως υλοποιούνται μέσα από τα δεδομένα εισαγωγής:



Σχήμα 3 - Υλοποίηση και Βελτιστοποίηση των Προϋποθέσεων

Στο παραπάνω Σχήμα, η προϋπόθεση της υπηρεσίας 1 βελτιώνει τη πρώτη προϋπόθεση που ορίζεται μέσα στη δυνατότητα. Τα δεδομένα εισαγωγής της ομάδας 1 δίνουν τα πραγματικά δεδομένα για τη βελτιωμένη προ- υπόθεση. Η προ- υπόθεση της υπηρεσίας 2 είναι μια νέα προ- υπόθεση, που δεν περιλαμβάνεται στη δυνατότητα που προβάλλεται, και υλοποιείται από τα δεδομένα εισαγωγής της δεύτερης ομάδας. Οι προ- υποθέσεις δυνατότητας 2 και 3 δεν έχουν μετατραπεί, άρα τα εισαχθέντα δεδομένα των ομάδων 3 και 4 αναφέρονται σε αυτές τις προ- υποθέσεις. Όπως είπαμε παραπάνω η βελτίωση μιας ή περισσότερων προ- υποθέσεων μαζί, είναι πιθανή.

Μέσω του ορισμού των δυνατοτήτων, η περιγραφή μιας υπηρεσίας προβάλλει αρκετές πληροφορίες για να καταστήσει δυνατή την αυτόματη εύρεση και σύνθεση. Οι βελτιώσεις και οι πληροφορίες πάνω στα δεδομένα που εισάγονται και εξάγονται μπορούν επίσης να χρησιμοποιηθούν στον εντοπισμό της επιθυμητής υπηρεσίας, όπως και στη σύνθεση εφόσον διαθέτουν την κατάλληλη εκφραστική δύναμη για τον αιτούντα. Έτσι ο αιτών δεν περιορίζεται μόνο στον εντοπισμό και το συνδυασμό γενικών δυνατοτήτων, αλλά μπορεί επίσης να εκφράσει και πιο λεπτομερείς ανάγκες.

2.1.5 Η Σχέση με τις Σημερινές Τεχνολογίες

Παρόλο που απαιτήσεις που παρουσιάστηκαν παραπάνω για την περιγραφή των λειτουργιών μιας υπηρεσίας αντανakλούν τις ιδέες που περιέχονται στην WSMF προσέγγιση, μπορούν να επεκταθούν και να βελτιώσουν τα μέσα περιγραφής που σκιαγραφούνται στο πλαίσιο εργασίας.

Μόλις αυτές οι επεκτάσεις και οι βελτιώσεις οριστούν θα πρέπει να εκφραστούν χρησιμοποιώντας τη κατάλληλη οντολογία, που θα παρέχει σημασιολογικά στοιχεία στη πληροφορία, που προβάλλεται από μια υπηρεσία. Με αυτή την έννοια ούτε το BPEL4WS αλλά ούτε και το BPML/WSCI περιλαμβάνουν κάποιο κατάλληλο μηχανισμό, καθώς δεν χρησιμοποιούν παρόμοιες έννοιες στις δυνατότητες ή στις βελτιώσεις και δεν προσθέτουν καμία σημασιολογική πληροφορία.

Για αυτό το λόγο η DAML-S είναι η μόνη επαναχρησιμοποιήσιμη πιθανή εργασία που μπορεί να ορίσει την επιθυμητή οντολογία. Η υπάρχουσα οντολογία υπηρεσιών περιλαμβάνει το προφίλ και τα μοντέλα υπηρεσιών, οι επιδιώξεις των οποίων είναι παρόμοιες με αυτές των δυνατοτήτων και των βελτιώσεων. Ωστόσο η οντολογία της DAML-S παρουσιάζει σοβαρούς περιορισμούς αν παραμείνει όπως είναι. Καταρχάς περιλαμβάνονται στο προφίλ τα εισαχθέντα και εξαχθέντα δεδομένα, αποτρέποντας τη πολυμορφία και τη χαρτογράφηση από n σε m ανάμεσα στο προφίλ και σε ορισμένα μοντέλα υπηρεσιών. Κατά δεύτερον, οι προ και μετά υποθέσεις του ολοκληρωμένου μοντέλου υπηρεσίας δεν σχετίζονται με αυτές που παρουσιάζονται στο προφίλ. Κατά τρίτον τα εισαχθέντα και εξαχθέντα δεδομένα δεν σχετίζονται με τις προ- υποθέσεις και τα αποτελέσματα. Τέλος η χρήση διαφορετικών συνόλων δεδομένων για μια δεδομένη προ ή μετά υπόθεση, δεν επιτρέπεται. Όλοι αυτοί οι περιορισμοί υποδεικνύουν προβλήματα στη μοντελοποίηση της υπηρεσίας.

Συμπεραίνοντας, η οντολογία DAML-S μπορεί να χρησιμοποιηθεί σαν βάση για τον ορισμό σημασιολογικών στοιχείων, για τις περιγραφές της λειτουργίας μιας υπηρεσίας, αλλά θα πρέπει να είναι εξαιρετικά αλλαγμένη και εκτεταμένη ώστε να μπορεί να υποδεικνύει τις ιδιότητες και τις απαιτήσεις που παρουσιάστηκαν.

2.1.6 Διαλειτουργικότητα και οι Σημερινές Τεχνολογίες

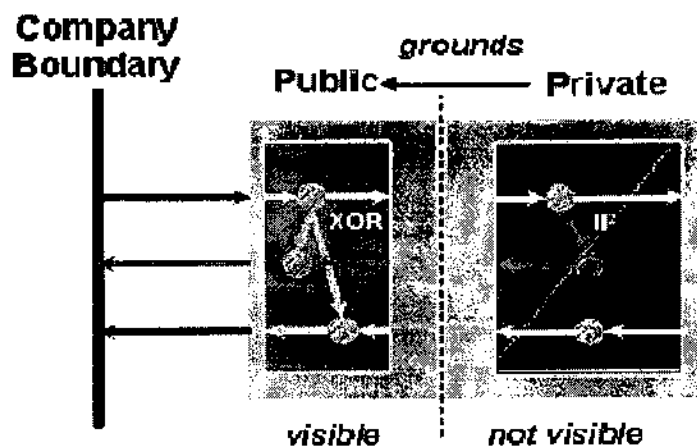
Ένας από τους κύριους σκοπούς των υπηρεσιών δικτύου είναι ο αυτοματισμός των εφαρμογών ενοποίησης εσωτερικά αλλά και κατά μήκος των οργανωτικών ορίων. Αυτό συνεπάγεται την ανάγκη για ύπαρξη διαλειτουργικότητας ανάμεσα στις υπηρεσίες. Διαλειτουργικότητα μπορεί να υπάρξει ανάμεσα στις υπηρεσίες ενός οργανισμού ή κατά μήκος διαφορετικών οργανωτικών ορίων. Για να εξασφαλίσουμε την αυτόματη διαλειτουργικότητα, τα μέσα περιγραφής θα πρέπει να ορίζονται από σημασιολογικά στοιχεία.

Οι επαγγελματικές συνεργασίες απαιτούν μακροπρόθεσμες αλληλεπιδράσεις καθοδηγούμενες από το μοντέλο διαδικασίας. Γιατί μια υπηρεσία πρέπει να μοντελοποιήσει τις επιχειρηματικές της διαδικασίες, που θα περιέχουν τους μηχανισμούς απόφασης για την εκτέλεση της υπηρεσίας. Ακολουθώντας όμως μία από τις βασικές αρχές της WSMF, καμία λεπτομέρεια, που αφορά την επιχειρηματική λογική που ακολουθείται από ένα οργανισμό, δεν θα πρέπει να είναι δημοσίως ορατή. Για αυτό παρόλο που το μοντέλο διαδικασίας, τα δεδομένα του και η ροή ελέγχου σχεδιάζονται για να υποστηρίξουν την εκτέλεση αλλά και τη «δημόσια» συμπεριφορά μιας υπηρεσίας, δεν θα πρέπει να εκτίθενται.

Γενικά η εξωτερική συμπεριφορά μιας υπηρεσίας σε όρους ανταλλαγής μηνυμάτων πρέπει να γίνει κοινή (δημόσια) προκειμένου να καταστεί δυνατή η αυτόματη διαλειτουργικότητα της υπηρεσίας με κάθε άλλη υπηρεσία.

Με αυτή την έννοια, η «δημόσια» περιγραφή μιας υπηρεσίας πρέπει να περιλαμβάνει ένα διαλλακτικό interface το οποίο θα επιτρέπει τη διαλειτουργικότητα, ενώ παράλληλα δεν θα αποκαλύπτει καμία εμπιστευτική λεπτομέρεια.

Το Σχήμα 4 δείχνει τη σχέση ανάμεσα στο ιδιωτικό μοντέλο διαδικασίας και στο δημόσιο μοντέλο διαδικασίας σε όρους προβολής. Το ιδιωτικό μοντέλο διαδικασίας υποστηρίζει το δημόσιο μοντέλο και καθορίζει τη συμπεριφορά του, αλλά μόνο το δημόσιο μοντέλο διαδικασίας γίνεται δημόσιο.



Σχήμα 4 - Η Προβολή Ατομικών και Δημόσιων μοντέλων Διαδικασίας

Σε ότι αφορά τα ιδιωτικά μοντέλα διαδικασίας τόσο η BPEL4WS όσο και η BPML προσφέρουν ένα πλούσιο σύνολο πρωταρχικών στοιχείων στη μοντελοποίηση της ροής εργασίας της υπηρεσίας, υποστηρίζοντας σύνθετες διαδικασίες, που βασίζονται σε υπηρεσίες δικτύου. Στην αναφορά [11] και [12] μπορούμε να βρούμε μία ανάλυση βασισμένη σε ένα μοτίβο, και των δύο γλωσσών. Η συγκεκριμένη εργασία αναλύει τις δύο γλώσσες χρησιμοποιώντας ένα σύνολο από μοτίβα για τη ροή εργασίας και την επικοινωνία για να διευκρινίσει αν παρέχουν επαρκή αρχικά στοιχεία μοντελοποίησης για κάθε πιθανή θεωρητική περίπτωση. Το αποτέλεσμα είναι παρόμοιο και για τις δύο, καθώς υποστηρίζουν τα περισσότερα μοτίβα από αυτά που περιγράφονται.

Οι δύο προσεγγίσεις διαχωρίζουν ξεκάθαρα τα ιδιωτικά και τα δημόσια μοντέλα διαδικασίας. Η BPEL4WS εισάγει την έννοια της εκτελέσιμης διαδικασίας για ιδιωτικές διαδικασίες και της θεωρητικής διαδικασίας για τις δημόσιες. Ομοίως η BPML χρησιμοποιείται στη μοντελοποίηση ιδιωτικών διαδικασιών. Ωστόσο και οι δυο γλώσσες στερούνται σημασιολογικών στοιχείων ώστε να μπορούν να προβάλλουν το δημόσιο interface καθώς και τη πιθανότητα εξέφρασης της χρήσης μιας υπηρεσίας μέσα στο ιδιωτικό μοντέλο διαδικασίας σε όρους της δυνατότητας που παρουσιάζει.

Για αυτό το σκοπό απαιτείται και η ανάλυση της DAML-S. Βέβαια η κατάλληλη οντολογία υπηρεσίας απαιτεί ένα πλουσιότερο σύνολο αρχικών στοιχείων μοντελοποίησης διαδικασιών, γιατί οι έννοιες που περιγράφονται στο μοντέλο διαδικασίας της DAML-S δεν είναι αρκετά ισχυρές για να υποστηρίξουν ορισμένα από τα μοτίβα επικοινωνίας και ροής εργασίας που απαιτούνται. Επιπλέον η DAML-

S δεν κάνει το διαχωρισμό ανάμεσα στις ιδιωτικές και δημόσιες διαδικασίες, επιτρέποντας στις εσωτερικές λεπτομέρειες να προβάλλονται μέσω μιας σύνθετης διαδικασίας. Παρόλο που η DAML-S παρέχει οντολογική υποστήριξη στη μοντελοποίηση διαδικασιών, οι περιορισμοί που έχει αποτρέπουν την άμεση χρήση της στην έκδοση διαλειτουργικών πληροφοριών.

Η δική μας πρόταση είναι να προσθέσουμε σημασιολογικά στοιχεία είτε στο μηχανισμό μοντελοποίησης BPEL4WS είτε στο BPML/WSCI για διαλλακτικό interface καθώς και ενοποίηση αυτών των σημασιολογικών στοιχείων στην οντολογία της υπηρεσίας. Αυτό σημαίνει αντικατάσταση του μοντέλου διαδικασίας στην DAML-S από ένα μοντέλο βασισμένο είτε στην BPEL4WS είτε στην BPML/WSCI, συμπεριλαμβάνοντας τις απαραίτητες δημόσιες πληροφορίες για να προβάλλει τη συμπεριφορά της υπηρεσίας σε ότι αφορά τα μηνύματα που ανταλλάσσονται.

Συνοψίζοντας η δημόσια περιγραφή απαιτήσεις για διαλειτουργικότητα είναι οι εξής:

- Θα πρέπει να περιγράφεται η εκτεταμένη συμπεριφορά της υπηρεσίας σε όρους ανταλλαγής των μηνυμάτων καθώς και η ακολουθία των μηνυμάτων.

- Δεν θα πρέπει να προβάλλεται καμία πληροφορία που αφορά την εσωτερική επιχειρηματική λογική.

- Η δημόσια διαδικασία που προβάλλεται θα πρέπει να υποστηρίζεται από μια κατάλληλη ατομική διαδικασία, η οποία θα πρέπει να επιτρέπει τη χρήση δυνατοτήτων και βελτιώσεων κατά τη διάρκεια σχεδιασμού για να προσδιορίσει την υπηρεσία που θα χρησιμοποιηθεί καθώς και τη δυναμική εύρεση και σύνθεσή της.

Στόχος μας δεν είναι να αποφασίσουμε ποια από τις δύο γλώσσες μπορεί να χρησιμοποιηθεί για να εκπληρώσει αυτές τις απαιτήσεις. Σε κάθε περίπτωση θα πρέπει να υποστηρίζεται σημασιολογικά και να είναι εκτεταμένη ώστε να χρησιμοποιεί τις γενικές δυνατότητες και βελτιώσεις κατά τη διάρκεια σχεδιασμού με τον τρόπο που περιγράφηκε παραπάνω.

Επίσης αναλύεται η χρήση του Abstract State Machines (ASMs) για τη μοντελοποίηση επιχειρηματικών διαδικασιών, καθώς παρουσιάζει διάφορες ενδιαφέρουσες ιδιότητες. Χάρη στις ιδιότητες του, μπορεί να περιγράψει τα διαλλακτικά interfaces και τις αντίστοιχες υποστηρίξεις, παρόλα αυτά καμιά από τις προσπάθειες που γίνονται τώρα δεν χρησιμοποιεί ASMs.

2.1.7 Επίκληση μιας Υπηρεσίας

Οι προϋποθέσεις που παρουσιάστηκαν μέχρι στιγμής ασχολούνται με την έκφραση μιας εμφαντικής λειτουργικότητας και την έκδοση ενός διαλλακτικού interface. Η πληροφορία επίκλησης που παρουσιάζεται από μια δεδομένη υπηρεσία θα πρέπει να συμβαδίζει με τις τεχνολογίες που θα την υποστηρίξουν. Σε κάθε περίπτωση, οι λεπτομέρειες θα πρέπει να είναι διαθέσιμες τη τρέχουσα στιγμή για τον αιτών για να εκτελέσει μια επίκληση. Αυτές οι λεπτομέρειες θα πρέπει να σχετίζονται κάθε πτυχή της εμφαντικής λειτουργικότητας με ένα μηχανισμό υποστήριξης, π.χ. η SOAP πάνω στην HTTP.

Οι μηχανισμοί υποστήριξης είναι εφοδιασμένοι με BPEL4WS, BPML/WSCI και DAML-S, αν και τα περισσότερα παραδείγματα που είναι διαθέσιμα εστιάζονται στη WSDL και SOAP υποστήριξη.

Αν λάβουμε υπόψη την ανάγκη για αποτελεσματική και ανεξάρτητη σχεδίου επίκληση, τα εισαχθέντα και εξαχθέντα δεδομένα, τα μηνύματα και η ακολουθία μηνυμάτων θα πρέπει να σχετίζονται με μια συγκεκριμένη τεχνολογία και να εκτίθενται στη δημόσια περιγραφή της υπηρεσίας. Με αυτή την έννοια η οντολογία μιας υπηρεσίας θα πρέπει να περιλαμβάνει έννοιες για να εκφράσει αυτή τη σχέση, όπως η έννοια της υποστήριξης που ορίζεται στην οντολογία της DAML-S.

Λόγω του σημασιολογικού συνδέσμου με την απαιτούμενη υποστήριξη, η DAML-S θα έπρεπε κανονικά να χρησιμοποιείται σαν σημείο έναρξης καθώς περιέχει μηχανισμό υποστήριξης. Ωστόσο, η οντολογία DAML-S σχετίζει την υποστήριξη απευθείας με μια δεδομένη υπηρεσία για αυτό το λόγο δεν υποστηρίζει την πολυμορφία. Διαφορετική υποστήριξη για κάθε σύνολο από εισαχθέντα δεδομένα θα πρέπει να οριστεί και η χρήση μιας σταθερής υποστήριξης, θα πρέπει να αποφασιστεί κατά τη τρέχουσα στιγμή. Επιπλέον το διαλλακτικό interface θα πρέπει να σχετίζεται με ένα παρόμοιο τρόπο με μια σταθερή τεχνολογία.

Ο βασικός μηχανισμός υποστήριξης της DAML-S μπορεί να επαναχρησιμοποιηθεί και να είναι χρήσιμος στην αυτόματη εύρεση.

2.1.8 Επανόρθωση και άλλες Απαιτήσεις

Επειδή λάθη εμφανίζονται συχνά στα συστήματα υπολογιστών, μια ευφυής περιγραφή υπηρεσίας, πρέπει να λάβει υπόψη της πιθανά λάθη και πως θα τα αντιμετωπίσει.

Για αυτό το λόγο τα λανθασμένα δεδομένα θα πρέπει να περιγράφονται παράλληλα με τα εισαχθέντα και εξαχθέντα δεδομένα. Μία SWS περιγραφή θα πρέπει να περιλαμβάνει μία ή περισσότερες θύρες λάθους, που θα παρέχουν τις λανθασμένες πληροφορίες στον αιτών, σε διαφορετικά σημεία ενδεχομένως της εκτέλεσης. Αυτές οι θύρες λάθους θα πρέπει να αναφέρονται σε μία κατάλληλη οντολογία με τον ίδιο τρόπο που αναφέρονται τα εισαχθέντα και εξαχθέντα δεδομένα. Οι θύρες λάθους μπορούν να θεωρηθούν ως ειδικοί τύποι εξαχθέντων δεδομένων, ώστε να εφαρμόζονται οι ίδιες απαιτήσεις σε αυτές, αν και δεν αναφέρονται σε καμία προ ή μετά υπόθεση. Οι θύρες λάθους, όπως τα εισαχθέντα και εξαχθέντα δεδομένα, θα χρησιμοποιηθούν με τον ίδιο τρόπο στον ορισμό του διαλλακτικού interface, ορίζοντας σε ποιο σημείο της εκτέλεσης απαιτείται ένα σταθερό εισαχθέν δεδομένο, τότε παραδίδονται τα εξαχθέντα δεδομένα στον αιτών καθώς και το που οι συγκεκριμένες θύρες λάθους μπορούν να αναφέρουν τις λανθασμένες πληροφορίες. Αυτές οι θύρες θα περιληφθούν στις πληροφορίες υποστήριξης της υπηρεσίας.

Στο περιεχόμενο των σημασιολογικών υπηρεσιών δικτύου, η δυναμική εύρεση και ο συνδυασμός των υπηρεσιών υπονοεί ότι δεν μπορεί να γίνει καμία εκ των προτέρων υπόθεση σε ότι αφορά τη διάρκεια της επίκλησης μιας υπηρεσίας. Για αυτό το λόγο η χρήση παραδοσιακών ACID συναλλαγών στην αντιμετώπιση των σφαλμάτων, δεν είναι χρήσιμη για αυτό το περιεχόμενο, καθώς απαιτούν blocking resources για μια απροσδιόριστη ποσότητα χρόνου. Για αυτό το λόγο εμφανίστηκε η έννοια της επανόρθωσης, για να υποκαταστήσει τις κλασσικές συναλλαγές. Επανόρθωση μιας υπηρεσίας σημαίνει ότι κάνουμε επίκληση σε μια ή περισσότερες υπηρεσίες για να αναιρεθούν οι ενέργειες της προηγούμενης. Παραδείγματος χάριν η υπηρεσία για την κράτηση μιας πτήσης μπορεί να διορθωθεί με την επίκληση μιας υπηρεσίας για την ακύρωση της πτήσης.

Η επανόρθωση βασίζεται στην υπόθεση της ύπαρξης μιας αντίστροφης υπηρεσίας. Αν δεν ισχύει αυτή η υπόθεση τότε τα αποτελέσματα μιας υπηρεσίας δεν μπορούν να διορθωθούν.

Εξαιτίας της ανάγκης για μια υπηρεσία που θα αντιστρέφει την ενέργεια της υπηρεσίας που έχουμε επικαλεστεί, η τοποθεσία μιας τέτοιας υπηρεσίας παίζει σημαντικό ρόλο στην επανόρθωση. Για τον εντοπισμό αυτής της υπηρεσίας, τρία πράγματα μπορεί να συμβούν: 1) η επικαλούμενη υπηρεσία δείχνει σε μια υπηρεσία που μπορεί να αναιρέσει την ενέργειά της 2) η επικαλούμενη υπηρεσία περιγράφει τη δυνατότητα της υπηρεσίας που χρειάζεται να επανορθώσει την επίκλησή της 3) η επικαλούμενη υπηρεσία δεν δίνει καμιά πληροφορία σχετικά με την υπηρεσία επανόρθωσης ή τη δυνατότητα που θα έπρεπε να παρουσιάζει. Στη τελευταία περίπτωση οι μηχανισμοί υποστήριξης πρέπει να αναπτυχθούν για να ανακαλύψουμε τις απαραίτητες δυνατότητες και τους απαραίτητους περιορισμούς για να επανορθώσουμε την υπηρεσία. Οι πρώτες δύο περιπτώσεις περιλαμβάνουν πληροφορίες μέσω της περιγραφόμενης δυνατότητας (2^η περίπτωση) ή τις δυνατότητες της υπηρεσίας που υποδεικνύεται (1^η περίπτωση) για κάθε περιορισμό επανόρθωσης π.χ η οικονομική ποινή όταν ακυρώνεται ένα αεροπορικό εισιτήριο.

Για αυτό το λόγο μια SWS περιγραφή πρέπει να επιτρέπει τη συμπερίληψη της πληροφορίας επανόρθωσης, αν και δεν μπορεί να είναι υποχρεωτική, καθώς μια δεδομένη υπηρεσία μπορεί να μην επιθυμεί να ορίσει πληροφορίες σχετικά με την επανόρθωση, ή μπορεί η επανόρθωση να μην είναι εφικτή. Διαφορετικοί βαθμοί πολυπλοκότητας, μπορούν να μοντελοποιηθούν.

Η DAML-S δεν περιλαμβάνει κανένα λάθος ή καμιά πληροφορία επανόρθωσης, έτσι η οντολογία πρέπει να επεκταθεί ξανά για να αντικατοπτρίσει το λάθος και τα δεδομένα επανόρθωσης. Η BPEL4WS και η BPML/WSCI μελετούν αυτή τη πληροφορία, για αυτό θα πρέπει να λαμβάνονται υπόψη καθώς αλλάζουν την οντολογία, για να συμπεριληφθούν τα δεδομένα που συζητήθηκαν.

Πληροφορίες όπως η ποιότητα μιας υπηρεσίας, η γεωγραφική περιοχή, ο χρόνος απόκρισης και άλλες μη λειτουργικές πληροφορίες πρέπει να αντικατοπτρίζονται στη περιγραφή μιας υπηρεσίας. Η λίστα των μη λειτουργικών ιδιοτήτων μιας υπηρεσίας πρέπει να είναι επεκτάσιμη. Η επεξεργασία ενός ολοκληρωμένου συνόλου από μη λειτουργικές ιδιότητες θα ολοκληρωθεί στο μέλλον.

[13]

2.2 Εισαγωγή Σημασιολογικών Στοιχείων Στις Υπηρεσίες Δικτύου

2.2.1 Εισαγωγή

Η ανάγκη για ευρέως αποδεκτά πρότυπα που θα καθιστούν ικανές τις υπηρεσίες δικτύου γίνεται επιτακτική. Η WSDL [14] είναι ήδη καθιερωμένη ως θεμέλιος λίθος στη διαδικασία ανάπτυξης της τεχνολογίας των υπηρεσιών δικτύου. Η WSDL στην ουσία προσδιορίζει τη σύνταξη των μηνυμάτων που εισέρχονται και εξέρχονται, μιας βασικής υπηρεσίας, καθώς και οτιδήποτε άλλο είναι απαραίτητο, για την επίκληση μιας υπηρεσίας. Ωστόσο δεν υποστηρίζει τη προδιαγραφή της ροής εργασιών που συνθέτουν οι βασικές υπηρεσίες. Σε αυτό το τομέα εξέχουσα θέση καταλαμβάνει η γλώσσα Business Process Execution Language (BPEL4WS) για υπηρεσίες δικτύου, που βρίσκεται υπό ανάπτυξη στην OASIS. Στο τομέα της διαφήμισης και εύρεσης υπηρεσιών, το UDDI [15] έχει λάβει μέχρι σήμερα τη περισσότερη προσοχή.

Συγχρόνως είναι έντονη και η ανάγκη του σημασιολογικού εμπλουτισμού των υπηρεσιών. Μια πλούσια γλώσσα αναπαράστασης, μας δίνει ένα πιο κατανοητό προσδιορισμό των διαφορετικών πλευρών μιας υπηρεσίας και αποτελεί μια καλή βάση σε ένα ευρύ φάσμα δραστηριοτήτων στη διάρκεια του κύκλου ζωής μιας υπηρεσίας. Για παράδειγμα αυτοματοποιείται η επίκληση και η επιλογή μιας υπηρεσίας, μεταφράζεται αυτόματα το περιεχόμενο μηνυμάτων ανάμεσα σε ετερογενείς υπηρεσίες. Περαιτέρω η πλούσια σημασιολογία βοηθά στην αυτοματοποίηση ενεργειών όπως η επαλήθευση στοιχείων, η προσομοίωση, η διαμόρφωση, στην παροχή αλυσιδωτής διαχείρισης, στη δημιουργία συμβολαίων, και στις διαπραγματεύσεις των υπηρεσιών.

Για να ανταποκριθούν σε αυτές τις ανάγκες οι ερευνητές έχουν αναπτύξει γλώσσες, αρχιτεκτονικές και διάφορες γενικά προσεγγίσεις. Η ιδέα του OWL-S είναι μια προσπάθεια δημιουργίας των θεμελίων για την ουσιαστικότερη εξέλιξη των υπηρεσιών δικτύου και των δυνατοτήτων τους, που μπορεί να υποστηριχτεί με τις υπάρχουσες τεχνολογίες. Ταυτόχρονα, αποτελεί στόχο η προώθηση της υιοθέτησης τεχνολογιών που μπορούμε να εκφράσουμε σημασιολογικά και είναι ήδη κατανοητές. Για αυτό έχουν γίνει προσπάθειες κατασκευής μηχανισμών, όπου θα μπορεί να χρησιμοποιείται το OWL-S παράλληλα με τα κύρια πρότυπα όπως το WSDL. Σκοπός

όσων θα αναφέρουμε παρακάτω είναι η δημιουργία ενός αρχικού χάρτη πορείας της ανάπτυξης των σημασιολογικών υπηρεσιών δικτύου, χρησιμοποιώντας το OWL-S παράλληλα με το WSDL και τα σχετικά πρότυπα. Τέλος θα αναφερθούν τα οφέλη που θα αποκομίσουμε αν εμπλουτιστούν οι υπηρεσίες με σημασιολογικά στοιχεία.

Θα δείξουμε πως μπορούμε να χρησιμοποιούμε το OWL-S παράλληλα με άλλα πρότυπα, εστιάζοντας στο WSDL, καθώς και την αξία που προστίθεται στις υπηρεσίες από τα σημασιολογικά στοιχεία. Για την καλύτερη επεξήγηση των παραπάνω θα χρησιμοποιήσουμε ένα απλό παράδειγμα.

2.2.2 Παράδειγμα

Η Amazon.com παρέχει μια ανοιχτή υπηρεσία η οποία επιτρέπει στα προγράμματα πελατών να ψάχνουν τις βάσεις δεδομένων του Amazon, να εντοπίζουν βιβλία και άλλα προϊόντα για τα οποία θα υπάρχει πρόσβαση μέσω της κύριας ιστοσελίδας του Amazon, χρησιμοποιώντας ένα φυλλομετρητή για την ολοκλήρωση των αγορών. Τα προγράμματα πελατών των υπηρεσιών δικτύου απαιτούν ένα ευρύ πεδίο με λέξεις κλειδιά που θα περιέχεται στη βάση δεδομένων του Amazon. Οι πελάτες μπορούν να ψάξουν για τα βιβλία ενός συγκεκριμένου συγγραφέα, για τα προϊόντα ενός συγκεκριμένου κατασκευαστή ή για ταινίες ενός συγκεκριμένου σκηνοθέτη. Είναι επίσης διαθέσιμα τα σχόλια των πελατών για τα εμπορεύματα.

Το Amazon παρέχει μία WSDL προδιαγραφή για την υπηρεσία δικτύου του, περιγράφοντας τις λειτουργίες που εκτελούνται καθώς και διδακτικά έγγραφα και κώδικες για πελάτες, που χρησιμοποιούνται σαν παραδείγματα. Τα διδακτικά έγγραφα και οι κωδικοί είναι απαραίτητοι ώστε να μπορούν οι προγραμματιστές να υλοποιήσουν σωστά την επιφάνεια εργασίας WSDL. Κανένα σύστημα λογισμικού (πράκτορας) δεν θα μπορούσε να διαβάσει και να υλοποιήσει επιφάνεια εργασίας WSDL, χωρίς την ανθρώπινη βοήθεια, λόγω του ότι η γλώσσα προδιαγραφής του WSDL δεν μπορεί να συμπεριλάβει τις αναπαραστάσεις των σημασιολογικών στοιχείων των λειτουργιών που έχουν οριστεί καθώς και των σχετικών στοιχείων που περιέχουν τα μηνύματα. Παραδείγματος χάριν όλα τα δεδομένα που εισάγονται και εξάγονται και αφορούν τις WSDL λειτουργίες του Amazon είναι τύπου string. Στόχος του σημασιολογικού δικτύου και του OWLS είναι να γεφυρώσει αυτό το χάσμα. Το

OWL-S είναι μια γλώσσα που μπορεί να ορίσει τη λειτουργία (τις προϋποθέσεις και τα αποτελέσματα) ενός χειρισμού και τους σημασιολογικούς τύπους για κάθε ένα από τα δεδομένα που εισάγονται και εξάγονται σε μια υπηρεσία. Το OWL-S βασίζεται στην υπόθεση ότι οι ορισμοί αυτών σημασιολογικών εννοιών είναι διαθέσιμοι σε URI παραπομπές στο σημασιολογικό δίκτυο, ώστε τα προγράμματα πελάτη και υπηρεσίας να μπορούν να μοιράζονται τους όρους και οι πελάτες να μπορούν να βρίσκουν τους ορισμούς όλων των παραπαιεμπτικών εννοιών, που αναπαρίστανται στην OWL-S.

Το αποτέλεσμα είναι ότι αν πάρουμε μια OWL-S περιγραφή των υπηρεσιών μαζί με μια WSDL περιγραφή, ένα πρόγραμμα πελάτη μπορεί να διαχωρίσει τη λειτουργία αίτησης για μια βιντεοκάμερα από εκείνη που ζητά το όνομα ενός συγγραφέα. Ο πελάτης μπορεί επίσης να μεταφράσει τα αποτελέσματα των ερωτημάτων χωρίς να κάνει τον προγραμματισμό σε αυτή τη συγκεκριμένη επιφάνεια εργασίας. Όταν χρησιμοποιούμε το OWL για τη σημασιολογική γραφή των στοιχείων επικοινωνίας, ο πελάτης της Amazon μπορεί να αναγνωρίσει αυτόματα ποια από τα δεδομένα που εισάγονται, απαιτούνται για το επιθυμητό είδος εύρεσης και να μετατρέψει αυτά τα στοιχεία, αν χρειάζεται, στη κατάλληλη μορφή και να μεταφράσει τα στοιχεία του μηνύματος που επιστρέφεται.

Η WSDL προδιαγραφή των εξερχόμενων δεδομένων για κάθε κλήση προς την υπηρεσία, στερείται ομοίως του σημασιολογικού ορισμού. Όλες οι λειτουργίες εύρεσης του Amazon επιστρέφουν αποτελέσματα χρησιμοποιώντας την ίδια δομή δεδομένων, που ονομάζεται *Details* ανεξάρτητα από το είδος των πληροφοριών που ζητούνται. Μπορούμε να συμπεράνουμε τα είδη των προϊόντων από τη δομή των δεδομένων, αναλύοντας τα στοιχεία που είναι συμπληρωμένα. Για παράδειγμα το *Details* περιέχει ένα πεδίο για Συγγραφείς που χρησιμοποιείται για την περιγραφή βιβλίων και ένα πεδίο που ονομάζεται Σκηνοθέτες και χρησιμοποιείται για τη περιγραφή ταινιών. Είναι πλέον στο χέρι του πελάτη να αναγνωρίσει ότι οι τιμές σε αυτά τα πεδία δείχνουν αν πρόκειται για βιβλίο ή για ταινία.

Η αδυναμία της WSDL να περιγράφει σημασιολογικά το νόημα των δεδομένων που εισάγονται και εξάγονται καθιστά αδύνατη την ανάπτυξη λογισμικών για πελάτες που θα μπορούν, χωρίς την ανθρώπινη βοήθεια, να βρίσκουν δυναμικά και να καλούν με επιτυχία, μία υπηρεσία. Οι προδιαγραφές της WSDL πρέπει να μεταφράζονται από τους προγραμματιστές, οι οποίοι μεταφράζουν τα ονόματα των λέξεων κλειδιά που υπάρχουν στα μηνύματα ώστε να ενώσουν ορισμένες υπηρεσίες

με τις εφαρμογές των πελατών τους. Στόχος του σημασιολογικού δικτύου είναι να υποστηρίξει τους πελάτες που μπορούν να βρουν και να υλοποιήσουν νέες υπηρεσίες χωρίς πρόσθετο προγραμματισμό. Αυτοί οι πελάτες θα μπορούν παραδείγματος χάριν, να βρίσκουν ιστοσελίδες που πουλούν βιβλία ή CD και να συγκρίνουν τις τιμές των συγκεκριμένων προϊόντων. Αυτές τις υπηρεσίες μπορούμε να τις βρούμε χρησιμοποιώντας σημασιολογικές περιγραφές που θα χαρακτηρίζουν τις υπηρεσίες, χωρίς την ανθρώπινη παρέμβαση. Επιπλέον τόσο η εύρεση όσο και η χρήση αυτών των υπηρεσιών είναι σταθερή, παρόλο που μπορεί να αλλάζει η σχεδίαση των υπηρεσιών στη διάρκεια του χρόνου, επειδή τα πρωτόκολλα των υπηρεσιών θα είχαν εκδοθεί και μεταφραστεί ξανά από το λογισμικό του πελάτη τη στιγμή της χρήσης.

2.2.3 Εισαγωγή στα Σημασιολογικά Στοιχεία

Το OWL-S είναι μια OWL οντολογία που περιλαμβάνει 3 υποκατηγορίες οντολογιών αλληλοσχετιζόμενες, που είναι γνωστές ως προφίλ (profile), μοντέλο διαδικασίας (process model), και (υποστήριξη) grounding. Εν συντομία το προφίλ χρησιμοποιείται για να εκφράσει τι κάνει μια υπηρεσία για σκοπούς διαφήμισης, δημιουργίας αιτήσεων για μια υπηρεσία και ταυτοποίησης. Το μοντέλο διαδικασίας περιγράφει το πώς λειτουργεί η επίκληση, η σύνθεση, η παρακολούθηση, η θέσπιση και η ανάκτηση. Η υποστήριξη αντιστοιχεί τις δομές του μοντέλου διαδικασίας με λεπτομερείς προδιαγραφές των δομών των μηνυμάτων και των πρωτοκόλλων.

Η WSDL θεωρεί τη προδιαγραφή των λειτουργιών ως το βασικό οικοδομικό λίθο των υπηρεσιών δικτύου. Οι λειτουργίες, προσφέρουν μια οργανωτική δομή γύρω από την οποία ορίζονται το συντακτικό και το πρότυπο των εισερχόμενων και εξερχόμενων μηνυμάτων. Η WSDL προσφέρει μια ανάλογη αλλά πιο θεωρητική δομή, γνωστή ως ατομική διαδικασία, η οποία αρχικά χαρακτηρίζεται σε όρους των δεδομένων της που εισάγονται, εξάγονται, των προϋποθέσεων και των αποτελεσμάτων.

Τα εισερχόμενα και εξερχόμενα δεδομένα μιας ατομικής διαδικασίας προκύπτουν από το σύστημα τύπων της OWL. Παραδείγματος χάριν το δείγμα του συνοδευτικού κώδικα μας δίνει μια απλουστευμένη δήλωση του OWLS για μια ατομική διαδικασία με τις δικές της προδιαγραφές εισόδου/εξόδου. Σε αυτή τη

περίπτωση η ατομική διαδικασία, τα δεδομένα που εισάγονται και εξάγονται και οι τύποι παραμέτρων ανήκουν στο namespace του μοντέλου διαδικασίας. Υποθέτουμε ότι τα Human, Book Title και ISBN είναι τάξεις οι οποίες ορίζονται στην κατάλληλη κυριότητα οντολογιών και έχουν διαφορετικά namespaces. Η υποστήριξη για αυτή την ατομική διαδικασία θα καθιέρωνε την επικοινωνία με μια συγκεκριμένη λειτουργία της WSDL και την επικοινωνία με κάθε στοιχείο εισόδου και εξόδου από ένα συγκεκριμένο μέρος ενός WSDL μηνύματος. Επίσης, αν είναι απαραίτητο, η υποστήριξη θα μπορούσε να ορίσει ένα κείμενο σε XSLT, για να μετατρέψει κάθε δεδομένο που εισάγεται και είναι εκφρασμένο σε όρους του OWLS, στη συντακτική μορφή που ορίζεται από την WSDL και αντίστροφα ισχύει το ίδιο για τα δεδομένα που εξάγονται. Πρόσθετες λεπτομέρειες και παραδείγματα για την υποστήριξη του OWL-S μπορούμε να βρούμε στην [16].

Ένα σημαντικό κομμάτι στη περιγραφή των υπηρεσιών δικτύου είναι ο ορισμός των προϋποθέσεων και των περιορισμών, συμπεριλαμβάνοντας τις προϋποθέσεις και τις επιδράσεις μιας διαδικασίας ή μιας υπηρεσίας. Οι προϋποθέσεις είναι μια λογική συνθήκη, που θα πρέπει να ικανοποιηθεί, από αυτόν που αιτείται την υπηρεσία πριν από την εκτέλεση της. Τα αποτελέσματα είναι μια λογική συνθήκη η οποία θα γίνει αληθής όταν εκτελεστεί με επιτυχία η υπηρεσία. Πολλές από τις πληροφορίες που παρέχουν οι υπηρεσίες δεν έχουν επιδράσεις. Γενικά πάντως, οι υπηρεσίες που αφορούν κυρίως συναλλαγές έχουν επίδραση, όπως παραδείγματος χάριν η χρέωση της πιστωτικής κάρτας ενός χρήστη ή η αποστολή προϊόντων κ.τ.λ. Η περιγραφή αυτών των επιδράσεων, είναι κρίσιμη για κάποιες πτυχές του αυτοματισμού των υπηρεσιών δικτύου.

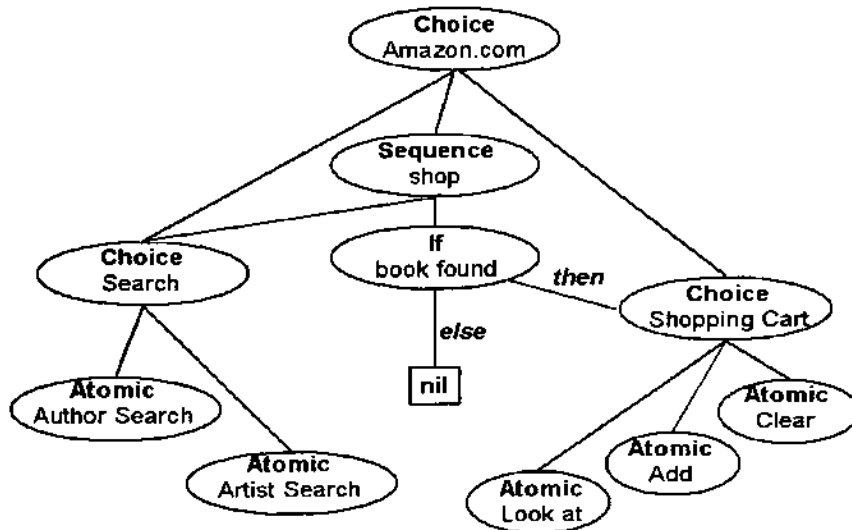
Για το προσδιορισμό των προϋποθέσεων και των επιδράσεων μιας διαδικασίας, το OWLS επιτρέπει τη χρήση μιας περισσότερο εκφραστικής γλώσσας από την OWL, όπως είναι η RuleML [17], η DRS [18] ή κάποιες άλλες από τις πρόσφατα προτεινόμενες OWL Rules γλώσσες [19]. Παραδείγματος χάριν κάποια από αυτές τις γλώσσες θα μπορούσε να χρησιμοποιηθεί για να εκφράσει τις προϋποθέσεις για μια υπηρεσία που πουλάει βιβλία, δηλώνοντας ότι κάποιος θα πρέπει να έχει έγκυρο λογαριασμό καθώς και έγκυρη πιστωτική κάρτα ώστε να κάνει μια αγορά.

2.2.4 Η Θέσπιση

Με τον όρο θέσπιση εννοούμε τη διαδικασία με την οποία ένας πελάτης ζητά μια επεξηγηματική περιγραφή μιας υπηρεσίας προκειμένου να ζητήσει κάτι από την υπηρεσία και να μεταφράσει την απάντηση. Εδώ η περιγραφή που μεταφράζεται είναι το μοντέλο διαδικασίας του OWLS που εκδίδεται από την υπηρεσία μαζί με τις προδιαγραφές της WSDL. Η θέσπιση ξεκινάει αιτιολογώντας προς τα πίσω, από τα δεδομένα που απαιτείται να εισαχθούν από την επιλεγμένη υπηρεσία για να βρει τις πληροφορίες που είναι διαθέσιμες στο πελάτη που θέλει να επικαλεστεί με επιτυχία μια υπηρεσία. Οι τιμές των εισαχθέντων δεδομένων αντιστοιχίζονται μέσω της υποστήριξης της υπηρεσίας με τα στοιχεία ανταπόκρισης στο μοτίβο ενός WSDL μηνύματος, το οποίο καταλήγει τελικά σε ένα μήνυμα επικοινωνίας με την υπηρεσία. Το μήνυμα που εξάγεται το διαχειριζόμαστε με ουσιαστική αντιστροφή της υπηρεσίας. Το μήνυμα που εξάγεται από την WSDL και λαμβάνεται από το πελάτη μετασχηματίζεται ξανά μέσω της υποστήριξης σε μία OWLS αναπαράσταση του περιεχομένου του μηνύματος το οποίο μπορεί να μεταφραστεί από το μηχανισμό υποστήριξης του πελάτη.

Για την υλοποίηση αυτής της διαδικασίας με την υπηρεσία δικτύου Amazon, απαιτείται πρώτα μια OWLS περιγραφή αυτής της υπηρεσίας, η οποία αναπαριστά πιο ολοκληρωμένα τα δεδομένα που εισάγονται και εξάγονται σε μια υπηρεσία. Κατά ένα μέρος μπορούμε να αυτοματοποιήσουμε τη δημιουργία αυτής της περιγραφής, παράγοντας μια αρχική OWL-S περιγραφή της υπηρεσίας δικτύου Amazon χρησιμοποιώντας εργαλεία που μετατρέπουν την WSDL σε μια τμηματική προδιαγραφή του OWL-S [20]. Εφόσον η περιγραφή της WSDL δεν περιλαμβάνει επαρκείς πληροφορίες για τη διαμόρφωση ενός ολοκληρωμένου μοντέλου διαδικασίας προσθέτουμε τη περιγραφή που δημιουργήθηκε. Η διαδικασία αυτή χωρίζεται σε δύο μέρη:

- 1) προσθέτουμε σημασιολογικές περιγραφές από κάθε παράμετρο που εισάγεται, στο μοντέλο διαδικασίας που προκύπτει καθώς και παρέχοντας κάθε απαραίτητη μετατροπή δεδομένων
- 2) κατασκευάζουμε ένα σύνθετο μοντέλο διαδικασίας που θα συνδέει τις διάφορες λειτουργίες που παρέχονται από την υπηρεσία δικτύου με πρότυπα μηνυμάτων που έχουν σημασιολογικό περιεχόμενο.



Σχήμα 5 - Απλοποιημένο Μοντέλο Διαδικασίας για την Υπηρεσία Δικτύου Amazon

Το μοντέλο διαδικασίας που έχει προκύψει δίνεται στο Σχήμα 5, το οποίο δείχνει τη σχέση ανάμεσα στις διάφορες λειτουργίες της υπηρεσίας που αναπαρίστανται στο τελικό μοντέλο διαδικασίας του OWL-S. Ο πελάτης μπορεί να εκτελέσει τριών ειδών εργασίας: να ψάξει τις βάσεις δεδομένων του Amazon χρησιμοποιώντας την εύρεση για συγγραφέα ή για κάποιο καλλιτέχνη ή και άλλους τύπους εύρεσης, να προβάλλει ή να μετατρέψει το shopping cart προσθέτοντας νέα προϊόντα, εκκαθαρίζοντας το ή απλά προβάλλοντας τα περιεχόμενα του ή τέλος μπορεί να εκτελέσει μια σύνθετη διαδικασία αγοράς που συνδυάζει τα δύο παραπάνω είδη, δηλαδή αρχικά ψάχνει και έπειτα προσθέτει το προϊόν που βρήκε στο shopping cart. Η WSDL περιγραφή της Amazon περιγράφει μόνο τις λειτουργίες που απεικονίζονται κυκλικά στο γράφημα.

Κάθε μία από τις περιγραφές διαδικασίας του OWL-S ορίζει τους σημασιολογικούς τύπους των δεδομένων που απαιτούνται σαν δεδομένα που εισάγονται και επιστρέφονται σαν εξαχθέντα. Παραδείγματος χάριν το δεδομένο που εισάγεται για την εύρεση του συγγραφέα θα πρέπει να είναι ένα στιγμιότυπο της τάξης Human που έχει συγκεκριμένη σχέση με το βιβλίο που αναζητείται. Η χρήση

OWL τάξεων και ιδιοτήτων ως περιορισμοί πάνω σε στιγμιότυπα που πρέπει να αναγνωρίζονται ως μεταβλητές εισόδου, είναι σημαντική για τις διαδικασίες συμπερασματολογίας. Η χρήση OWL τάξεων και ιδιοτήτων ως περιορισμοί, επιτρέπει τη διαμόρφωση της επίκλησης για μια υπηρεσία, χωρίς να χρειάζεται να γράψει κάποιος προγραμματιστής ειδικό κώδικα για κάθε πιθανό τύπο επίκλησης.

Αν ο πελάτης έχει στόχο να βρει τη τιμή για ένα συγκεκριμένο βιβλίο, μπορεί να κάνει σύνθετη αναζήτηση βάζοντας σαν κριτήρια, στοιχεία σχετικά με αυτό που ψάχνει. Εφόσον η υπηρεσία που θα επικαλεστούμε επιλέγεται επειδή το είδος της πληροφορίας που ψάχνουμε περιγράφεται σαν τμήμα των εξαχθέντων δεδομένων της υπηρεσίας και περιγράφει αυτή τη πληροφορία όπως αυτή σχετίζεται με τα στοιχεία των βάσεων δεδομένων για τα βιβλία, ο πελάτης μπορεί να κρίνει από την περιγραφή που εξάγεται (το ISBN ενός βιβλίου) και από τα απαραίτητα στοιχεία εισαγωγής (ο συγγραφέας, ο τίτλος του βιβλίου που έχει το ISBN που ψάχναμε). Επομένως ο πελάτης γνωρίζει ποια δεδομένα θα επιστραφούν, χωρίς να χρειάζεται να μαντέψει από το στιγμιότυπο της δομής δεδομένων Details.

Η υποστήριξη του OWL-S αναλαμβάνει τη διαδρομή από τις έννοιες που περιγράφουν τα δεδομένα των διαδικασιών που εισάγονται και εξάγονται έως τα δεδομένα που εισάγονται και εξάγονται από τις λειτουργίες αλληλογραφίας της προδιαγραφής WSDL της Amazon. Αυτό έχει σαν αποτέλεσμα, η πραγματική επίκληση να είναι συνεπής με τις απαιτήσεις της Amazon. Μπορούμε να αλληλεπιδράσουμε επιτυχώς με την ιστοσελίδα του Amazon χρησιμοποιώντας τη DAML-S Virtual Machine [20].

2.2.5 Η Εύρεση

Εύρεση είναι η διαδικασία αναζήτησης υπηρεσιών δικτύου με δεδομένες δυνατότητες. Γενικά η εύρεση προϋποθέτει ότι οι υπηρεσίες δικτύου έχουν γνωστοποιήσει τις δυνατότητες τους μέσω καταχωρήσεων, καθώς και ότι οι υπηρεσίες που ψάχνουν ελέγχουν τις καταχωρήσεις για υπηρεσίες με συγκεκριμένες δυνατότητες. Ο ρόλος των καταχωρήσεων είναι να αποθηκεύει τις γνωστοποιημένες δυνατότητες των υπηρεσιών αλλά και να εκτελεί τη διαδικασία ταυτοποίησης ανάμεσα στο αίτημα και τις δυνατότητες των υπηρεσιών.

Η διαδικασία εύρεσης απαιτεί μια γλώσσα, που θα μπορεί να κωδικοποιήσει τις δυνατότητες των υπηρεσιών δικτύου για τις γνωστοποιήσεις και για τις αιτήσεις. Επιπλέον η εύρεση απαιτεί μια διαδικασία ταυτοποίησης που θα συγκρίνει τις γνωστοποιήσεις με τις αιτήσεις για να εξακριβώσει αν ταιριάζουν.

Παρακάτω θα περιγράψουμε πως το OWL-S μπορεί να χρησιμοποιηθεί για να εκφράσει και να ταυτοποιήσει δυνατότητες. Τέλος θα δείξουμε πως μπορεί να χρησιμοποιηθεί για να προσθέσει τη δυνατότητα ταυτοποίησης στο UDDI, στο διαδεδομένο δηλαδή σύστημα εύρεσης καταχωρήσεων για υπηρεσίες δικτύου.

2.2.6 Αναπαράσταση Δυνατοτήτων

Οι δυνατότητες των υπηρεσιών δικτύου ανταποκρίνονται στις λειτουργίες που παρέχονται από τις υπηρεσίες δικτύου. Μιλώντας ευρέως υπάρχουν δύο τρόποι για να αναπαραστήσουμε τις λειτουργίες. Η πρώτη προσέγγιση μας δίνει μια εκτεταμένη οντολογία από λειτουργίες όπου η κάθε τάξη στην οντολογία ανταποκρίνεται σε μια τάξη από ομογενείς λειτουργίες. Ένα απλό παράδειγμα οντολογίας που ορίζει μια ταξινόμια από ηλεκτρονικές υπηρεσίες φαίνεται παρακάτω. Η χρήση μιας τέτοιας οντολογίας χαρακτηρίζει υπηρεσίες όπως η Amazon σαν στιγμιότυπα τάξεων που αναπαριστούν τις δυνατότητες τους. Η Amazon για παράδειγμα μπορεί να διαφημιστεί σαν υπηρεσία πώλησης βιβλίων.

Ο δεύτερος τρόπος για να αναπαραστήσουμε τις δυνατότητες είναι να παρέχουμε μια γενική περιγραφή της λειτουργίας σε όρους της μετατρέψιμης κατάστασης που παράγει. Ο τελευταίος τυπικά χρησιμοποιείται από γλώσσες σχεδιασμού της τεχνητής νοημοσύνης, όπως η PDDL [21]. Παραδείγματος χάριν η Amazon μπορεί να ορίσει ότι παρέχει μια υπηρεσία η οποία θα ζητά το τίτλο βιβλίου, τον συγγραφέα, τη διεύθυνση και έναν έγκυρο αριθμό πιστωτικής κάρτας. Έτσι δημιουργείται η φάση μεταβίβασης όπου το βιβλίο παραδίδεται στην αντίστοιχη διεύθυνση, η πιστωτική κάρτα χρεώνεται και τελικά το βιβλίο θα αλλάξει ιδιοκτησία. Παρά τις διαφορές και οι δύο τρόποι αναπαριστούν τις δυνατότητες χρησιμοποιώντας οντολογίες για να δημιουργήσουν τη σύνδεση ανάμεσα σε αυτά που η υπηρεσία δικτύου κάνει και στη γενική περιγραφή του περιβάλλοντος μέσα στο οποίο η υπηρεσία δικτύου λειτουργεί.

Υπάρχουν ανταλλαγές ανάμεσα στις δύο αναπαραστάσεις των λειτουργιών που βοηθούν να επιλεγεί μία αναπαράσταση, μέσω της ανάλυσης των εργασιών που πρέπει να γίνουν. Η χρήση μιας σαφής οντολογίας των δυνατοτήτων διευκολύνει τη διαδικασία εύρεσης εφόσον η διαδικασία ταυτοποίησης μειώνεται σε δευτερεύουσα υπόθεση ανάμεσα στις δυνατότητες της οντολογίας. Από την άλλη πλευρά η απαρίθμηση όλων των πιθανών δυνατοτήτων ακόμα και στις περιορισμένες κυριότητες για κωδικοποίηση οντολογιών, είναι αρκετά δύσκολη. Παραδείγματος χάριν, μπορούμε να λάβουμε υπόψη το πρόβλημα της αναπαράστασης υπηρεσιών μετάφρασης από μια γλώσσα πηγή π.χ L_S στη γλώσσα στόχο L_T . Αν υποθέσουμε ότι υπάρχουν n πιθανές γλώσσες, υπάρχουν n^2 πιθανοί τύποι υπηρεσιών μετάφρασης. Η ταξινομία της υπηρεσίας μπορεί να έχει διαφορετικές τάξεις υπηρεσιών για κάθε ζευγάρι γλωσσών που θα μπορούσε να μεταφραστεί, ή θα μπορούσε απλά να αναπαριστά υπηρεσίες μετάφρασης σαν μια γενική κατηγορία, με ιδιότητες που επιτρέπουν σε συγκεκριμένες υπηρεσίες να περιγράφουν τις γλώσσες που μπορούν να μεταφράσουν ή να μεταφράσουν σε αυτές. Αυτή η προσέγγιση είναι συνεπής με τη περιγραφή των δυνατοτήτων σε όρους της μετατρέψιμης κατάστασης. Διαχωρίζει τους μεταφραστές μέσω της περιγραφής της παραγωγής των διαφορετικών ειδών αποτελεσμάτων.

Πρέπει να σημειώσουμε ότι οι τύποι των δεδομένων που εισάγονται και εξάγονται, μιας τέτοιας υπηρεσίας δεν επαρκούν ώστε να μπορούμε να διαχωρίσουμε τις δυνατότητες. Ας λάβουμε υπόψη μας παραδείγματος χάριν, μια υπηρεσία η οποία δέχεται σαν εισαχθέν δεδομένο γεωγραφικές περιοχές και σαν αποτέλεσμα αναφέρει τα ονόματα διαφορετικών κρασιών. Αυτό το ζευγάρι δεδομένων (που εισάγονται και εξάγονται) μπορεί να χρησιμοποιηθεί από δύο πολύ διαφορετικές υπηρεσίες: μία που θα αναφέρει ποια κρασιά παράγονται σε μια συγκεκριμένη περιοχή και μια άλλη που θα αναφέρει ποια κρασιά πωλούνται σε μια συγκεκριμένη γεωγραφική περιοχή.

Το OWL-S μπορεί να υποστηρίξει και τους δυο τρόπους αναπαράστασης των δυνατοτήτων των υπηρεσιών. Η υπομονάδα της OWL-S Service Profile παρέχει μια υψηλού επιπέδου περιγραφή των υπηρεσιών, σαν μετατροπή από μια κατάσταση σε μια άλλη. Σε αυτό το επίπεδο το Service Profile μας δίνει μια άποψη της υπηρεσίας δικτύου ως μια διαδικασία η οποία απαιτεί εισαχθέντα δεδομένα, καθώς και κάποιες έγκυρες προϋποθέσεις και η οποία μας δίνει αληθή αποτελέσματα. Πιο συγκεκριμένα το Service Profile παρέχει δύο τύπους πληροφοριών, ο πρώτος είναι μια λειτουργική

περιγραφή της υπηρεσίας δικτύου σε όρους της μετατροπής που η ίδια η υπηρεσία δικτύου παράγει. Ο δεύτερος είναι ένα σύνολο από μη λειτουργικές ιδιότητες που ορίζουν τους περιορισμούς για την υπηρεσία που παρέχεται. Η λειτουργική περιγραφή περιγράφει τόσο τη μετατροπή των πληροφοριών που προκύπτει από την παραγωγή των δεδομένων που εξάγονται, όσο και τη μετατρεπόμενη κατάσταση που προκύπτει από την δημιουργία συνεπειών ξεκινώντας από μια κατάσταση όπου οι προϋποθέσεις ικανοποιούνται. Οι μη λειτουργικές ιδιότητες ορίζουν τη ποιότητα της υπηρεσίας που παρέχεται από την υπηρεσία δικτύου ή τις απαιτήσεις ασφάλειας.

Εφόσον η OWL-S συνθέτει την εκτεταμένη αλλά και λειτουργική άποψη των υπηρεσιών δικτύου, παρέχει μια ολοκληρωμένη περιγραφή των υπηρεσιών που περιγράφει. Μπορεί να εκμεταλλευτεί τις οντολογίες των υπηρεσιών και των προϊόντων όπου και αν υπάρχουν, μέχρι το σημείο όπου θα μπορούν να αναπαραστήσουν τις δυνατότητες μιας υπηρεσίας δικτύου. Επιπλέον μπορεί να χρησιμοποιήσει τη μετατροπή που παράγεται από την υπηρεσία δικτύου για να μπορεί να παρέχει μια καλύτερη περιγραφή της υπηρεσίας δικτύου ή για να είναι σε θέση να περιγράψει τις επιδράσεις της χρήσης μιας υπηρεσίας δικτύου ακόμα και όταν η δυνατότητα της δεν συμφωνεί με καμία λειτουργική περιγραφή.

2.2.7 Ταυτοποίηση Ικανοτήτων

Η ταυτοποίηση δυνατοτήτων συγκρίνει τις δυνατότητες που παρέχονται από κάθε υπηρεσία, που έχει γνωστοποιήσει τις δυνατότητες της, με τις δυνατότητες που χρειάζεται ο αιτών. Ο στόχος είναι να βρει την διαφήμιση που θα δίνει τα αποτελέσματα που χρειάζεται ο αιτών. Γενικά, δεν θα ήταν ρεαλιστικό να περιμένουμε ότι η δυνατότητα που προσφέρεται θα ταιριάζει απόλυτα με το ερώτημα. Π.χ η απαιτούμενη υπηρεσία μπορεί να αφορά πληροφορίες για μετοχές και η εργασία του μηχανισμού ταυτοποίησης είναι να αποφασίσει αν μπορεί να εκπληρωθεί από μια υπηρεσία που παρέχει οικονομικά νέα. Ο μηχανισμός ταυτοποίησης θα πρέπει να καθορίσει πόσο πιθανό είναι κάθε διαφήμιση ικανότητας, να υποδεικνύει ότι η υπηρεσία θα ολοκληρώσει τη συγκεκριμένη λειτουργία που καθορίζεται σε αυτήν την αίτηση.

Ένας αριθμός από αλγόριθμους ταυτοποίησης ικανοτήτων έχουν προταθεί για το OWL-S. Χρησιμοποιούν τις περιγραφές υπηρεσιών στα προφίλ υπηρεσιών και

στις οντολογίες που είναι διαθέσιμες, για να αποφασίσουν αν υπάρχει μια σχέση ανάμεσα στις αιτήσεις για υπηρεσίες και στις διαφημίσεις των παρεχόμενων υπηρεσιών. Γενικά, αξιοποιούν μια από τις δύο απόψεις των δυνατοτήτων που προαναφέραμε.

Οι ταυτόσημοι αλγόριθμοι, έχοντας σαν δεδομένο την διαθεσιμότητα των λειτουργιών των οντολογιών, μπορούν να εκφράσουν δυνατότητες. Η ταυτοποίηση ανάμεσα στην αίτηση και τη διαθέσιμη καταχώρηση περιορίζεται στην δευτερεύουσα ανάμεσά τους σχέση. Διαφορετικοί βαθμοί ταυτοποίησης εντοπίζονται ανάλογα με το αν η καταχώρηση και η αίτηση περιγράφουν την ίδια ικανότητα ή όταν η μία το υποθέτει για την άλλη.

Άλλοι αλγόριθμοι ταυτοποίησης, υποθέτουν ότι οι ικανότητες ταυτοποίησης περιγράφονται από το επίπεδο μετασχηματισμού που παράγεται από την υπηρεσία Ιστού. Αυτοί οι μηχανισμοί ταυτοποίησης συγκρίνουν το επίπεδο μετασχηματισμού που περιγράφεται σε κάθε καταχώρηση με αυτό που περιγράφεται στην αίτηση. Κάνουν δύο ταυτοποιήσεις, μία που συγκρίνει τα δεδομένα που εξάγονται και μία που συγκρίνει τα δεδομένα που εισάγονται. Αν το αποτέλεσμα που απαιτείται από τους αιτούντες καλύπτεται από τη καταχώρηση, ελέγχονται τα εισαχθέντα δεδομένα. Αν τα δεδομένα που καθορίζονται στην αίτηση εντάσσονται στους αποδεκτούς τύπους εισαχθέντων τη υπηρεσίας, τότε μπορούμε να θεωρήσουμε την υπηρεσία υποψήφια για την ολοκλήρωση των απαιτήσεων του αιτούντα.

Στην πραγματικότητα υπάρχει μια ασυμμετρία ανάμεσα στη ταυτοποίηση των δεδομένων και των αποτελεσμάτων μιας υπηρεσίας δικτύου. Τελικά ο αιτών χρειάζεται μια υπηρεσία δικτύου που παράγει τα επιθυμητά αποτελέσματα. Όταν βρεθεί η υπηρεσία δικτύου που παρέχει τα απαιτούμενα αποτελέσματα, ο αιτών μπορεί είτε να αποπειραθεί να λάβει από εκείνη όλα τα εισαχθέντα δεδομένα, ή να χρησιμοποιήσει στις δικές του ικανότητες σύνθεσης για να βρει άλλες υπηρεσίες δικτύου που θα μπορούν να παρέχουν τα απαιτούμενα δεδομένα.

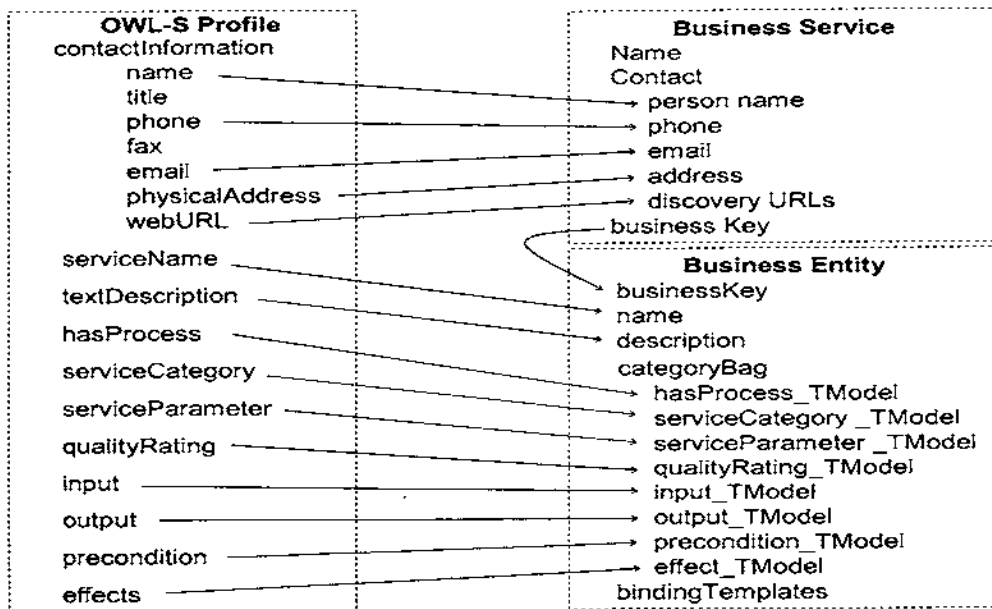
2.2.8 Η Σχέση με τη UDDI

Η UDDI (Universal Description Discovery and Integration) είναι μια βιομηχανική πρωτοβουλία της οποίας σκοπός είναι να δημιουργήσει ένα ευρύ διαδικτυακό δίκτυο εγγραφών στις υπηρεσίες Ιστού. Η UDDI επιτρέπει στις επιχειρήσεις να εγγραφούν στον ιστό καθορίζοντας τους τρόπους επαφής σε όρους

των θυρών που χρησιμοποιούνται από την υπηρεσία, για την επεξεργασία των αιτημάτων και σε όρους φυσικών επαφών με ανθρώπους που μπορούν να απαντήσουν σε ερωτήσεις σχετικές με την υπηρεσία. Επιπροσθέτως, η UDDI παρέχει μια γλώσσα, για να καθορίζει ένα απεριόριστο σύνολο χαρακτηριστικών των υπηρεσιών που μπορούν να βοηθήσουν στην διαδικασία του εντοπισμού, της επιλογής και της επίκλησης μιας υπηρεσίας.

Η UDDI υποστηρίζεται από εξέχουσες επιχειρήσεις κατασκευής λογισμικών, που έχουν επενδύσει σημαντικά στις υπηρεσίες Ιστού. Εξαιτίας, αυτής της υποστήριξης, η UDDI γίνεται η δεδομένη «αποθήκη» για τις υπηρεσίες Ιστού. Παρότι έχει αυτό το ρόλο, η UDDI παρέχει ένα πολύ ισχύο μηχανισμό εύρεσης, που δεν επιτρέπει την εύρεση υπηρεσιών Ιστού, παρά μόνο στη βάση του είδους των προβλημάτων που παρέχει. Το κύριο πρόβλημα με την UDDI είναι ότι δεν παρέχει μια γλώσσα αναπαράστασης δυνατοτήτων όπως το Προφίλ Υπηρεσίας του OWL-S. Ως συνέπεια, η UDDI δεν παρέχει την λειτουργία της εύρεσης βάσει ικανοτήτων. Το αποτέλεσμα είναι ότι η UDDI υποστηρίζει την τοποθεσία των πληροφοριών σχετικά με τις υπηρεσίες Ιστού, όταν είναι γνωστό ποια υπηρεσία δικτύου θα χρησιμοποιηθεί, αλλά είναι αδύνατο να εντοπίσουμε μια υπηρεσία Ιστού έχοντας σαν βάση μόνο τα προβλήματα που μπορεί να επιλύσει.

Το OWL-S και η UDDI συμπληρώνουν η μια την άλλη. Η UDDI παρέχει μια παγκοσμίου εύρους κατανεμημένη εγγραφή και αποτελεί ουσιαστικά βιομηχανικό πρότυπο. Από την άλλη πλευρά το OWL-S παρέχει τις πληροφορίες που απαιτούνται για την ικανότητα ταυτοποίησης. Ο μηχανισμός ταυτοποίησης της OWL-S/UDDI ενσωματώνει τη δυνατότητα ταυτοποίησης της OWL-S στη καταχώρηση της UDDI. Αυτή η ένωση βασίζεται στην αντιστοίχιση του προφίλ υπηρεσιών του OWL-S με την αναπαράσταση UDDI υπηρεσιών δικτύου, όπως παρουσιάζεται στο Σχήμα 6. Η λειτουργία αντιστοίχισης καθορίζει ένα σύνολο ειδικών UDDI Tmodels που αποθηκεύουν τις πληροφορίες του OWL-S, αλλά δεν μπορούν να τα αναπαραστήσουν στη πρότυπη αναπαράσταση υπηρεσιών Δικτύου της UDDI.



Σχήμα 6 - Η αντιστοίχηση του OWL-S με τη UDDI

Η ένωση OWL-S/UDDI παρέχει όλες τις λειτουργικότητες που παρέχονται από την UDDI χρησιμοποιώντας ακριβώς το ίδιο API, έτσι ώστε κάθε UDDI να μπορεί να αλληλεπιδράσει με αυτό για να ανακτήσει πληροφορίες για τις διαθέσιμες υπηρεσίες ιστού. Επίσης η OWL-S/UDDI υποστηρίζει τη δυνατότητα ταυτοποίησης, εκμεταλλεζόμενη τη δυνατότητα αναπαράστασης και την λειτουργία ταυτοποίησης του OWL-S. Το αποτέλεσμα είναι μία UDDI, στην οποία μπορούμε να ψάξουμε και να βρούμε υπηρεσίες δικτύου με βάση τις ικανότητές τους.

2.2.9 Η Σύνθεση

Η σύνθεση είναι η επεξεργασία επιλογής, ένωσης και εκτέλεσης υπηρεσιών ιστού προκειμένου να επιτευχθεί ο σκοπός του χρήστη. «Ανέλαβε τις ταξιδιωτικές ετοιμασίες για το ταξίδι μου στο WWW 2004 CONFERENCE» ή «αγόρασε μου ένα Apple iPod στην καλύτερη διαθέσιμη τιμή» είναι παραδείγματα πιθανών απαιτήσεων από το χρήστη που καθορίζονται από τη σύνθεση.

Οι άνθρωποι κάνουν χειροκίνητα τη σύνθεση WS εκμεταλλεζόμενοι τις δικές τους γνώσεις για το τι κάνει μια υπηρεσία ιστού (π.χ. η www.apple.com θα χρεώσει τη πιστωτική σας κάρτα και θα σας στείλει ένα iPod), καθώς και τις πληροφορίες που

παρέχονται στην ιστοσελίδα της υπηρεσίας δικτύου, με σκοπό να εκτελέσουν μια συλλογή υπηρεσιών και κατά συνέπεια να πετύχουν το στόχο τους. Για να αυτοματοποιήσουμε τη σύνθεση WS, όλες αυτές οι πληροφορίες θα πρέπει να κωδικοποιούνται σε μία ξεκάθαρη και ερμηνεύσιμη από τους Η/Υ, μορφή. Κανένα από τα υπάρχοντα βιομηχανικά δεδομένα για την περιγραφή WS δεν κωδικοποιεί σε αυτό το επίπεδο λεπτομέρειας. Επίσης, οι περιγραφές που παρέχουν δεν είναι ερμηνεύσιμες από τους Η/Υ και αυτό έχει ως συνέπεια να μην παρέχει έναν αξιόπιστο χειρισμό από ένα αυτοματοποιημένο σύστημα υποστήριξης, δηλαδή την ανάγκη για OWL-S.

Η αυτοματοποιημένη σύνθεση WS είναι παρόμοια με ένα πρόβλημα σχεδίασης της τεχνητής νοημοσύνης και ένα λογισμικό πρόβλημα σύνθεσης, και αγγίζει και τα δύο πεδία έρευνας. Με σκοπό να επιτύχει την αυτοματοποιημένη σύνθεση WS, ένα σύστημα υποστήριξης πρέπει να διατάσσει να ενώνει και να εκτελεί τις υπηρεσίες ιστού που συνολικά επιτυγχάνουν του στόχο του χρήστη. Αυτό εμπεριέχει την επίλυση περιορισμών ανάμεσα στα δεδομένα, τα αποτελέσματα, τις προϋποθέσεις και τις επιδράσεις (IOPEs) των υπηρεσιών ιστού και τ' αποτελέσματα και τις επιρροές (OEs) που ο χρήστης επιθυμεί. Παραδείγματος χάριν αν κάποιος ξεκινήσει με το σκοπό ενός ατζέντη και τον ταιριάζει μαζί με τα αποτελέσματα και τις επιδράσεις μιας υπηρεσίας ιστού, το αποτέλεσμα είναι ένα στιγμιότυπο της διαδικασίας, συν τις περιγραφές των νέων στόχων που θα πρέπει να ικανοποιηθούν βασιζόμενοι στα δεδομένα και στις προϋποθέσεις αυτής της επεξεργασίας. Οι νέοι στόχοι (δεδομένα και προϋποθέσεις), φυσικά θα ταιριάζουν με άλλες διαδικασίες (αποτελέσματα και επιδράσεις), και έτσι λογικά προκύπτει η σύνθεση. Οι περιορισμοί ανάμεσα στα δεδομένα, τα αποτελέσματα, τις προϋποθέσεις και τις επιδράσεις, υπαγορεύουν την σύνθεση των υπηρεσιών. Δύο τύποι προβλημάτων σύνθεσης παρουσιάζονται:

i) αυτοί που περιλαμβάνουν μόνο τις πληροφορίες που παρέχονται από τις υπηρεσίες

ii) αυτοί που περιλαμβάνουν και τις πληροφορίες που παρέχονται και τις εναλλακτικές υπηρεσίες.

Η πρώτη απαιτεί μια πλούσια σημασιολογική παρουσίαση δεδομένων και αποτελεσμάτων (IO). Η δεύτερη απαιτεί μια παρουσίαση των IOPE. Υπενθυμίζουμε ότι οι επιρροές (E) είναι οι επιδράσεις του προγράμματος (π.χ. η www.apple.com θα χρεώσει τη πιστωτική σας κάρτα και θα σας στείλει ένα iPod). Οι προϋποθέσεις WS

και οι επιδράσεις δεν είναι κωδικοποιημένες σε κανένα υπάρχον βιομηχανικό πρότυπο. Κωδικοποιούνται σε μια σαφώς ερμηνεύσιμη μορφή για τον Η/Υ στο OWL-S. Από τη στιγμή που συμπληρώνονται οι πληροφορίες που περιέχονται στην WSDL, δεν υπάρχει καμία υποστήριξη γι' αυτά τα χαρακτηριστικά στο επίπεδο WSDL.

Επίσης για την ταυτοποίηση των IOPE, το πρόβλημα της αυτοματοποιημένης WS σύνθεσης μπορεί να περιλαμβάνει την επιλογή από εναλλακτικές υπηρεσίες δικτύου, οι οποίες ταιριάζουν στους IOPE περιορισμούς του προβλήματος της σύνθεσης. Π.χ υπάρχουν πολλές υπηρεσίες ιστού μέσω των οποίων ο χρήστης μπορεί να αγοράσει ένα iPod. Για να μπορεί να επιλέξει ανάμεσα σε εναλλακτικές υπηρεσίες, ο μηχανισμός σύνθεσης απαιτεί κάποια μορφή επιλογής υπηρεσίας.

Μπορούμε να πούμε ότι είναι παρόμοιο με το πρόβλημα της εύρεσης που περιγράψαμε νωρίτερα, και όπως αναφέραμε απαιτεί αναπαράσταση των ιδιοτήτων, των ικανοτήτων και των λειτουργιών μιας υπηρεσίας ιστού.

Υπάρχουν αρκετές διαφορετικές προσεγγίσεις για την WS σύνθεση. Όλοι χαρακτηρίζουν τις επεξεργασίες OWL-S ως ενέργειες με δεδομένα, αποτελέσματα, προϋποθέσεις και επιδράσεις και χρησιμοποιούν τεχνολογία σχεδιασμού για να επιτύχουν τη σύνθεση WS. Π.χ η εργασία που έχει γίνει για τα μοντέλα [22] διεκπαιρώνεται με την ίδια τυποποίηση όπως ο χειρισμός STRIPS [23] και σχεδιάζει από μια συνέχεια των υπηρεσιών ιστού, για να επιτύχει το σκοπό του χρήστη. Αρχικά το σύστημα μπορεί να βάλει σε σειρά ένα σύνολο ενεργειών για να φθάσει σε ένα νέο σχέδιο αντιμετώπισης μιας υπηρεσία ιστού. Το σύστημα όπως περιγράφεται βρίσκεται σ' ένα πρώιμο στάδιο ανάπτυξης, και δεν μπορεί να ανταποκριθεί σε τέτοιου είδους βασικά προβλήματα όπως το να ανταπεξέλθει σε απρόβλεπτα αποτελέσματα ενεργειών. Η [24] ερευνά τη χρήση σχεδίων σύνθεσης για τη σύνθεση WS, παρότι έχουν εστιάσει στο συγκεκριμένο πρόβλημα σχεδιασμού με την υπάρχουσα σύνθεση υπηρεσιών δικτύου, ωστόσο η εργασία που έχει γίνει βρίσκεται σε πρώιμο στάδιο.

Σε αντίθεση με αυτή τη προσέγγιση στη σύνθεση WS, αρκετοί άλλοι ερευνητές χρησιμοποιούν μια άλλη προσέγγιση: χρησιμοποιούν ένα είδος σεναρίου ή μοντέλων ενεργειών που περιγράφουν τον τρόπο για να επιτύχουν ένα στόχο. Αυτό το υψηλού επιπέδου σχέδιο έχει εκπονηθεί και καθοριστεί χρησιμοποιώντας ένα μηχανισμό αυτόματης υποστήριξης. Το πρώτο τέτοιο σύστημα που δημιουργήθηκε ήταν το σύστημα Golog (π.χ. [25]). Μοντελοποιεί τόσο τις εναλλακτικές όσο και τις υπηρεσίες

παροχής πληροφοριών ως ενέργειες με IOPEs. Χρησιμοποιεί τις διαδικασίες Golog για να παρουσιάσει γενικές διαδικασίες για το πως θα πρέπει να γίνουν οι ενέργειες και χρησιμοποιεί online μία συμπερασματική σύνθεση και εκτέλεση για τη δημιουργία μιας σειράς από υπηρεσίες ιστού που είναι προσαρμοσμένες στις επιθυμίες του χρήστη και στους περιορισμούς. Οι ενέργειες συγκέντρωσης πληροφοριών κρίνονται ως απαραίτητες, καθώς οι ενέργειες παγκόσμιας διαφοροποίησης γίνονται ή παρουσιάζονται με σκοπό να δώσουν τη δυνατότητα στο σύστημα να μελετηθεί πριν εκτελεστούν οι υπηρεσίες παγκόσμιας διαφοροποίησης.

Στο ίδιο πνεύμα, κάποιιο άλλοι ερευνητές έχουν χρησιμοποιήσει το παράδειγμα του σχεδιασμού Ιεραρχικών Ενεργειών Δίκτυο (Hierarchical Task Network, HTN) για να εκτελέσουν την αυτοματοποιημένη WS σύνθεση. Σε αυτό το παράδειγμα ο σχεδιαστής έχει στη διάθεσή του μια βιβλιοθήκη δεδομένων σχεδίων, όπου το καθένα χαρακτηρίζεται από αυτό που υποτίθεται ότι θα πρέπει να επιτύχει. Χρησιμοποιεί το σύστημα SHOP2. Για να λυθεί ένα πρόβλημα σύνθεσης, θα πρέπει να δίνεται στο SHOP2 ένα υψηλού επιπέδου σχεδιάγραμμα του σχεδίου που έχει συνδεθεί. Τελικά, πολλά από τα βήματα του σχεδίου περιγράφονται σε ένα υψηλό επίπεδο λεξιλόγιο, που επιτρέπει πολλαπλές εναλλακτικές υποκατηγορίες σχεδίων, για να γίνουν αυτά τα βήματα. Το σύστημα ερευνά τρόπους συνδυασμού αυτών των υποκατηγοριών σχεδίου με σκοπό να φθάσουν σε ένα συνολικό σχέδιο. Κομβικό σημείο στη προσέγγιση για το σχεδιασμό υπηρεσιών με το SHOP2 είναι η αξιοποίηση μιας ακριβούς διάκρισης ανάμεσα στις υπηρεσίες παροχής πληροφοριών και παγκόσμιας διαφοροποίησης στην επεξεργασία σχεδιασμού, δεδομένου ότι η πληροφορία που παρέχεται από τις υπηρεσίες είναι συχνά κρίσιμη στο να βρεθεί ένα σχέδιο. Όταν η χαρτογράφηση από ένα σύνολο περιγραφών υπηρεσιών του OWL-S σε μια SHOP2 κυριότητα, εντοπίζονται υπηρεσίες συγκέντρωσης πληροφοριών και κωδικοποιούνται με τέτοιο τρόπο ώστε να εκτελεστούν κατά την ώρα σχεδιασμού.

Ο σχεδιασμός HTN έχει επίσης χρησιμοποιηθεί για να συνθέσει υπηρεσίες Ιστού στο ταξιδιωτικό πεδίο και στον οργανισμό μιας αλυσίδας παροχής υπηρεσιών B2B. Η βασική ιδέα που εξετάζεται σε αυτή την εργασία είναι ότι οι υπηρεσίες Ιστού επεκτείνουν τις δικές τους ικανότητες διαμέσου της συνεργασίας. Συνεπώς σύμφωνα με την εργασία που παρουσιάστηκε παραπάνω, κατά τη διάρκεια της επεξεργασίας σχεδιασμού, αποτελέσματα και προϋποθέσεις ικανοποιούνται είτε άμεσα χρησιμοποιώντας μια ενέργεια που μπορεί να εκτελεστεί από τις υπηρεσίες Ιστού είτε ζητώντας από άλλες υπηρεσίες ιστού να κάνουν κάτι που θα ικανοποιεί τα

αποτελέσματα ή τις προϋποθέσεις. Ο εντοπισμός των κατάλληλων υπηρεσιών ιστού μπορεί να γίνει χρησιμοποιώντας το λογισμικό ταυτοποίησης OWL-S/UDDI.

Υπάρχουν πολλά συστήματα που ασχολούνται με το πρόβλημα σύνθεσης υπηρεσιών χωρίς να λαμβάνουν υπόψη τις προϋποθέσεις και τις επιδράσεις. Όταν η επεξεργασία της BPEL4WS απαιτεί ένα συγκεκριμένο δεδομένο, που περιγράφεται από ένα τύπο δεδομένων XML, το σύστημα ερευνά για ένα WS, που θα μπορεί να μεταφραστεί από τους διαθέσιμους τύπους στις επιθυμητές δομές. Για παράδειγμα, αν μια επεξεργασία χρειάζεται ένα σύνθετο τύπο που περιέχει την ημερομηνία σε μορφή US και μια γνωστή υπηρεσία παρέχει ένα τύπο δεδομένων πανομοιότυπο εκτός από το ότι η ημερομηνία είναι σε μορφή UK, το σύστημα ερευνά για μια υπηρεσία μετάφρασης που μπορεί να κάνει τον απαιτούμενο μετασχηματισμό δεδομένων. Αν κριθεί απαραίτητο θα διαχωρίσει τη διαδικασία μετατροπής σε υποβήματα και περιοδικά θα ψάχνει για μεθόδους ολοκλήρωσης των υποβημάτων. Μια παρόμοια προσέγγιση ολοκληρώνεται με ένα σύστημα αλληλεπίδρασης της σύνθεσης ενός τελικού χρήστη, το STEER που περιγράφεται στην [26]. Αυτές οι προσεγγίσεις παρουσιάζουν πρωτότυπες λύσεις σε μια σημαντική υποκατηγορία της σύνθεσης υπηρεσιών, που ονομάζεται ενδολειτουργία μεταφοράς δεδομένων. Για να μπορεί να λειτουργήσει, είναι απαραίτητο για τις περιγραφές επεξεργασίας να περιλαμβάνουν ερμηνεύσιμες από τους Η/Υ, περιγραφές των δεδομένων και των αποτελεσμάτων (IO) μιας επεξεργασίας.

Αν και η πρώιμη εργασία που έχει γίνει υπόσχεται πολλά, έχουμε ακόμα κάποια απόσταση από την επίτευξη του στόχου της αυτοματοποιημένης σύνθεσης WS. Έχουμε συμφωνήσει ότι χρειαζόμαστε πλούσιες αναπαραστάσεις των υπηρεσιών ιστού σε μια γλώσσα με καλά καθορισμένη σημασιολογία, για να αυτοματοποιήσουμε τη σύνθεση WS. Πιο συγκεκριμένα χρειάζονται πλούσιες, ακριβείς περιγραφές των IOPE των υπηρεσιών ιστού για να αποφασίσουμε τη σύνθεση, και απαιτούνται πλούσιες αναπαραστάσεις των ιδιοτήτων, των δυνατοτήτων και των λειτουργιών των υπηρεσιών για να ενεργοποιήσουν την επιλογή WS κατά τη διάρκεια επεξεργασίας της σύνθεσης. Έχουμε επιτύχει και τις δύο αυτές απαιτήσεις σε μεγάλο βαθμό με το OWL-S. Αντιθέτως, τα υπάρχοντα βιομηχανικά πρότυπα για την περιγραφή, WS περιγράφουν μόνο τα δεδομένα και τα αποτελέσματα WS και το κάνουν σε μια γλώσσα που δεν είναι εκφραστική και δεν έχει μια καλά καθορισμένη σημασιολογία.

Απαιτούνται επίσης πλούσιες αναπαραστάσεις των σύνθετων επεξεργασιών στο OWL-S, τις οποίες θα χρησιμοποιήσουμε κατά την εργασία σύνθεσης της WS. Η λύση μπορεί να βρεθεί σε μια γλώσσα που είναι περισσότερο εκφραστική από το OWL. Για την πραγματοποίηση του στόχου της αυτοματοποιημένης σύνθεσης της WS, απαιτείται περαιτέρω πρόοδος στο τομέα της αυτοματοποιημένης τεχνολογίας υποστήριξης/σχεδιασμού για την WS. Ένας τελικός φραγμός στο στόχο της αυτοματοποιημένης σύνθεσης WS, είναι η ανάγκη για ευρέως διαδεδομένη υιοθέτηση των περιγραφών OWL-S WS.

Παρά το ότι υπάρχουν πολλά που πρέπει να γίνουν ακόμα, είναι αξιοσημείωτο το τι έχει επιτευχθεί μέχρι στιγμής. Με την υπάρχουσα τεχνολογία, μπορούμε να παρέχουμε αυτοματοποιημένη σύνθεση των υπηρεσιών συγκέντρωσης πληροφοριών. Επίσης έχει αποδειχθεί ότι μπορούμε να αυξήσουμε τις δυνατότητες των εργαλείων ενορχήστρωσης και χορογραφίας, με την τεχνολογία σύνθεσης για τη διαλειτουργικότητα της μεταφοράς δεδομένων και για τη τρέχουσα σύνδεση των υπηρεσιών ιστού. Αυτά τα συστήματα ενεργοποιούν τη χειροκίνητη σύνθεση της WS. Μπορούμε να τα ενισχύσουμε ενοποιώντας σημασιολογικά τις πηγές των δεδομένων. Τελικά μπορούμε να αυτοματοποιήσουμε τη σύνθεση WS, τόσο για τις υπηρεσίες συγκέντρωσης πληροφοριών όσο και για τις ατομικές υπηρεσίες παγκόσμιας διαφοροποίησης κάτω από ελεγχόμενες συνθήκες. Με την υιοθέτηση προσεγγίσεων στην περιγραφή WS όπως αυτή του OWL-S και με τα οφέλη από τις σχετικές τεχνολογίες σχεδιασμού, πιστεύουμε ότι η ευρεία αυτοματοποιημένη σύνθεση WS είναι πολύ κοντά στην πραγματικότητα.

2.2.10 Σχετικές Εργασίες

Κατά τη διάρκεια της εργασίας έχουμε αναφέρει σχετικές εργασίες που αξιοποιούν το OWL-S. Εδώ σημειώνουμε άλλες εργασίες για τις Σημασιολογικές Υπηρεσίες Ιστού που δεν χρησιμοποιούν το DAML-S ή το OWL-S για να περιγράψουν τις υπηρεσίες ιστού.

Οι περισσότερες εργασίες πάνω στην εύρεση υπηρεσιών Ιστού που χρησιμοποιούν το σημασιολογικό Ιστό βασίζονται στο OWL-S. Άλλες εργασίες πάνω στην εύρεση, δεν προϋποθέτουν το OWL-S, και βασίζουν τις περιγραφές υπηρεσιών Ιστού στο MIT Process Handbook. Σε αυτήν την εργασία, η επεξεργασία

ταυτοποίησης βασίζεται στην περιγραφή της ροής εργασιών του μοντέλου επεξεργασίας μιας υπηρεσίας Ιστού και όχι σε μια αφηρημένη αναπαράσταση όπως το Προφίλ OWL-S. Ο μηχανισμός ανάκτησης αντιστοιχεί την αίτηση ανάμεσα με όλα τα μοντέλα διαδικασιών που είναι καταχωρημένα από τις διαθέσιμες υπηρεσίες, μέχρι να ανακτηθεί το μοντέλο διαδικασίας που ταιριάζει με την αίτηση.

Η διαδικασία ταυτοποίησης επιτρέπει στον αιτούντα να ζητήσει υπηρεσίες Ιστού που «κάνουν το χ πριν το ψ». Με άλλα λόγια ο αιτών μπορεί να περιορίσει, όχι μόνο το τύπο της διαδικασίας που εκτελείται και τ' αποτελέσματα που επιτυγχάνονται, αλλά και τον τρόπο με τον οποίο η υπηρεσία επιτυγχάνεται. Αναμφίβολα, αυτό προϋποθέτει ότι ο αιτών και ο προμηθευτής έχουν μια κοινή γνώση για το πως γίνονται οι διαδικασίες. Αυτό σημαίνει ότι ο προμηθευτής και ο αιτών θα πρέπει να μοιράζονται οντολογίες όπως το MIT Process Handbook. Ενώ το OWL-S δεν κάνει κάποιες ισχυρές υποθέσεις στις οντολογίες που χρειάζονται για την εύρεση, όταν αυτές οι υποθέσεις θεωρούνται ότι ισχύουν, μπορούμε να έχουμε παρόμοια αποτελέσματα από τη χρήση διαδικασιών ταυτοποίησης που προτείνονται για την OWL-S, επιλέγοντας πρώτα υπηρεσίες Ιστού με δεδομένη ικανότητα και μετά επιλέγοντας αυτές τις υπηρεσίες των οποίων το μοντέλο επεξεργασίας ικανοποιεί το προσωρινό περιορισμό.

Στην περιοχή της σύνθεσης WS, το μεγαλύτερο μέρος της πρώιμης εργασίας έχει αξιοποιήσει το OWL-S. Πιο πρόσφατα, οι ερευνητές που ασχολούνται με τον σχεδιασμό έχουν αρχίσει να εξετάζουν το πρόβλημα της σύνθεσης WS. Παρόλα αυτά δεν έχουν σαφώς προσδιορίσει το πρόβλημα της περιγραφής υπηρεσιών, πέρα από τη μοντελοποίηση της υπηρεσίας IOPEs ως ενέργειες λογικής πρώτης τάξης.

2.2.11 Σύνοψη

Ο στόχος μας ήταν να δείξουμε πως το OWL-S μπορεί να χρησιμοποιηθεί σύντομα, στο περιεχόμενο των ανερχόμενων προτύπων υπηρεσιών Ιστού, όπως στη WSDL, UDDI και στη BPEL. Έχουμε εξηγήσει ορισμένα από τα βασικά σημεία του OWL-S και τις τεχνικές με τις οποίες μπορεί να χρησιμοποιηθεί μαζί με αυτά τα πρότυπα.

Έχουμε συζητήσει για τα οφέλη των εμπλουτισμένων περιγραφών υπηρεσιών που υποστηρίζονται από το OWL-S, εστιάζοντας αρχικά στις περιγραφές των

δεδομένων, αποτελεσμάτων προϋποθέσεων και επιδράσεων των υπηρεσιών. Στην περιοχή της θέσπισης το OWL-S υποστηρίζει την προδιαγραφή σύνθετων διαδικασιών, και επιτρέπει την ευλύγιστη, δυνατή επίκληση και ενδολειτουργικότητα μεταξύ των υπηρεσιακών πελατών και των προμηθευτών. Επίσης οι μηχανισμοί υποστήριξης του OWL-S επιτρέπουν τις περιγραφές διαδικασιών και τις διαδικασίες θέσπισης να χρησιμοποιούνται μαζί με την WSDL. Στην περιοχή της εύρεσης, το OWL-S επιτρέπει στις εγγραφές υπηρεσιών και στους αλγόριθμους ταυτοποίησης να επωφελούνται από τα δύο βασικά είδη υπηρεσιών, των οποίων η χρήση μπορεί να ολοκληρωθεί με την UDDI. Στην περιοχή της σύνθεσης υπηρεσιών, υπάρχει μια σειρά προσεγγίσεων που επιχειρηματολογούν για τα IOPE του OWL-S, πάνω στην υποστήριξη χειροκίνητων, ημιαυτοματοποιημένων και υπό έλεγχο συνθηκών, και στην αυτοματοποιημένη σύνθεση των υπηρεσιών συγκέντρωσης πληροφοριών και παγκόσμιας διαφοροποίησης.

Συμπεραίνοντας το OWL-S μπορεί να βοηθήσει στη δημιουργία πλήρη αυτοματισμού και δυναμισμού σε πολλές πτυχές των υπηρεσιών Ιστού, μπορεί να υποστηρίξει τη κατασκευή ισχυρών εργαλείων και μεθοδολογιών και να προάγει τη χρήση μιας καλά θεμελιωμένης, σημασιολογικά, υποστήριξης για τις υπηρεσίες. [27]

3 ΚΕΦΑΛΑΙΟ: ΤΟ ΠΡΟΤΥΠΟ OWL-S

3.1 OWL-S: Σημασιολογική Σήμανση Για Τις Υπηρεσίες Δικτύου.

3.1.1 Εισαγωγή

Ο Σημασιολογικός Ιστός θα έπρεπε να έχει μεγαλύτερη πρόσβαση όχι μόνο στο περιεχόμενο, αλλά και στις υπηρεσίες του δικτύου. Χρήστες και πράκτορες λογισμικού θα έπρεπε να έχουν τη δυνατότητα εύρεσης, σύνθεσης και ελέγχου των πηγών του δικτύου προσφέροντας συγκεκριμένες υπηρεσίες και έχοντας συγκεκριμένες ιδιότητες και τέλος θα έπρεπε όλα τα παραπάνω να μπορούν να τα κάνουν σε ένα υψηλό επίπεδο αυτοματισμού, αν το επιθυμούν. Ισχυρά εργαλεία θα έπρεπε να ενεργοποιηθούν από τις υπηρεσίες περιγραφής, κατά τη διάρκεια του κύκλου ζωής των υπηρεσιών δικτύου. Το OWL-S είναι μια οντολογία υπηρεσιών η οποία κάνει αυτή την λειτουργικότητα εφικτή. Σε αυτό το έγγραφο περιγράφουμε τη συνολική δομή της οντολογίας και τα τρία κύρια μέρη της: το προφίλ υπηρεσιών για υπηρεσίες εύρεσης και καταχώρησης, το μοντέλο επεξεργασίας το οποίο παρέχει μια λεπτομερή περιγραφή της λειτουργίας μιας υπηρεσίας και την υποστήριξη η οποία παρέχει λεπτομέρειες για το πως μπορούμε να λειτουργήσουμε με μια υπηρεσία διαμέσου μηνυμάτων.

3.1.2 Οι Υπηρεσίες στον Σημασιολογικό Ιστό

Οι προσπάθειες για τη δημιουργία του σημασιολογικού ιστού προχωρούν με κεκτημένη ταχύτητα. Σύντομα θα είναι εφικτή η πρόσβαση στις πηγές του δικτύου μέσω του περιεχομένου και όχι μόνο μέσω λέξεων - κλειδιά. Μια σημαντική βοήθεια σε αυτή τη κίνηση είναι η ανάπτυξη μιας νέας γενιάς γλωσσών σήμανσης όπως είναι η OWL [28] και η προκάτοχός της DAML+OIL [29]. Οι γλώσσες αυτές βοηθούν στην δημιουργία οντολογιών για κάθε κυριότητα και στην δημιουργία στιγμιότυπων αυτών των οντολογιών στην περιγραφή ειδικών συγκεκριμένων υπηρεσιών δικτύου.

Ανάμεσα στις πιο σημαντικές πηγές δικτύου, είναι αυτές που παρέχουν υπηρεσίες. Με τον όρο υπηρεσία εννοούμε τις ιστοσελίδες που βρίσκονται στο δίκτυο, οι οποίες δεν παρέχουν μόνο στατική ενημέρωση, αλλά μας επιτρέπουν μέσω κάποιων ενεργειών να δημιουργούμε αλλαγές, όπως η πώληση ενός προϊόντος. Ο Σημασιολογικός Ιστός θα έπρεπε να καθιστά ικανούς τους χρήστες να εντοπίζουν, να επιλέγουν, να κατέχουν, να συνθέτουν και να ελέγχουν τις υπηρεσίες δικτύου αυτόματα.

Για να μπορούμε να χρησιμοποιήσουμε μια υπηρεσία δικτύου, ο πράκτορας λογισμικού θα πρέπει να έχει μια περιγραφή της υπηρεσίας ερμηνεύσιμη από Η/Υ. Ένας σημαντικός στόχος των γλωσσών σήμανσης (mark up) του Σημασιολογικού Ιστού είναι να θέσουν ένα πλαίσιο εργασίας μέσα στο οποίο οι περιγραφές αυτές θα δημιουργούνται και θα μοιράζονται. Οι ιστοσελίδες θα έπρεπε να μπορούν να κατέχουν μια σταθερή οντολογία, που θα περιλαμβάνει ένα σύνολο από βασικές τάξεις και ιδιότητες, για τη δήλωση και περιγραφή υπηρεσιών. Οι μηχανισμοί δόμησης οντολογιών της OWLS, παρέχουν ένα κατάλληλο πλαίσιο εργασίας, με μια γλώσσα αναπαράστασης συμβατή με το δίκτυο, ώστε να γίνουν τα παραπάνω.

Αυτό το κείμενο περιγράφει μια συνεργατική προσπάθεια από ερευνητές σε διάφορους οργανισμούς, για να καθορίσουν μια τέτοια οντολογία. Αυτή την οντολογία αποκαλούμε, OWL-S. Σε αυτά που ακολουθούν, θα ενεργοποιήσουμε πρώτα την OWL-S σε όρους απλών εργασιών που έχει σχεδιαστεί να υποστηρίζει. Στο κεντρικό μέρος του κειμένου περιγράφουμε μια ανώτερη οντολογία για υπηρεσίες που έχουμε αναπτύξει, συμπεριλαμβάνοντας και τις υπο-οντολογίες τους για τα προφίλ, τις διαδικασίες και τις υποστηρίξεις τους. Η οντολογία ακόμα εξελίσσεται και δημιουργεί σχέσεις σε άλλες προσπάθειες ανάπτυξης, όπως τη δημιουργία οντολογιών για το χρόνο και τις πηγές. Ορισμένες φορές θα αναφερθούμε στην οντολογία OWL-S, σαν μια γλώσσα για την περιγραφή υπηρεσιών, αναγνωρίζοντας το γεγονός ότι παρέχει ένα σταθερό λεξιλόγιο το οποίο μπορεί να χρησιμοποιηθεί και με άλλους παράγοντες της περιγραφικής γλώσσας OWL για να δημιουργεί περιγραφές υπηρεσιών.

3.1.3 Παραδείγματα

Οι ατομικές υπηρεσίες είναι αυτές όπου ένα μονοδιάστατο πρόγραμμα, προσβάσιμο μέσω δικτύου, αισθητήριο όργανο, ή ένα μηχάνημα επικαλείται μέσα

από ένα μήνυμα αίτησης, εκτελεί την εργασία του και πιθανώς παράγει μια μοναδική απάντηση στον αιτών. Με τις ατομικές υπηρεσίες δεν υπάρχει καμία αλληλεπίδραση ανάμεσα στο χρήστη και την υπηρεσία. Π.χ. όταν μία υπηρεσία επιστρέφει ένα ταχυδρομικό κωδικό ή το γεωγραφικό μήκος όταν του δίνεται η διεύθυνση, ανήκει σ' αυτή την κατηγορία. Αντιθέτως, οι πολύπλοκες ή σύνθετες υπηρεσίες συνθέτονται από πολλαπλές πρωτογενείς υπηρεσίες και μπορεί να απαιτούν μία εκτεταμένη αλληλεπίδραση ή μία συναλλαγή ανάμεσα στον αιτούντα και το σύνολο των υπηρεσιών που χρησιμοποιούνται. Η διαδικασία αλληλεπίδρασης ενός χρήστη με το www.amazon.com προκειμένου να αγοράσει ένα βιβλίο έχει ως εξής: Ο χρήστης ψάχνει για βιβλία έχοντας διάφορα κριτήρια, πιθανόν να διαβάζει κάποιες κριτικές, μπορεί να το αγοράσει ή να μην το αγοράσει και δίνει πληροφορίες για την αποστολή και για την πιστωτική του κάρτα. Το OWL-S έχει δημιουργηθεί για να υποστηρίξει και τις δύο κατηγορίες των υπηρεσιών. Οι επόμενοι τρεις τύποι εργασιών θα δώσουν στον αναγνώστη μια ιδέα, για τα είδη των εργασιών που θα μπορεί να υποστηρίξει το OWL-S.

1. Αυτόματη Εύρεση Υπηρεσιών Δικτύου

Η αυτόματη εύρεση υπηρεσιών δικτύου είναι μια αυτόματη διαδικασία για τον εντοπισμό υπηρεσιών δικτύου που παρέχουν ένα ειδικό επίπεδο ικανοτήτων των υπηρεσιών, ενώ διατηρούν τους περιορισμούς που έχουν οριστεί από τους πελάτες. Π.χ. ο χρήστης μπορεί να θέλει να βρει μια υπηρεσία που πουλάει αεροπορικά εισιτήρια ανάμεσα σε 2 δοθείσες πόλεις και δέχεται μία συγκεκριμένη πιστωτική κάρτα. Μέχρι στιγμής η ενέργεια αυτή θα πρέπει να γίνει από έναν χρήστη που πιθανότατα θα χρησιμοποιήσει μία μηχανή αναζήτησης για να βρει την υπηρεσία, θα διαβάσει την ιστοσελίδα, και θα το κάνει χειροκίνητα, ούτως ώστε να αποφασίσει αν συμφωνεί με τους περιορισμούς. Με την OWL-S σήμανση των υπηρεσιών, η πληροφορία που είναι απαραίτητη για την εύρεση των υπηρεσιών δικτύου θα μπορούσε να καθορισθεί στην ιστοσελίδα της υπηρεσίας ως μια μεταφράσιμη από τον Η/Υ σημασιολογική σήμανση. Η καταχώρηση μιας υπηρεσίας ή μιας μηχανής αναζήτησης εμπλουτισμένης με οντολογίες, θα μπορούσε να χρησιμοποιηθεί για να εντοπίζει τις υπηρεσίες αυτόματα. Εναλλακτικά, ο εξυπηρετητής θα μπορούσε πρώτα να διαφημίσει τον ίδιο στο OWL-S με μία καταχώρηση υπηρεσίας, ενδιάμεσος πράκτορας, έτσι ώστε οι αιτούντες να μπορούν να τον βρουν όταν ψάχνουν τις

καταχωρήσεις. Επιπλέον το OWL-S επιτρέπει επεξηγηματικές καταχωρήσεις για τις ιδιότητες και τις ικανότητες μιας υπηρεσίας, οι οποίες μπορεί να χρησιμοποιηθούν για την αυτόματη εύρεση μιας υπηρεσίας.

2. Υπηρεσίες Δικτύου Αυτόματης Επίκλησης

Η αυτόματη επίκληση υπηρεσιών δικτύου είναι η αυτόματη επίκληση μιας υπηρεσίας δικτύου από ένα πρόγραμμα υπολογιστή ή από ένα πράκτορα, όταν δίνεται η επεξηγηματική περιγραφή αυτής της υπηρεσίας, σε αντίθεση με την περίπτωση όπου ο πράκτορας έχει προγραμματιστεί νωρίτερα, για να καλέσει τη συγκεκριμένη υπηρεσία. Αυτή η διαδικασία είναι απαραίτητη ώστε να μπορεί παραδείγματος χάριν ο χρήστης να ζητήσει την αγορά, από μία ιστοσελίδα η οποία βρέθηκε από τον χρήστη και επιλέχθηκε από αυτόν, ενός αεροπορικού εισιτηρίου για μια συγκεκριμένη πτήση. Η εκτέλεση μίας υπηρεσίας Δικτύου μπορεί να θεωρηθεί ως μία συλλογή απομακρυσμένων κλήσεων. Η OWL-S σήμανση των υπηρεσιών Δικτύου παρέχει μία επεξηγηματική, μεταφράσιμη από H/Y API, η οποία περιέχει τη σημασιολογία των επιχειρημάτων που χρειάζονται για τη διεκπεραίωση της κλήσης και τη σημασιολογία αυτών που επιστρέφονται από τα μηνύματα, όταν η εκτέλεση των υπηρεσιών πετυχαίνει ή αποτυγχάνει. Ένας ατζέντης λογισμικού θα πρέπει να είναι σε θέση να μεταφράζει τη σήμανση, για να αναγνωρίζει ποια είναι τα απαραίτητα δεδομένα εισόδου για την επίκληση μιας υπηρεσίας και τι πληροφορίες θα επιστραφούν. Το OWL-S, σε αντίθεση με το πεδίο ορισμού των οντολογιών που καθορίζονται από το OWL, παρέχει σταθερά μέσα για τον ξεκάθαρο ορισμό των API για τις υπηρεσίες δικτύου, τα οποία καθιστούν δυνατή την, αυτού του είδους, αυτόματη εκτέλεση των Υπηρεσιών Δικτύου.

3. Αυτόματη σύνθεση και εσωτερική λειτουργία των Υπηρεσιών Δικτύου

Αναφέρεται στην αυτόματη επιλογή, σύνθεση και εσωτερική λειτουργία των υπηρεσιών δικτύου για να τελέσουν κάποιες σύνθετες ενέργειες, δεδομένης μιας υψηλού επιπέδου περιγραφής του στόχου. Παραδείγματος χάριν, ο χρήστης μπορεί να θέλει να κάνει όλους τους απαραίτητους ταξιδιωτικούς διακανονισμούς για να πραγματοποιήσει ένα ταξίδι προκειμένου να παραστεί σε ένα συνέδριο. Όπως έχουν

τα πράγματα μέχρι στιγμής, ο χρήστης θα πρέπει να επιλέξει τις υπηρεσίες δικτύου, να καθορίσει τη σύνθεση χειροκίνητα, και να βεβαιωθεί ότι κάθε λογισμικό που χρειάζεται για την ενδολειτουργία των υπηρεσιών που θα πρέπει να μοιραστούν πληροφορίες, έχει δημιουργηθεί κατά παραγγελία. Με την γλώσσα σήμανσης OWL-S, οι απαραίτητες πληροφορίες για την επιλογή και τη σύνθεση υπηρεσιών, θα είναι κωδικοποιημένες μέσα στις ιστοσελίδες της υπηρεσίας. Το λογισμικό μπορεί να είναι γραμμένο με τέτοιο τρόπο ώστε να εκμεταλλεύεται τις αναπαραστάσεις, μαζί με ένα καθορισμό των στόχων αυτής της ενέργειας, ώστε να επιτυγχάνει το στόχο αυτόματα. Για να το υποστηρίξει, το OWL-S παρέχει επακριβείς καθορισμούς των προϋποθέσεων και των συνεπειών της αίτησης των ξεχωριστών υπηρεσιών, καθώς και μια γλώσσα που θα περιγράφει τη σύνθεση των υπηρεσιών και τις αλληλεπιδράσεις της ροής δεδομένων.

Κάθε προσπελάσιμο πρόγραμμα/ αισθητήρας/ μηχάνημα που είναι δηλωμένο ως υπηρεσία θα αντιμετωπίζεται ως υπηρεσία. Το OWL-S δεν αποκλείει ότι απλές και στατικές ιστοσελίδες μπορεί να είναι υπηρεσίες. Αλλά ο κύριος στόχος μας στο καθορισμό του OWL-S είναι να υποστηρίξει πιο σύνθετες λειτουργίες από αυτές που περιγράψαμε παραπάνω.

3.1.4 Μια Ανώτερη Οντολογία Για Τις Υπηρεσίες

Η δομή της οντολογίας των υπηρεσιών μας, είναι παρακινήμενη από την ανάγκη να παρέχει 3 βασικούς τύπους γνώσης για την υπηρεσία. Κάθε μια από τις οποίες θα χαρακτηρίζεται από την ερώτηση που απαντά:

- *Τι παρέχει η υπηρεσία στους πιθανούς πελάτες;*

Η απάντηση σε αυτήν την ερώτηση δίνεται από το προφίλ που χρησιμοποιείται για να διαφημίσει την υπηρεσία. Κάθε στιγμιότυπο της τάξης Υπηρεσία παρουσιάζει ένα ServiceProfile.

- *Πως χρησιμοποιείται;*

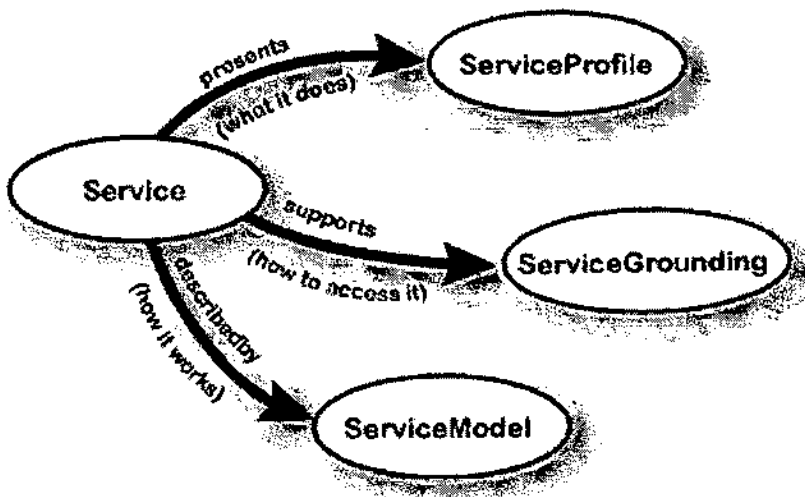
Η απάντηση στην ερώτηση αυτή δίνεται από το μοντέλο διαδικασίας. Αυτή η πτυχή καλύπτεται από τη τάξη του ServiceModel. Στιγμιότυπα της τάξης Service χρησιμοποιούν την ιδιότητα describedBy για να αναφερθούν στο ServiceModel της υπηρεσίας.

- Πως πραγματοποιείται η αλληλεπίδραση;

Η απάντηση στην ερώτηση αυτή δίνεται στην υποστήριξη. Η υποστήριξη παρέχει τις απαραίτητες λεπτομέρειες για τα πρωτόκολλα μεταφοράς. Στιγμιότυπα της τάξης Service έχουν μια ιδιότητα υποστήριξης η οποία αναφέρεται στο ServiceGrounding.

Η τάξη Service μας δίνει ένα οργανωτικό σημείο αναφοράς για μια δηλωμένη υπηρεσία δικτύου, ένα στιγμιότυπο της τάξης Service θα υπάρχει για κάθε δηλωμένη υπηρεσία. Οι ιδιότητες presents, describedBy, και supports, είναι ιδιότητες της τάξης Service. Οι τάξεις ServiceProfile, ServiceModel και ServiceGrounding είναι οι περιοχές που αντιστοιχούν αυτές οι ιδιότητες. Κάθε στιγμιότυπο της τάξης Service θα παρουσιάζει (*present*) τη περιγραφή του ServiceProfile, θα περιγράφεται από μια (*describedBy*) περιγραφή του ServiceModel και θα υποστηρίζει (*support*) το ServiceGrounding. Οι λεπτομέρειες των προφίλ, των μοντέλων και των υποστηρίξεων μπορεί να ποικίλλουν αρκετά από τον ένα τύπο της υπηρεσίας στον άλλο, δηλαδή από ένα στιγμιότυπο της τάξης Service σε ένα άλλο. Αλλά κάθε μια από τις τρεις πτυχές της υπηρεσίας παρέχουν ένα απαραίτητο τύπο πληροφοριών για την υπηρεσία, όπως περιγράφουμε παρακάτω.

Γενικότερα μιλώντας, το ServiceProfile παρέχει τις απαραίτητες πληροφορίες για ένα πράκτορα για να βρει μια υπηρεσία, ενώ παράλληλα το ServiceModel μαζί με το ServiceGrounding παρέχουν αρκετές πληροφορίες ώστε να μπορεί ένας πράκτορας να χρησιμοποιήσει την υπηρεσία, εφόσον τη βρει.



Σχήμα 7 - Το Υψηλότερο Επίπεδο Οντολογίας σε μια Υπηρεσία

Το προφίλ της υπηρεσίας παρουσιάζει το «τι κάνει μια υπηρεσία» με τρόπο κατάλληλο, για ένα πράκτορα που αναζητεί υπηρεσίες ή για έναν πράκτορα που ψάχνει εκ μέρους άλλου πράκτορα που ψάχνει υπηρεσίες, ώστε να μπορεί να καθορίσει εάν η υπηρεσία καλύπτει τις ανάγκες του. Αυτή η μορφή αντιπροσώπευσης παρέχει μια περιγραφή του τι επιτυγχάνεται από την υπηρεσία, τους περιορισμούς στην εφαρμοσιμότητα και στην ποιότητα της υπηρεσίας, καθώς και τις απαιτήσεις που θα πρέπει να ικανοποιήσει ο αιτών για να κάνει χρήση της υπηρεσίας επιτυχώς.

Το μοντέλο υπηρεσίας ενημερώνει τον πελάτη πως να χρησιμοποιεί την υπηρεσία, περιγράφοντας το σημασιολογικό περιεχόμενο των απαιτήσεων, τις συνθήκες κάτω από τις οποίες θα προκύψουν συγκεκριμένα αποτελέσματα, και όπου είναι αναγκαίο, οι βήμα προς βήμα διαδικασίες που θα οδηγήσουν σε αυτά τα αποτελέσματα. Περιγράφει πως ζητείται η υπηρεσία και τι συμβαίνει όταν η υπηρεσία εκτελείται. Για ασυνήθιστες υπηρεσίες (αυτές που συνθέτονται σε διάφορα βήματα μέσα στον χρόνο), αυτή η περιγραφή μπορεί να χρησιμοποιηθεί από ένα αιζέντη που ψάχνει για υπηρεσίες, με το λιγότερο 4 διαφορετικούς τρόπους.

(1) Να γίνει μια σε βάθος ανάλυση για να δει αν η υπηρεσία ικανοποιεί τις απαιτήσεις του.

(2) Να συνθέσει περιγραφές υπηρεσίας από πολλαπλές υπηρεσίες για να εκτελέσει μια συγκεκριμένη ενέργεια.

(3) Κατά τη διάρκεια του σταδίου, της θεσμοθέτησης της υπηρεσίας να συντονίσει τις ενέργειες των διαφόρων συμμετεχόντων.

(4) Να ελέγχει την εκτέλεση της υπηρεσίας.

Η υποστήριξη μιας υπηρεσίας αναφέρει τις λεπτομέρειες για το πως ο πράκτορας μπορεί να έχει πρόσβαση στην υπηρεσία. Τυπικά η υποστήριξη αναφέρεται σε ένα πρωτόκολλο επικοινωνίας, στους τύπους μηνυμάτων και σε άλλες ειδικές λεπτομέρειες για την υπηρεσία όπως οι αριθμοί θύρας που χρησιμοποιούνται για την επικοινωνία με την υπηρεσία. Επίσης, η υποστήριξη θα πρέπει να ορίζει για κάθε σημασιολογικό τύπο δεδομένων ή αποτελεσμάτων που αναφέρονται στο Μοντέλο Υπηρεσίας (Service Model), έναν ευκρινή τρόπο ανταλλαγής δεδομένων στοιχείων τέτοιου τύπου με την υπηρεσία.

Η ανώτερη οντολογία της υπηρεσίας, αναφέρει μόνο δύο σύνολα στοιχείων περιορισμού: Μια υπηρεσία μπορεί να περιγράψει από το πολύ ένα μοντέλο

υπηρεσίας και η υποστήριξη πρέπει να σχετίζεται με μια μόνο υπηρεσία. Η ανώτερη οντολογία σκόπιμα δεν ορίζει κανένα ελάχιστο σύνολο για τις ιδιότητες presents ή described by. Ούτε όμως η ανώτερη οντολογία ορίζει μέγιστο σύνολο για τις ιδιότητες presents ή supports.

Τέλος, θα πρέπει να σημειώσουμε ότι ενώ καθορίζουμε μια συγκεκριμένη ανώτερη οντολογία για το προφίλ, μια για το μοντέλο της υπηρεσίας και μια για την υποστήριξη, παρόλα αυτά το OWL-S επιτρέπει την δημιουργία εναλλακτικών προσεγγίσεων σε κάθε περίπτωση. Ο σκοπός μας εδώ δεν είναι να καθορίσουμε μια απλή προσέγγιση για κάθε μια από τις τρεις περιοχές, αλλά περισσότερο να παρέχουμε προκαθορισμένες προσεγγίσεις που θα είναι χρήσιμες στη πλειονότητα των περιπτώσεων.

3.1.5 Προφίλ Υπηρεσιών

Μια συναλλαγή σε μια δικτυακή αγορά αφορά 3 μέρη: τους αιτούντες, τον παροχέα της υπηρεσίας και τους παράγοντες υποδομής. Ο αιτών, συνήθως είναι αγοραστής, ψάχνει μια υπηρεσία για να ολοκληρώσει την εργασία του. Ο παροχέας της υπηρεσίας ο οποίος μπορεί να αναγνωριστεί και ως πωλητής, παρέχει την υπηρεσία που απαιτείται από τον αιτούντα. Σε ένα ανοικτό περιβάλλον όπως το Διαδίκτυο, ο αιτών μπορεί να μην γνωρίζει άμεσα την ύπαρξη του προμηθευτή, έτσι βασίζεται στους παράγοντες υποδομής που λειτουργούν ως καταχωρήσεις για να βρει τον κατάλληλο προμηθευτή. Ο ρόλος των καταχωρήσεων, είναι η ταυτοποίηση του αιτήματος με τις προσφορές των προμηθευτών και να αναγνωρίζουν ποιες από αυτές ταιριάζουν περισσότερο. Μέσα στα πλαίσια του OWL-S, το προφίλ υπηρεσιών (Service Profile) παρέχει ένα τρόπο να περιγράφει τις υπηρεσίες που προσφέρονται από τους προμηθευτές και ζητούνται από τους πελάτες.

Το προφίλ υπηρεσιών δεν διατάσσει καμία παρουσίαση των υπηρεσιών. Χρησιμοποιώντας μια υποκατηγορία τάξης του OWL-S είναι πιθανό να δημιουργήσουμε ειδικές παρουσιάσεις υπηρεσιών οι οποίες θα μπορούν να χρησιμοποιηθούν ως προφίλ υπηρεσιών. Το OWL-S παρέχει μια πιθανή παρουσίαση μέσα από τη τάξη του προφίλ. Το προφίλ του OWL-S περιγράφει μια υπηρεσία ως μια λειτουργία 3 βασικών τύπων πληροφοριών: τι οργάνωση παρέχει η υπηρεσία, τι λειτουργία η υπηρεσία υπολογίζει, και μια σειρά χαρακτηριστικών οι

οποίες εξειδικεύουν τα χαρακτηριστικά της υπηρεσίας. Τα τρία μέρη πληροφοριών επαναλαμβάνονται με τη σειρά παρακάτω.

Ο προμηθευτής πληροφοριών περιέχει πληροφορίες επικοινωνίας που αναφέρονται στην οντότητα που παρέχει την υπηρεσία, π.χ. οι πληροφορίες επικοινωνίας μπορεί να αναφέρονται στον χειριστή συντήρησης που είναι υπεύθυνος για τη λειτουργία της υπηρεσίας ή σε ένα αντιπροσωπευτικό πελάτη που μπορεί να παρέχει πρόσθετες πληροφορίες για την υπηρεσία.

Η λειτουργική περιγραφή της υπηρεσίας εκφράζεται σε όρους της μετατροπής που παράγεται από την υπηρεσία. Ειδικότερα, αναφέρεται στην εισαγωγή δεδομένων που απαιτούνται από την υπηρεσία και στα αποτελέσματα που γεννούνται. Επιπροσθέτως, όταν μια υπηρεσία μπορεί να απαιτεί εξωγενείς συνθήκες για να ικανοποιηθεί και μπορεί να έχει επιρροή στην αλλαγή τέτοιων συνθηκών, το προφίλ περιγράφει τις προϋποθέσεις που απαιτούνται από την υπηρεσία και τις αναμενόμενες επιρροές που δημιουργούνται από την εκτέλεση της υπηρεσίας. Παραδείγματος χάριν μια υπηρεσία πωλήσεων μπορεί να απαιτεί ως προϋπόθεση μια έγκυρη πιστωτική κάρτα και σαν εισαγωγή δεδομένων, να έχει τον αριθμό της πιστωτικής κάρτας και την ημερομηνία λήξης της. Ως αποτέλεσμα θα έχουμε μια απόδειξη και σαν επιρροή τη χρέωση της κάρτας.

Τελικά το προφίλ επιτρέπει την περιγραφή των ιδιοτήτων που χρησιμοποιούνται για να περιγράψουν τα χαρακτηριστικά της υπηρεσίας. Ο πρώτος τύπος πληροφοριών ορίζει την κατηγορία της δοθείσας υπηρεσίας π.χ. την κατηγορία της υπηρεσίας μέσα στο UNSPSC σύστημα κατάταξης. Ο δεύτερος τύπος πληροφοριών αναφέρεται στην εκτίμηση της ποιότητας της υπηρεσίας: ορισμένες υπηρεσίες μπορεί να είναι πολύ καλές, αξιόπιστες και γρήγορες στην ανταπόκριση, άλλες μπορεί να είναι αναξιόπιστες ή αργές. Πριν χρησιμοποιήσει μια υπηρεσία, ο αιτών μπορεί να θέλει να ελέγξει με τι είδους υπηρεσία θα συνεργαστεί. Για αυτό το λόγο μπορεί να θέλει να δημοσιοποιήσει την βαθμολογία του μέσα σε κάποιο ειδικό σύστημα βαθμολόγησης ποιότητας, ούτως ώστε να εκθέσει την ποιότητα των υπηρεσιών που παρέχει. Είναι θέμα του αιτούντα μετά, να χρησιμοποιήσει τις πληροφορίες αυτές για να εξακριβώσει αν είναι όντως σωστές και να αποφασίσει τι θα πράξει. Ο τελευταίος τύπος πληροφοριών είναι μια προαιρετική λίστα παραμέτρων της υπηρεσίας που μπορεί να περιέχει οποιοδήποτε τύπο πληροφορίας. Το προφίλ του OWL-S παρέχει ένα μηχανισμό για την παρουσίαση

τέτοιων παραμέτρων, οι οποίοι μπορεί να εσωκλείουν παραμέτρους που να παρέχουν μια εκτίμηση του μέγιστου χρόνου ανταπόκρισης.

3.1.5.1 Συνθέτοντας το Προφίλ: Η Σχέση με το Μοντέλο Διαδικασίας

Το προφίλ μιας υπηρεσίας παρέχει μια ακριβή περιγραφή της υπηρεσίας σε μια καταχώρηση, αλλά εφόσον η υπηρεσία επιλεγεί, το προφίλ δεν έχει πλέον χρησιμότητα. Αντιθέτως ο αιτών θα χρησιμοποιήσει το μοντέλο διαδικασίας για να ελέγξει την αλληλεπίδραση με την υπηρεσία. Παρόλο που το προφίλ και το μοντέλο διαδικασίας παίζουν διαφορετικούς ρόλους κατά την διάρκεια μιας συναλλαγής ανάμεσα στις υπηρεσίες δικτύου, είναι δύο διαφορετικές παρουσιάσεις της ίδιας υπηρεσίας, επομένως, είναι φυσικό να περιμένουμε ότι τα δεδομένα, τα αποτελέσματα, οι προϋποθέσεις, τα αποτελέσματα και οι επιρροές (IOPE) της μιας θα επηρεάζουν το IOPE της άλλης.

Το OWL-S δεν υπαγορεύει κανένα περιορισμό ανάμεσα στο προφίλ και το μοντέλο διαδικασίας, έτσι ώστε οι δύο περιγραφές μπορεί να είναι ασύμβατες χωρίς να επηρεάζεται η εγκυρότητα έκφρασης του OWL-S. Ωστόσο, αν το προφίλ παρουσιάζει μια υπηρεσία που δεν είναι συμβατή, με την υπηρεσία που παρουσιάζει το μοντέλο διαδικασίας, η αλληλεπίδραση θα σπάσει σε κάποιο σημείο. Σαν ένα ακραίο παράδειγμα, φανταστείτε μια υπηρεσία που διαφημίζει ένα ταξιδιωτικό ατζέντη, αλλά υιοθετεί το μοντέλο διαδικασίας ενός ατζέντη πωλητή βιβλίων. Θα επιλεγεί να κλείνει ταξίδια, αλλά θα αποτύχει γιατί θα ζητάει τίτλους βιβλίων και ISBN αριθμούς. Από την άλλη πλευρά, ποτέ δεν θα επιλεγεί από υπηρεσίες που επιθυμούν την αγορά βιβλίων, έτσι δεν θα πουλάει ούτε βιβλία.

Η επιλογή του IOPE για να περιγράψει το προφίλ, είναι μια διαδικασία που μπορεί να ξεγελάσει. Θα πρέπει να αποφύγει την λανθασμένη καταχώρηση στοιχείων της υπηρεσίας, έτσι θα απαιτούνται όλα τα IOPE που χρησιμοποιούνται στο μοντέλο διαδικασίας. Από την άλλη πλευρά ορισμένα από τα IOPE μπορεί να είναι τόσο γενικά που να μην περιγράφουν την υπηρεσία. Ένα άλλο πράγμα που πρέπει να αναφέρουμε είναι ο αλγόριθμος καταχώρησης για την ταυτοποίηση των αιτήσεων με τους προμηθευτές. Επιπλέον το προφίλ προσδιορίζει με σαφήνεια τον σκοπό της υπηρεσίας. Διαφημίζει αυτές τις λειτουργίες που η υπηρεσία επιθυμεί να παρέχει, ενώ παράλληλα μπορεί να κρύβει (δεν δηλώνει δημόσια) άλλες λειτουργίες.

Παραδείγματος χάριν, ας θεωρήσουμε μια υπηρεσία πώλησης βιβλίων που μπορεί να περιέχει 2 λειτουργίες. Η πρώτη επιτρέπει στις άλλες υπηρεσίες να φυλλομετρούν την ιστοσελίδα της, για να βρουν βιβλία δικού τους ενδιαφέροντος και η άλλη επιτρέπει στους χρήστες να αγοράσουν τα βιβλία που βρήκαν. Ο πωλητής βιβλίων έχει την επιλογή να διαφημίσει μόνο τη λειτουργία αγοράς βιβλίων ή ταυτόχρονα τη λειτουργία αγοράς και φυλλομέτρησης μαζί. Στην τελευταία περίπτωση η υπηρεσία δημοσιοποιεί ότι μπορεί να παρέχει υπηρεσίες φυλλομέτρησης και επιτρέπει σε όλους να φυλλομετρούν τις εγγραφές της χωρίς να αγοράσουν βιβλίο. Σε αντίθεση με το να διαφημίζει μόνο την λειτουργία πώλησης βιβλίων, όχι όμως τη φυλλομέτρηση, αποθαρρύνει τους αιτούντες, που δεν σκοπεύουν να αγοράσουν, από τη διαδικασία φυλλομέτρησης. Η απόφαση για το ποιες λειτουργίες θα διαφημιστούν δείχνει πως η υπηρεσία θα χρησιμοποιηθεί. Ο αιτών που σκοπεύει να φυλλομετρήσει και όχι να αγοράσει, θα επέλεγε μια υπηρεσία που διαφημίζει μαζί τις λειτουργίες αγοράς και φυλλομέτρησης, και όχι μία που διαφημίζει μόνο τη δυνατότητα αγοράς.

Στην έως τώρα περιγραφή, έχουμε συμφωνήσει σε ένα μοντέλο εγγραφής στο οποίο οι δυνατότητες της υπηρεσίας διαφημίζονται και έπειτα ταυτοποιούνται με τις απαιτήσεις για την υπηρεσία. Αυτό το μοντέλο υιοθετείται από εγγραφές όπως η UDDI. Ενώ αυτό είναι το πιο πιθανό μοντέλο που υιοθετείται από τις υπηρεσίες δικτύου, είναι πιθανές και άλλες μορφές εγγραφής. Παραδείγματος χάριν, όταν η ζήτηση για μια υπηρεσία είναι μεγαλύτερη από την προσφορά, οι διαφημιστικές ανάγκες για την υπηρεσία είναι πιο ουσιώδεις από την διαφήμιση προσφερόμενων υπηρεσιών, εφόσον ο προμηθευτής μπορεί να επιλέξει την επόμενη αίτηση. Παράλληλα σε μια P2P αρχιτεκτονική δεν θα υπήρχε καμία εγγραφή. Πράγματι, οι τύποι εγγραφής ποικίλλουν και μέχρι τώρα υπάρχουν 28 διαφορετικοί τύποι. Χρησιμοποιώντας μια επεξηγηματική παρουσίαση των υπηρεσιών δικτύου, το προφίλ υπηρεσίας δεν δεσμεύεται σε καμία φόρμα εγγραφής, αλλά μπορεί να χρησιμοποιηθεί σε όλες αυτές. Εφόσον, το προφίλ της υπηρεσίας παρουσιάζει μαζί τις παροχές και τις απαιτήσεις της υπηρεσίας, τότε μπορεί να χρησιμοποιηθεί σε μια αντίστροφη εγγραφή η οποία καταγράφει τις ανάγκες και τα ερωτήματα στις προσφορές. Το Προφίλ υπηρεσίας μπορεί να χρησιμοποιηθεί και στους 28 τύπους εγγραφής.

3.1.5.2 Οι Ιδιότητες του Προφίλ

Παρακάτω περιγράφουμε λεπτομερώς τα κύρια μέρη του προφίλ. Τα κατηγοριοποιούμε σε 4 τομείς. Στην πρώτη περιγράφουμε τις ιδιότητες που συνδέουν το Προφίλ υπηρεσιών με την τάξη Υπηρεσία και το Μοντέλο Διαδικασίας, ο δεύτερος τομέας περιγράφει τη φόρμα πληροφοριών επικοινωνίας και την περιγραφή του προφίλ. Αυτή η πληροφορία συνήθως χρησιμεύει για ανθρώπινη κατανάλωση. Στο τρίτο τομέα θα αναφερθούμε στη λειτουργική παρουσίαση σε σχέση με το IOPE. Τέλος στο τελευταίο τομέα περιγράφουμε τα χαρακτηριστικά του προφίλ.

1. Προφίλ Υπηρεσιών

Η τάξη Προφίλ Υπηρεσίας (Service Profile) παρέχει μια υπερ-τάξη για κάθε τύπο υψηλού επιπέδου περιγραφής της υπηρεσίας. Το Προφίλ υπηρεσίας δεν διατάσσει καμία παρουσίαση της υπηρεσίας, αλλά διευθύνει τις βασικές πληροφορίες για να συνδέσει κάθε στιγμιότυπο του προφίλ με ένα στιγμιότυπο της υπηρεσίας.

Υπάρχει μια αμφίδρομη σχέση μεταξύ της υπηρεσίας και του προφίλ, ούτως ώστε η υπηρεσία να μπορεί να σχετίζεται με το προφίλ και το προφίλ με την υπηρεσία. Αυτές οι σχέσεις εκφράζονται με τις ιδιότητες `presents` και `presentsBy`.

Presents: περιγράφει τη σχέση ανάμεσα στο στιγμιότυπο μιας υπηρεσίας και στο στιγμιότυπο του προφίλ, ουσιαστικά υποδηλώνει ότι η υπηρεσία περιγράφεται από το προφίλ.

PresentsBy: είναι το αντίστροφο του `presents`. Ορίζει ότι ένα δεδομένο προφίλ περιγράφει μια υπηρεσία.

2. Όνομα Υπηρεσίας, Επαφές Και Περιγραφή

Ορισμένες ιδιότητες παρέχουν αναγνώσιμη από τον άνθρωπο πληροφορία, η οποία είναι απίθανο να επεξεργαστεί αυτόματα. Αυτές οι ιδιότητες εμπεριέχουν το `serviceName`, `textDescription` και το `contactInformation`.. Ένα προφίλ μπορεί να έχει το πολύ ένα όνομα υπηρεσίας και μια περιγραφή κειμένου, αλλά όσα αντικείμενα πληροφοριών επικοινωνίας θέλει ο προμηθευτής να προσφέρει.

service name: αναφέρεται στο όνομα υπηρεσίας που προσφέρεται. Μπορεί να χρησιμοποιηθεί και ως μηχανισμός αναγνώρισης της υπηρεσίας.

textDescription: παρέχει μια σύντομη περιγραφή της υπηρεσίας. Συνοψίζει τι προσφέρει η υπηρεσία, περιγράφει τι χρειάζεται η υπηρεσία για να δουλέψει και υποδεικνύει όλες τις πρόσθετες πληροφορίες που ο μεταφραστής του προφίλ επιθυμεί να μοιραστεί με τους αποδέκτες.

contactInformation: παρέχει ένα μηχανισμό που αναφέρεται στους ανθρώπους ή στους υπευθύνους για την υπηρεσία. Το εύρος αυτής της ιδιότητας είναι ακαθόριστο στο OWL-S, αλλά μπορεί να περιοριστεί σε κάποια άλλη οντολογία όπως η FOAF, η VCard ή η ActorClass που παρέχεται σε προηγούμενες εκδόσεις του OWL-S.

3. Περιγραφή Λειτουργικότητας

Ένας σημαντικός παράγοντας του προφίλ, είναι ο καθορισμός του είδους της λειτουργικότητας που παρέχει η υπηρεσία και ο καθορισμός των όρων που πρέπει να ικανοποιηθούν για ένα επιτυχές αποτέλεσμα. Επιπλέον το προφίλ προσδιορίζει ποιες προϋποθέσεις προκύπτουν από την υπηρεσία, συμπεριλαμβάνοντας τα αναμενόμενα και τα μη αναμενόμενα αποτελέσματα από την λειτουργία της υπηρεσίας. Το προφίλ του OWL-S παρουσιάζει δύο πλευρές της λειτουργικότητας μιας υπηρεσίας, το μετασχηματισμό των πληροφοριών (που αναπαρίσταται από τα δεδομένα που εισάγονται και εξάγονται) και την αλλαγή κατάστασης που παράγεται από την εκτέλεση της υπηρεσίας (που αναπαρίσταται από τις προϋποθέσεις και τα αποτελέσματα). Παραδείγματος χάριν, για να γίνει μια πώληση, μια υπηρεσία πώλησης βιβλίων απαιτεί ως δεδομένο εισαγωγής τον αριθμό της πιστωτικής κάρτας και την ημερομηνία λήξης, αλλά και τη προϋπόθεση ότι η κάρτα όντως υπάρχει. Το αποτέλεσμα της πώλησης είναι μια απόδειξη η οποία βεβαιώνει τη σωστή εκτέλεση της συναλλαγής, και θα έχει ως αποτέλεσμα την αλλαγή ιδιοκτησίας και τη φυσική μεταφορά του βιβλίου από την αποθήκη στην διεύθυνση του αγοραστή.

Το προφίλ της οντολογίας δεν παρέχει ένα σχήμα που να περιγράφει τα στιγμιότυπα του IOPE. Ωστόσο, τέτοιο σχήμα υπάρχει στην οντολογία Διαδικασίας. Ιδανικά, οραματιζόμαστε ότι τα δημοσιευμένα IOPEs από το προφίλ, είναι ένα υποσύνολο αυτών που έχουν δημοσιοποιηθεί από τη διαδικασία. Για αυτό το λόγο το κομμάτι της περιγραφής Διαδικασία θα δημιουργήσει όλα τα IOPE στιγμιότυπα και το

στιγμιότυπο του Προφίλ μπορεί απλά να υποδεικνύει αυτά τα στιγμιότυπα. Στην περίπτωση αυτή, ένα μοναδικό στιγμιότυπο έχει δημιουργηθεί για κάθε IOPE, σε αντίθεση με τις προηγούμενες εκδόσεις του OWL-S, όταν για μια IOPE δημιουργούνταν ένα στιγμιότυπο στο τμήμα της OWL-S περιγραφής του προφίλ και της διαδικασίας. Παρόλα αυτά αν τα IOPEs του προφίλ είναι διαφορετικά από εκείνα της διαδικασίας, το προφίλ μπορεί ακόμα να δημιουργήσει τα δικά του IOPE στιγμιότυπα χρησιμοποιώντας το σχήμα που προσφέρεται στην οντολογία της διαδικασίας.

Η οντολογία του προφίλ ορίζει τις παρακάτω ιδιότητες της τάξης του προφίλ για να υποδεικνύουν τα IOPEs:

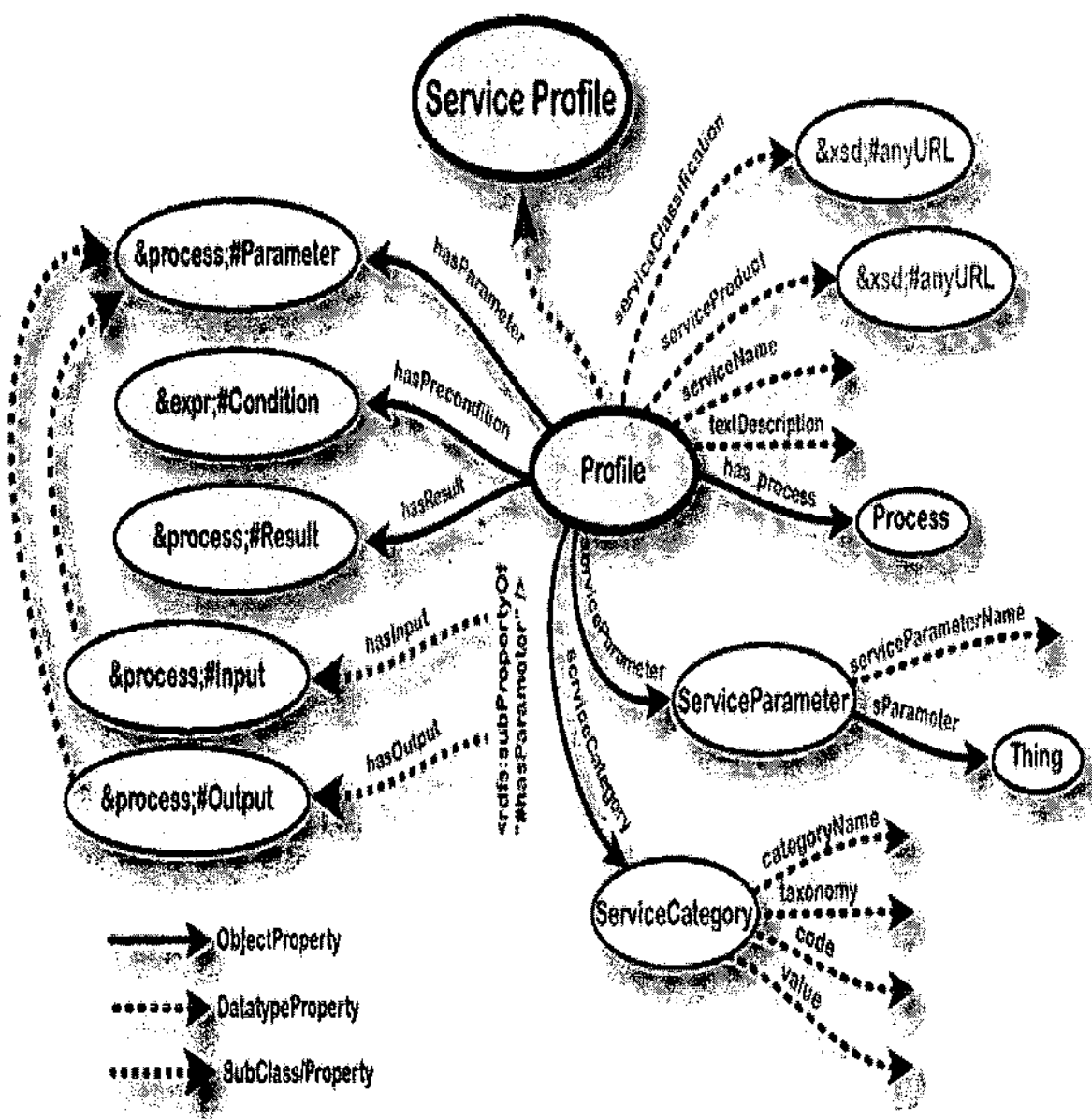
hasParameter: κατατάσσεται στο στιγμιότυπο της Parameter (Παραμέτρου) της οντολογίας διαδικασίας. Θα πρέπει να σημειώσουμε ότι η εισαγωγή δεδομένων και τα αποτελέσματα που εμπεριέχονται στον μετασχηματισμό πληροφορίας, είναι διαφορετικά από τις προϋποθέσεις και τα αποτελέσματα. Συνεπώς δεν περιμένουμε να μετατραπεί σε στιγμιότυπα η συγκεκριμένη τάξη. Ο ρόλος του είναι να καθιστά σαφή τη γνώση της κυριότητας.

hasInput: κατατάσσεται στα στιγμιότυπα των δεδομένων που εισάγονται όπως αυτά ορίζονται στην οντολογία διαδικασίας.

hasOutput: κατατάσσεται στα στιγμιότυπα του τύπου των αποτελεσμάτων στην οντολογία διαδικασίας

hasPrecondition: ορίζει μια από τις προϋποθέσεις της υπηρεσίας και αναφέρεται στο στιγμιότυπο Precondition, που ορίζεται σύμφωνα με το σχήμα στην οντολογία διαδικασίας.

hasResult: ορίζει ένα από τα αποτελέσματα της υπηρεσίας όπως ορίζονται από τη τάξη Result στην οντολογία Process. Επισημαίνει κάτω από ποιες συνθήκες παράγονται τα αποτελέσματα. Επιπλέον η τάξη Result ορίζει ποιες αλλαγές κυριότητας δημιουργούνται κατά τη διάρκεια εκτέλεσης της υπηρεσίας.



Σχήμα 8 - Επιλεγμένες Τάξεις και Ιδιότητες του Προφίλ

4. Χαρακτηριστικά του Προφίλ

Στο προηγούμενο κομμάτι αναφέραμε τη λειτουργική περιγραφή των υπηρεσιών, υπάρχουν όμως και άλλες πλευρές των υπηρεσιών που θα πρέπει να γνωρίζουν οι χρήστες. Αυτά τα επιπρόσθετα χαρακτηριστικά εμπεριέχουν τις εγγυήσεις ποιότητας που παρέχονται από την υπηρεσία, πιθανή κατηγοριοποίηση

της υπηρεσίας και επιπρόσθετες παραμέτρους που μπορεί να θέλει η υπηρεσία να ορίσει.

serviceParameter: είναι μια επεκτάσιμη λίστα ιδιοτήτων που μπορεί να συνοδεύει τη περιγραφή του προφίλ. Η τιμή της ιδιότητας αυτής είναι ένα στιγμιότυπο της τάξης Service Parameter.

serviceCategory: αναφέρεται στην είσοδο σε μια οντολογία ή σε μια ταξινόμια υπηρεσιών. Η τιμή της ιδιότητας αυτής είναι ένα στιγμιότυπο των παραμέτρων της τάξης ServiceCategory.

3.1.5.3 Παράμετροι της Υπηρεσίας

serviceParameterName: είναι το όνομα μιας πραγματικής παραμέτρου, ή πιθανόν το URI της παραμέτρου επεξεργασίας.

sParameter: στοχεύει στην τιμή της παραμέτρου μέσα σε κάποια οντολογία OWL.

3.1.5.4 Κατηγορία Υπηρεσίας

Το Service Category περιγράφει κατηγορίες της υπηρεσίας στις βάσεις κάποιας κατηγοριοποίησης που μπορεί να είναι έξω από το OWL-S και πιθανότατα έξω από το OWL. Στην τελευταία περίπτωση μπορεί να απαιτούνται κάποιοι εξειδικευμένοι μηχανισμοί αιτιολόγησης.

categoryName: είναι το όνομα της αληθινής κατηγορίας η οποία θα μπορούσε να είναι πραγματικότητα, ή πιθανόν το URI της παραμέτρου επεξεργασίας (μια ιδιότητα).

taxonomy: αποθηκεύει την αναφορά στο σχήμα ταξινόμησης. Μπορεί να είναι το URI της ταξινόμησης ή το URL όπου η ταξινόμηση εδρεύει ή το όνομα της ταξινόμησης ή οτιδήποτε άλλο.

value: υποδεικνύει τη τιμή μιας ορισμένης ταξινόμησης. Μπορεί να υπάρχουν περισσότερες από μια τιμές για κάθε ταξινόμηση, οπότε δεν προστίθεται περιορισμός σε αυτή τη περίπτωση.

code: σε κάθε τύπο υπηρεσιών αποθηκεύει τον κωδικό που σχετίζεται με την ταξινόμηση.

3.1.5.5 Κατηγοριοποιώντας τον Τύπο της Υπηρεσίας και το Προϊόν

Οι δυο ιδιότητες, η **serviceClassification** και η **serviceProduct**, χρησιμοποιούνται για να ορίζουν τον τύπο της υπηρεσίας που παρέχεται και τα προϊόντα που διαχειρίζεται η υπηρεσία. Η τιμή των δύο ιδιοτήτων είναι στιγμιότυπα των τάξεων που καθορίζονται στις OWL οντολογίες των υπηρεσιών και των προϊόντων. Οι ιδιότητες **serviceClassification** και **serviceProduct** είναι παρόμοιες με το **serviceCategory** που περιγράψαμε παραπάνω, αλλά διαφέρουν στο ότι οι τιμές των ιδιοτήτων είναι στιγμιότυπα του OWL.

serviceClassification: ορίζει μια αντιστοίχιση από ένα προφίλ σε μια οντολογία OWL του προϊόντος, όπως μία OWL προδιαγραφή του NAICS.

serviceProduct: ορίζει μια αντιστοίχιση από ένα προφίλ σε μια οντολογία OWL του προϊόντος, όπως την OWL προδιαγραφή του UNSPSC.

3.1.6 Μοντελοποιώντας Υπηρεσίες ως Διαδικασίες

Για να δώσουμε μια λεπτομερή περιγραφή για το πως θα γίνει η αλληλεπίδραση με μια υπηρεσία, μπορούμε να την θεωρήσουμε ως επεξεργασία. Συγκεκριμένα, το OWL-S 1.1 καθορίζει μια υποκατηγορία του **ServiceModel**, **Process**, η οποία βασίζεται σε μια καλά δομημένη εργασία σε διάφορους τομείς, εμπιρεύοντας εργασίες στην τεχνητή νοημοσύνη, στην προτυποποίηση των γλωσσών σχεδιασμού, των γλωσσών προγραμματισμού και στα κατανεμημένα συστήματα, ανερχόμενα πρότυπα στη μοντελοποίηση διαδικασιών και στη τεχνολογία της ροής εργασίας όπως το NIST's **Process Specification Language (PSL)** [30] και την **Workflow Management Coalition** προσπάθεια, εργασίες πάνω στη σήμανση [31], εργασία στην τεχνητή νοημοσύνη πάνω στη μοντελοποίηση σύνθετων ενεργειών και εργασίες πάνω στις γλώσσες επικοινωνίας των μεσαζόντων.

Είναι σημαντικό να κατανοήσουμε ότι η επεξεργασία δεν είναι πρόγραμμα που πρέπει να εκτελεστεί, είναι μία προδιαγραφή των τρόπων που μπορεί ο πελάτης να αλληλεπιδράσει με μια υπηρεσία. Μια ατομική επεξεργασία είναι μια περιγραφή της υπηρεσίας που περιμένει ένα, πιθανόν σύνθετο, μήνυμα και επιστρέφει ένα, πιθανόν σύνθετο, μήνυμα ως απάντηση. Ως σύνθετη επεξεργασία ορίζουμε εκείνη που

υποστηρίζει μια κατάσταση στην οποία σε κάθε μήνυμα που στέλνει ο πελάτης, το προάγει μέσα από τη διαδικασία.

Η επεξεργασία μπορεί να έχει δύο είδη σκοπού. Πρώτον, μπορεί να παράγει και να επιστρέφει κάποιες καινούριες πληροφορίες βασισμένη στις δοθέντες πληροφορίες. Η παραγωγή πληροφοριών περιγράφεται από τα δεδομένα και τα αποτελέσματα της επεξεργασίας. Δεύτερον, μπορεί να παράγει μια αλλαγή στον κόσμο. Αυτή η αλλαγή περιγράφεται από τις προϋποθέσεις και τις επιδράσεις της επεξεργασίας.

Η επεξεργασία μπορεί να έχει οποιοδήποτε αριθμό δεδομένων (ακόμα και 0) παρουσιάζοντας την πληροφορία, η οποία, κάτω από ορισμένες συνθήκες, απαιτείται για την απόδοση της επεξεργασίας. Μπορεί να έχει κάθε αριθμό αποτελεσμάτων, η πληροφορία όπου η επεξεργασία παρέχει στον αιτούντα. Μπορεί να έχει κάθε αριθμό προϋποθέσεων οι οποίες πρέπει να κρατούνται σε σειρά με σκοπό η επεξεργασία να γίνεται επιτυχώς. Τελικά η επεξεργασία μπορεί να έχει κάθε αριθμό επιδράσεων.

Πριν αναλύσουμε τη λειτουργία των διαδικασιών είναι απαραίτητο να εξηγήσουμε πως λειτουργούν τα δεδομένα που εισάγονται, τα δεδομένα που εξάγονται, οι προϋποθέσεις και τα αποτελέσματα (IOPE).

3.1.6.1 Παράμετροι και Εκφράσεις

Τα δεδομένα και τα αποτελέσματα είναι υποκατηγορίες μιας γενικής τάξης που καλείται παράμετρος (Parameter). Είναι βολικό να αναγνωρίζουμε τις παραμέτρους ως μεταβλητές, έτσι καλούνται στην SWRL, τη γλώσσα που εκφράζει τους κανόνες του OWL.

```
<owl:Class rdf:about="#Parameter">  
  <rdfs:subClassOf rdf:resource="#swrl;#Variable"/>  
</owl:Class>
```

Κάθε παράμετρος έχει ένα καθορισμένο τύπο χρησιμοποιώντας ένα URI. Δεν είναι αυτή η OWL τάξη στην οποία ανήκει η παράμετρος, αλλά ένας προσδιορισμός της τάξης που υπολογίζει την παράμετρο στην οποία ανήκει.

```

<owl:DatatypeProperty rdf:ID="parameterType">
  <rdfs:domain rdf:resource="#Parameter"/>
  <rdfs:range rdf:resource="&xsd:anyURI"/>
</owl:DatatypeProperty>

```

```

<owl:Class rdf:ID="Parameter">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#parameterType" />
      <owl:minCardinality rdf:datatype="&xsd;#nonNegativeInteger">
        1</owl:minCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

```

Τα δεδομένα και τα αποτελέσματα είναι υποκατηγορίες της παραμέτρου:

```

<owl:Class rdf:ID="Input">
  <rdfs:subClassOf rdf:resource="#Parameter"/>
</owl:Class>

```

```

<owl:Class rdf:ID="Output">
  <rdfs:subClassOf rdf:resource="#Parameter"/>
</owl:Class>

```

Δύο άλλες υποκατηγορίες τάξης των Parameter, Local και ResultVar περιγράφονται παρακάτω.

Ο τρόπος που γίνεται η μοντελοποίηση μεταβλητών για την Owl Rules, μπορεί να είναι παραπλανητική. Η RDF δεν έχει καμία αντίληψη του «αντικειμένου» μιας μεταβλητής, διότι η RDF δεν είναι τίποτα άλλο από μία συνάθροιση τριάδων. Μια μεταβλητή ονομάζεται με ένα URI και έτσι έχει μία παγκόσμια έκταση, ή πιο συγκεκριμένα, καμία αντίληψη έκτασης. Παρόλο που έχουμε έλλειψη δομής, συχνά χρησιμοποιούμε την RDF για να κωδικοποιήσουμε ιεραρχικές οντότητες όπως

φόρμουλες και δομές ελέγχου. Περιβάλλοντας τις αναφορές μεταβλητών εσωτερικά μπορούμε να επέμβουμε και να επιβάλλουμε τους δικούς μας κανόνες επίδρασης.

Η επεξεργασία δεν θα εκτελεστεί ομαλά αν δεν είναι αληθινές οι προϋποθέσεις, αν και όταν εκτελεστεί έχει ποικίλες επιδράσεις. Παραδείγματος χάριν ένας πράκτορας μπορεί να παραγγείλει 1000bolts από μια υπηρεσία δικτύου μόνο αν μπορέσει να πείσει την υπηρεσία να δεχτεί την υπόσχεση του ότι θα πληρώσει. Μια επίδραση της παραγγελίας είναι η μεταφορά της κυριότητας των bolts από την υπηρεσία στο πράκτορα.

Οι προϋποθέσεις και οι επιδράσεις παρουσιάζονται ως λογικές φόρμουλες. Το να έχουμε λογικές φόρμουλες στο RDF δεν ήταν εύκολο, άλλα είναι πλέον ξεκάθαρο το πώς πρέπει να προχωρήσουμε. Υπάρχουν αρκετές πιθανές προσεγγίσεις ανάλογα με το πόσο κοντά στο RDF/OWL θέλει κάποιος να παραμείνει. Συνήθως, η ύπαρξη πολλών επιλογών για μια τόσο κρίσιμη εργασία, δεν είναι καλή ιδέα, αλλά σε αυτήν τη περίπτωση οι περισσότερες διαφορές είναι επιφανειακές.

Η βασική ιδέα της προσέγγισης μας είναι να αντιμετωπίζουμε τις εκφράσεις, είτε ως literal είτε ως string literal είτε ως XML literal. Η τελευταία περίπτωση χρησιμοποιείται για γλώσσες των οποίων η κωδικοποίηση είναι σε XML, όπως στην SWRL[32] ή στην RDF. Η προηγούμενη περίπτωση αναφέρεται σε άλλες γλώσσες όπως η KIF[33] και η PDDL. Η οντολογία [<http://www.daml.org/services/owl-s/1.1/generic/Expression.owl>] καθορίζει τα Expressions και τις ιδιότητές τους.

```
<owl:Class rdf:ID="Expression">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#expressionLanguage"/>
      <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">
        1</owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#expressionBody"/>
      <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">
        1</owl:cardinality>
```



```
</owl:Restriction>
</rdfs:subClassOf>
</owl:Class>
```

Υπομνηματίζουμε τις εκφράσεις με τη γλώσσα μέσω της οποίας εκφράζονται.

```
<owl:ObjectProperty rdf:ID="&expr;#expressionLanguage">
  <rdfs:domain rdf:resource="&expr;#Expression"/>
  <rdfs:range rdf:resource="&expr;#LogicLanguage"/>
</owl:ObjectProperty>
```

Η ιδιότητα expressionBody δίνει την πραγματική έκφραση:

```
<owl:DatatypeProperty rdf:ID="expressionBody">
  <rdfs:domain rdf:resource="#Expression"/>
</owl:DatatypeProperty>
```

Ως παράδειγμα μπορούμε να δηλώσουμε ότι για να στείλουμε τον αριθμό μιας συγκεκριμένης πιστωτικής κάρτας σε ένα πράκτορα στο δίκτυο, θα πρέπει να γνωρίζουμε ποιος είναι ο αριθμός.

```
Description rdf:about="#process2">
  <hasPrecondition>
    <expr:KIF-Expression>
      <expr:expressionBody>
        (!agnt:know_val_is
          (!ecom:credit_card_num ?cc)
          ?num)
      </expr:expressionBody>
    </expr:KIF-Expression>
  </hasPrecondition>
</Description>
```

Σε περιπτώσεις που χρησιμοποιείται η κωδικοποίηση XML, θα δηλώνουμε την έκφραση ως XML. Εδώ είναι το ίδιο παράδειγμα χρησιμοποιώντας DRS σαν γλώσσας έκφρασης.

```
<Description rdf:about="#process2">
  <hasPrecondition>
    <Expression expressionLanguage="#drs:#DRS">
      <process:expressionBody>
        <drs:Atomic_formula>
          <rdf:predicate rdf:resource="#&agnt;#Know_val_is"/>
          <rdf:subject>
            <drs:Functional_term>
              <drs:function rdf:resource="#&ecom;credit_card_num"/>
              <drs:term_args rdf:parseType="Collection">
                <swrl:Variable rdf:resource="#CC"/>
              </drs:term_args>
            </drs:Functional_term>
          </rdf:subject>
          <rdf:object rdf:resource="#Num"/>
        </drs:Atomic_formula>
      </process:expressionBody>
    </Expression>
  </hasPrecondition>
</Description>
```

Οι αναφορές στα #cc και #Num στο παράδειγμα της DRS είναι για τις παραμέτρους οι οποίες είναι ίδιες με αυτές που αναγράφονται ως ?cc και ?Num στο παράδειγμα της KIF. Ακόμα δεν έχουμε βρει ένα μηχανισμό για να δηλώσουμε το πεδίο δράσης των μεταβλητών και πως αποκτούν τιμές. Πρόχειρα μιλώντας, οι μεταβλητές και οι παράμετροι αποκτούν το πεδίο δράσης της διαδικασίας στην οποία χρησιμοποιούνται, θα αναφερθούμε αργότερα πάνω σε αυτό, λεπτομερέστερα. Στο παράδειγμα, το #cc είναι μια παράμετρος εισόδου στην επεξεργασία, η οποία παρέχεται από τον πελάτη, αλλά #Num θεωρείται ότι είναι δεδομένο στην επεξεργασία της αιτιολογίας γύρω από αυτήν την προϋπόθεση, επιβεβαιώνοντας ότι

η επεξεργασία είναι εφικτή, απαιτεί επανάκτηση των 16 ψηφιακών αριθμών των πιστωτικών καρτών, το οποίο σχετίζεται με τη μεταβλητή #Num. Ονομάζουμε τέτοιες παραμέτρους ως local parameter.

```
<owl:Class rdf:ID="Local">  
  <rdfs:subClassOf rdf:resource="#Parameter"/>  
</owl:Class>
```

Οι 3 τύποι παραμέτρου αποσυντίθενται:

```
<rdf:Description rdf:about="#Input">  
  <owl:disjointWith rdf:resource="#Output"/>  
  <owl:disjointWith rdf:resource="#Local"/>  
</rdf:Description>
```

```
<rdf:Description rdf:about="#Output">  
  <owl:disjointWith rdf:resource="#Local"/>  
</rdf:Description>
```

Φυσικά το να σημάνουμε τα bits της RDF ως «literals» σημαίνει ότι ο RDF parser θα έπρεπε να τα αγνοεί. Η έξυπνη κίνηση σε αυτή τη περίπτωση είναι να έχουμε τον OWL-S parser να αποσπά το υλικό που αγνοείται και να το μεταφράζει κατάλληλα για το περιεχόμενό του, συμπεριφέροντας του ως ένα συνηθισμένο RDF μετά από τις μετατροπές όπως την αντικατάσταση της εμφάνισης των μεταβλητών με τις τιμές τους. Στο παραπάνω παράδειγμα η ύπαρξη του #Num και #cc μεταφράζονται ως τιμές αυτών των μεταβλητών και όχι οι ίδιες οι μεταβλητές. Στο παράδειγμα KIF οι εκφράσεις ?Num και ?cc θα πρέπει να μεταφραστούν παρόμοια.

Υπάρχουν 2 ειδικές περιπτώσεις του Expression: η Condition και η Effect. Επειδή έχουν υλοποιηθεί ως literals, δεν υπάρχει τρόπος να διακρίνουμε τη διαφορά, αλλά είναι μια χρήσιμη διάκριση για τον μελετητή της οντολογίας.

```
<owl:Class rdf:ID="Condition">  
  <owl:subClassOf rdf:resource="#&expr;#Expression"/>  
</owl:Class>
```

```

<owl:Class rdf:ID="Effect">
  <owl:subClassOf rdf:resource="&expr;#Expression"/>
</owl:Class>

```

3.1.6.2 Παράμετροι Επεξεργασίας και Αποτελέσματα

Συνδέουμε τις επεξεργασίες με τα «IOPE» τους, χρησιμοποιώντας τις ιδιότητες που παρουσιάζονται στον πίνακα.

<i>Property</i>	<i>Range</i>	<i>Kind</i>
hasParticipant	Participant	<i>Thing</i>
hasInput	Input	<i>Parameter</i>
hasOutput	Output	<i>Parameter</i>
hasLocal	Local	<i>Parameter</i>
hasPrecondition	Condition	<i>Expression</i>
hasResult	Result	(see below)

Πίνακας 1

Όπως αναφέραμε παραπάνω οι συνδέσεις μιας επεξεργασίας με τις παραμέτρους της, δίνουν σε αυτές μία εμβέλεια. Συμμετέχοντες, δεδομένα, αποτελέσματα και τοπικές παράμετροι έχουν σαν σκοπό τη συνολική διαδικασία στην οποία συμμετέχουν. Παρουσιάζουμε παρακάτω τα result vars που έχουν πιο περιορισμένη εμβέλεια. Παρακάτω θα αναλύσουμε τα περιεχόμενα του πίνακα.

- **Συμμετέχοντες (Participants)**

Μια επεξεργασία εμπεριέχει 2 ή περισσότερους μεσαζόντες. Ο ένας είναι ο πελάτης (TheClient), ο πράκτορας μέσω του οποίου η επεξεργασία περιγράφεται. Ο άλλος είναι ο προμηθευτής (TheServer), το βασικό στοιχείο της υπηρεσίας με το οποίο ο πελάτης συναλλάσσεται. Αν υπάρχουν άλλοι, καταχωρούνται χρησιμοποιώντας την ιδιότητα hasParticipant.

```
<owl:ObjectProperty rdf:ID="hasParticipant">
  <rdfs:domain rdf:resource="#Process"/>
</owl:ObjectProperty>
```

```
<owl:ObjectProperty rdf:ID="hasClient">
  <rdfs:subPropertyOf rdf:resource="#hasParticipant"/>
</owl:ObjectProperty>
```

```
<process:Parameter rdf:ID="TheClient">
<process:Parameter rdf:ID="TheServer">
```

- **Δεδομένα και Αποτελέσματα**

Τα δεδομένα και τα αποτελέσματα καθορίζουν τη μετατροπή των δεδομένων που δημιουργείται από την επεξεργασία. Τα δεδομένα καθορίζουν την πληροφορία, που η επεξεργασία απαιτεί για την εκτέλεση της. Για ατομικές επεξεργασίες, η πληροφορία πρέπει να έρχεται από τον πελάτη. Για μια σύνθετη επεξεργασία κάποια δεδομένα έρχονται απευθείας από τον πελάτη, και άλλα από κάποια προηγούμενα στάδια της επεξεργασίας.

Αναφέραμε παραπάνω, ότι μια ατομική επεξεργασία αντιστοιχεί σε μια υπηρεσία του ενός βήματος που περιμένει ένα μήνυμα, έτσι μπορεί να φαίνεται αντιφατικό για μια ατομική επεξεργασία να έχει πολλαπλά δεδομένα εισαγωγής. Η αντιφατικότητα αυτή εξηγείται με τη διάκριση ανάμεσα στα δεδομένα και στο μήνυμα που στέλνεται στην επεξεργασία. Υπάρχει μόνο ένα μήνυμα, αλλά μπορεί να συγκεντρώσει όσα εισαχθέντα δεδομένα απαιτούνται. Η συγκέντρωση δεδομένων καθορίζεται από την υποστήριξη του μοντέλου επεξεργασίας. Παρομοίως, τα αποτελέσματα παράγονται από την επίκληση μιας ατομικής επεξεργασίας και ρέουν πίσω στον πελάτη ως ένα μοναδικό μήνυμα του οποίου η δομή καθορίζεται από την υποστήριξη.

Το ακόλουθο παράδειγμα δείχνει τον ορισμό του hasParameter και τις υποιδιότητες hasInput, hasOutput, και hasLocal.

```
<owl:ObjectProperty rdf:ID="hasParameter">
```

```
<rdfs:domain rdf:resource="#Process"/>
<rdfs:range rdf:resource="#Parameter"/>
</owl:ObjectProperty>
```

```
<owl:ObjectProperty rdf:ID="hasInput">
  <rdfs:subPropertyOf rdf:resource="#hasParameter"/>
  <rdfs:range rdf:resource="#Input"/>
</owl:ObjectProperty>
```

```
<owl:ObjectProperty rdf:ID="hasOutput">
  <rdfs:subPropertyOf rdf:resource="#hasParameter"/>
  <rdfs:range rdf:resource="#Output"/>
</owl:ObjectProperty>
```

```
<owl:ObjectProperty rdf:ID="hasLocal">
  <rdfs:subPropertyOf rdf:resource="#hasParameter"/>
  <rdfs:range rdf:resource="#Local"/>
</owl:ObjectProperty>
```

- **Προϋποθέσεις Και Αποτελέσματα**

Αν μια επεξεργασία έχει προϋποθέσεις, τότε η επεξεργασία δεν μπορεί να εφαρμοστεί επιτυχώς εάν η προϋπόθεση δεν είναι αληθής.

```
owl:ObjectProperty rdf:ID="hasPrecondition">
  <rdfs:domain rdf:resource="#Process"/>
  <rdfs:range rdf:resource="&expr;#Condition"/>
</owl:ObjectProperty>
```

Πρέπει να μπορούμε να διακρίνουμε ανάμεσα σε μια συνθήκη που είναι αληθής και σε διάφορες άλλες ιδιότητες, όπως το να θεωρείται αληθής, το να αναπαρίσταται σε μια βάση δεδομένων ως αληθής κ.α. Στο OWL-S, αν η προϋπόθεση μιας διαδικασίας είναι ψευδή, οι συνέπειες της εκτέλεσης ή της αρχής μιας επεξεργασίας είναι ακαθόριστες.

Η απόδοση μιας επεξεργασίας μπορεί να έχει ως αποτέλεσμα αλλαγές στη κατάσταση του κόσμου (επιδράσεις) και την απόκτηση πληροφορίας από τον πράκτορα πελάτη. Ωστόσο δεν συνδέουμε την επεξεργασία άμεσα με τις επιδράσεις και τα αποτελέσματα της, γιατί αυτοί που μοντελοποιούν τις διαδικασίες συχνά επιθυμούν να μοντελοποιήσουν την εξάρτηση αυτών στο περιεχόμενο τους. Παραδείγματος χάριν αν μια επεξεργασία περιέχει το βήμα της αγοράς ενός αγαθού, υπάρχουν 2 δυνατά συμπεράσματα: είτε η αγορά πετυχαίνει είτε αποτυγχάνει. Στην πρώτη περίπτωση η επίδραση είναι ότι η ιδιοκτησία μεταφέρεται και το αποτέλεσμα είναι ένας επιβεβαιωτικός αριθμός. Στην τελευταία περίπτωση δεν υπάρχει καμία επίδραση και το αποτέλεσμα είναι ένα μήνυμα αποτυχίας.

Χρησιμοποιούμε τον όρο *result* για να αναφερθούμε στο ζευγάρι output και effect.

```
<owl:Class rdf:ID="Result">
  <rdfs:label>Result</rdfs:label>
</owl:Class>

<owl:ObjectProperty rdf:ID="hasResult">
  <rdfs:label>hasResult</rdfs:label>
  <rdfs:domain rdf:resource="#Process"/>
  <rdfs:range rdf:resource="#Result"/>
</owl:ObjectProperty>
```

- **Καθορίζοντας Τα Αποτελέσματα Και Τις Επιδράσεις**

Έχοντας δηλώσει ένα αποτέλεσμα, ένα μοντέλο επεξεργασίας μπορεί μετά να το περιγράψει βάσει 4 ιδιοτήτων:

```
<owl:ObjectProperty rdf:ID="inCondition">
  <rdfs:label>inCondition</rdfs:label>
  <rdfs:domain rdf:resource="#Result"/>
  <rdfs:range rdf:resource="#&expr;#Condition"/>
</owl:ObjectProperty>
```

```

<owl:ObjectProperty rdf:ID="hasResultVar">
  <rdfs:label>hasResultVar</rdfs:label>
  <rdfs:domain rdf:resource="#Result"/>
  <rdfs:range rdf:resource="#ResultVar"/>
</owl:ObjectProperty>

```

```

<owl:ObjectProperty rdf:ID="withOutput">
  <rdfs:label>withOutput</rdfs:label>
  <rdfs:domain rdf:resource="#Result"/>
  <rdfs:range rdf:resource="#OutputBinding"/>
</owl:ObjectProperty>

```

```

<owl:ObjectProperty rdf:ID="hasEffect">
  <rdfs:label>hasEffect</rdfs:label>
  <rdfs:domain rdf:resource="#Result"/>
  <rdfs:range rdf:resource="#Expression"/>
</owl:ObjectProperty>

```

Η ιδιότητα `inCondition` καθορίζει τις συνθήκες κάτω από τις οποίες αυτό το αποτέλεσμα (και όχι κάποιο άλλο) συμβαίνει. Οι ιδιότητες `withOutput` και `hasEffect` δηλώνουν τι ακολουθεί όταν η συνθήκη είναι ορθή. Η ιδιότητα `hasResultVar` δηλώνει τις μεταβλητές που περικλείονται στην `inCondition`. Αυτές οι μεταβλητές καλούνται `ResultVars`, είναι ανάλογες με τις `Locals` και εξυπηρετούν ένα παρόμοιο σκοπό. Όπου οι `Locals` είναι μεταβλητές που περικλείονται στις προϋποθέσεις και μετά χρησιμοποιούνται στη περιγραφή των αποτελεσμάτων και των επιδράσεων που σχετίζονται με αυτή την υπόθεση. Παραδείγματος χάριν αν μια διαδικασία έπρεπε να επικυρώσει μια πιστωτική κάρτα, τότε η `ResultVar CardAccepted` θα περιείχε το αποτέλεσμα αυτής της επεξεργασίας, η οποία θα μπορούσε έπειτα να επιστραφεί ως αποτέλεσμα.

```

<owl:Class rdf:about="ResultVar">
  <rdfs:subClassOf rdf:resource="#Parameter"/>
  <owl:disjointWith rdf:resource="#Input"/>
  <owl:disjointWith rdf:resource="#Output"/>

```



```

<owl:disjointWith rdf:resource="#Local"/>
</owl:Class>

```

Άλλο ένα τυπικό παράδειγμα είναι μια επεξεργασία που χρεώνει μια πιστωτική κάρτα. Η χρέωση γίνεται μόνο όταν δεν έχει διαγραφεί η κάρτα. Αν έχει διαγραφεί το μόνο αποτέλεσμα είναι μια αναφορά αποτυχίας. Έτσι η περιγραφή της επεξεργασίας πρέπει να περιέχει την περιγραφή δύο αποτελεσμάτων (two Results), πιθανών σε αυτήν την φόρμα:

```

<process:AtomicProcess rdf:ID="Purchase">
  <process:hasInput>
    <process:Input rdf:ID="ObjectPurchased"/>
  </process:hasInput>
  <process:hasInput>
    <process:Input rdf:ID="PurchaseAmt"/>
  </process:hasInput>
  <process:hasInput>
    <process:Input rdf:ID="CreditCard"/>
  </process:hasInput>
  <process:hasOutput>
    <process:Output rdf:ID="ConfirmationNum"/>
  </process:hasOutput>
  <process:hasResult>
    <process:Result>
      <process:hasResultVar>
        <process:ResultVar rdf:ID="CreditLimH">
          <process:parameterType rdf:resource="&ecom;#Dollars"/>
        </process:ResultVar>
      </process:hasResultVar>
      <process:inCondition>
        <expr:KIF-Condition>
          <expr:expressionBody>
            (and (current-value (credit-limit ?CreditCard)
              ?CreditLimH)

```

```

        (>= ?CreditLimH ?purchaseAmt))
    </expr:expressionBody>
</expr:KIF-Condition>
</process:inCondition>
<process:withOutput>
  <process:OutputBinding>
    <process:toParam rdf:resource="#ConfirmationNum"/>
    <process:valueFunction rdf:parseType="Literal">
      <cc:ConfirmationNum xsd:datatype="&xsd;#string"/>
    </process:valueFunction>
  </process:OutputBinding>
</process:withOutput>
<process:hasEffect>
  <expr:KIF-Condition>
    <expr:expressionBody>
      (and (confirmed (purchase ?purchaseAmt) ?ConfirmationNum)
        (own ?objectPurchased)
        (decrease (credit-limit ?CreditCard)
          ?purchaseAmt))
    </expr:expressionBody>
  </expr:KIF-Condition>
</process:hasEffect>
</process:Result>
<process:Result>
  <process:hasResultVar>
    <process:ResultVar rdf:ID="CreditLimL">
      <process:parameterType rdf:resource="&ecom;#Dollars"/>
    </process:ResultVar>
  </process:hasResultVar>
  <process:inCondition>
    <expr:KIF-Condition>
      <expr:expressionBody>
        (and (current-value (credit-limit ?CreditCard)
          ?CreditLimL)

```

```

        (< ?CreditLimL ?purchaseAmt))
    </expr:expressionBody>
</expr:KIF-Condition>
</process:inCondition>
<process:withOutput rdf:resource="&ecom;failureNotice"/>
    <process:OutputBinding>
        <process:toParam rdf:resource="#ConfirmationNum"/>
        <process:valueData rdf:parseType="Literal">
            <drs:Literal>
                <drs:litdefn xsd:datatype="&xsd;#string"
                    >00000000</drs:litdefn>
            </drs:Literal>
        </process:valueData>
    </process:OutputBinding>
</process:withOutput>
</process:Result>
</process:hasResult>
</process:AtomicProcess>

```

Σαν αποτέλεσμα αυτής της επεξεργασίας, η πιστωτική κάρτα χρεώνεται και τα χρήματα του λογαριασμού μειώνονται. Θα πρέπει να σημειώσουμε ότι υπάρχει μια θεμελιώδης διαφορά ανάμεσα στις επιδράσεις και τα αποτελέσματα. Οι επιδράσεις περιγράφουν τις συνθήκες στον κόσμο, ενώ τα αποτελέσματα περιγράφουν πληροφορίες. Σε μια πιο ρεαλιστική εκδοχή αυτού του παραδείγματος, η υπηρεσία μπορεί να στείλει μια ειδοποίηση ότι χρέωσε το λογαριασμό της πιστωτικής κάρτας. Αυτό το αποτέλεσμα είναι απλά ένα δεδομένο ενός τύπου ή κάποιου άλλου. Η επίδραση περιγράφει το πραγματικό γεγονός, ότι το αποτέλεσμα είναι μέρος αυτής της περιγραφής, ότι το σύνολο των χρημάτων στο λογαριασμό της πιστωτικής κάρτας έχουν μειωθεί και ότι στον πελάτη ανήκει το προϊόν που σκόπευε να αγοράσει. Τέλος, υπάρχει ένα χαρακτηριστικό περιγραφής αποτελεσμάτων, που καλείται result form και δεν σχετίζεται με μια μεταβλητή αλλά με το αποτέλεσμα.

```

<owl:DatatypeProperty rdf:ID="resultForm">
    <rdfs:label>resultForm</rdfs:label>

```

```

<rdfs:domain rdf:resource="#Result"/>
<rdfs:range rdf:resource="&rdf;#XMLLiteral"/>
</owl:DatatypeProperty>

```

Ο σκοπός του result form είναι να παρέχει μία αφηρημένη XML βάση για τα αποτελέσματα που στέλνονται πίσω στον πελάτη. Οι λόγοι που χρειαζόμαστε ένα τέτοιο μοτίβο είναι περίπλοκοι και δεν εφαρμόζονται πάντα. Συνήθως η υποστήριξη επαρκεί για να εκφράσει πως οι παράγοντες ενός μηνύματος τοποθετούνται π.χ. πως τα δεδομένα τοποθετούνται μαζί για να στείλουν ένα μήνυμα στην υπηρεσία και πως οι απαντήσεις είναι διαιρεμένες μέσα στα αποτελέσματα που πρόκειται να βγουν. Ως συνέπεια, αυτό που μπορούμε ή χρειάζεται να κάνουμε είναι να δομήσουμε και έπειτα να τεμαχίσουμε τις δομές εγγραφής. Αλλά στην περίπτωση μιας επεξεργασίας με πολλαπλά αποτελέσματα, θα ήταν ιδιαίτερα χρήσιμο να καθορίσουμε άλλα χαρακτηριστικά ενός τελικού μηνύματος που θα υποδεικνύουν πιο αποτέλεσμα έλαβε χώρα, εξοικονομώντας μας την εργασία για την παροχή αποτελεσμάτων όπου θα είναι κωδικοποιημένη αυτή η πληροφορία ή κάνοντας τον πελάτη να το συμπεράνει από άλλα πεδία. Για αυτό το λόγο υπάρχει η φόρμα αποτελεσμάτων (result form).

Στο παράδειγμα της αγοραπωλησίας μέσω πιστωτικής κάρτας, είχαμε 2 αποτελέσματα, στην μια περίπτωση μπορούσε να πληρωθεί ο λογαριασμός και στην άλλη που δεν μπορούσε. Θα μπορούσαμε να ενισχύσουμε κάθε αποτέλεσμα με μια ακόμη σύνδεση, όπως για την περίπτωση της αποτυχίας.

```

<owls:Result>
  <owls:hasResultVar>
    <owls:ResultVar rdf:ID="CreditLimL">
      <owls:parameterType rdf:resource="&ecom;#Dollars"/>
    </owls:ResultVar>
  </owls:hasResultVar>
  <process:inCondition>
    <expr:KIF-Condition>
      <expr:expressionBody>
        (and (current-value (credit-limit ?creditCard)
          ?CreditLimL)

```

```

        (< ?CreditLimL ?purchaseAmt))
    </expr:expressionBody>
</expr:KIF-Condition>
</process:inCondition>
<owls:resultForm rdf:parseType="Literal">
    <ecom:CreditExceededFailure>
        <ecom:gap expressionLanguage="&expr;#KIF"
            rdf:datatype="&xsd;#string">
            (- ?purchaseAmt ?CreditLimL)
        </ecom:gap>
    </ecom:CreditExceededFailure>
</owls:resultForm>
<withOutput rdf:resource="&ecom;failureNotice"/>
    ...
</withOutput>
</owls:Result>

```

3.1.6.3 Ατομικές και Απλές Επεξεργασίες

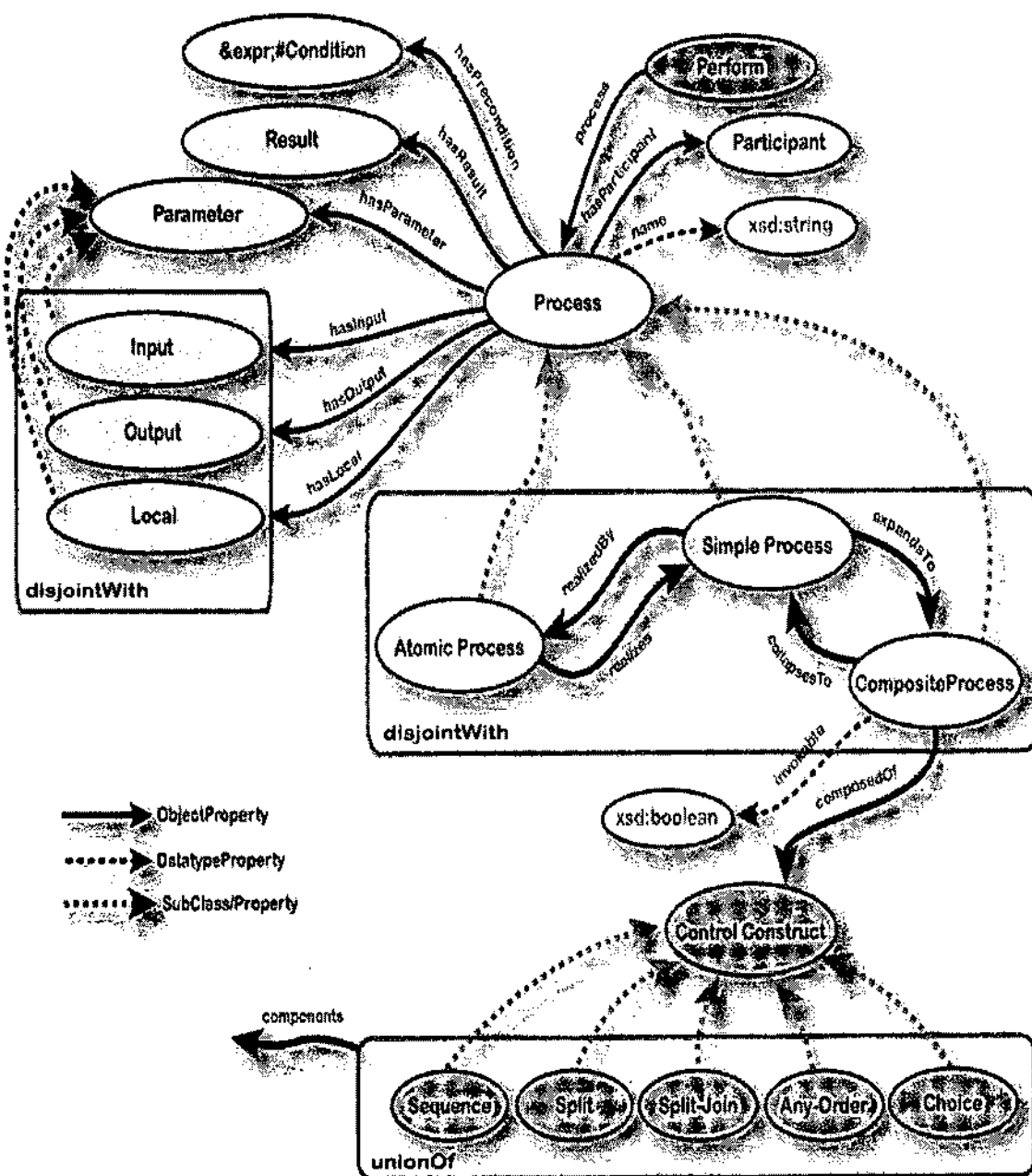
Μπορούμε τώρα να τυποποιήσουμε τις κατηγορίες της επεξεργασίας: ατομικές, σύνθετες και απλές.

```

<owl:Class rdf:ID="Process">
    <rdfs:comment> The most general class of processes </rdfs:comment>
    <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#AtomicProcess"/>
        <owl:Class rdf:about="#SimpleProcess"/>
        <owl:Class rdf:about="#CompositeProcess"/>
    </owl:unionOf>
</owl:Class>

```

Στο Σχήμα 9 μπορούμε να δούμε επιλεγμένες τάξεις και ιδιότητες του μοντέλου διαδικασίας.



Σχήμα 9 - Το Υψηλότερο Επίπεδο της Οντολογίας Διαδικασίας

Οι ατομικές επεξεργασίες αντιστοιχούν στις ενέργειες που μπορεί μια υπηρεσία να εκτελέσει όταν εμπλέκεται σε μια απλή αλληλεπίδραση: οι σύνθετες επεξεργασίες αντιστοιχούν σε ενέργειες που απαιτούν πολλών βημάτων πρωτόκολλα ή/και σε ενέργειες πολλαπλών εξυπηρετητών. Τελικά οι απλές επεξεργασίες παρέχουν έναν αφαιρετικό μηχανισμό για να παρέχουν πολλαπλές πτυχές της ίδιας επεξεργασίας.

Οι ατομικές επεξεργασίες είναι άμεσα επικλήσιμες, δεν έχουν υποεπεξεργασίες και εκτελούνται σε ένα βήμα, όσον αφορά τον αιτών. Παίρνουν ένα μήνυμα δεδομένων και μετά τους επιστρέφεται ένα μήνυμα αποτελεσμάτων. Για κάθε ατομική επεξεργασία, πρέπει να παρέχεται υποστήριξη, η οποία δίνει την δυνατότητα στον αιτούντα της υπηρεσίας να συνθέσει μηνύματα για την επεξεργασία από τα δεδομένα της και από τις αποδομημένες απαντήσεις.

```
<owl:Class rdf:ID="AtomicProcess">  
  <owl:subClassOf rdf:resource="#Process"/>  
</owl:Class>
```

Οι απλές επεξεργασίες δεν είναι επικλήσιμες και δεν σχετίζονται με την υποστήριξη, αλλά σαν ατομικές επεξεργασίες τις αντιλαμβανόμαστε ως διαδικασίες που εκτελούνται σε ένα βήμα. Οι απλές επεξεργασίες χρησιμοποιούνται ως αφηρημένα στοιχεία. Μια απλή επεξεργασία μπορεί να χρησιμοποιηθεί είτε για να παρέχει ένα μέρος μιας ατομικής επεξεργασίας ή για μια απλή αναπαράσταση κάποιων σύνθετων επεξεργασιών. Στο πρώτο παράδειγμα, η απλή επεξεργασία επιτυγχάνεται από την ατομική επεξεργασία, ενώ στο τελευταίο παράδειγμα η απλή επεξεργασία πραγματοποιείται (realized by) στη σύνθετη επεξεργασία, στο τελευταίο παράδειγμα η απλή διαδικασία επεκτείνεται (expands to) σε σύνθετη διαδικασία.

```
<owl:Class rdf:ID="SimpleProcess">  
  <rdfs:subClassOf rdf:resource="#Process"/>  
  <owl:disjointWith rdf:resource="#AtomicProcess"/>  
</owl:Class>
```

```
<owl:ObjectProperty rdf:ID="realizedBy">  
  <rdfs:domain rdf:resource="#SimpleProcess"/>  
  <rdfs:range rdf:resource="#AtomicProcess"/>  
  <owl:inverseOf rdf:resource="#realizes"/>  
</owl:ObjectProperty>
```

```
<owl:ObjectProperty rdf:ID="realizes">  
  <rdfs:domain rdf:resource="#AtomicProcess"/>
```

```
<rdfs:range rdf:resource="#SimpleProcess"/>
<owl:inverseOf rdf:resource="#realizedBy"/>
</owl:ObjectProperty>
```

Τελικά για μια ατομική επεξεργασία, υπάρχουν πάντα δύο συμμετέχοντες. Ο πελάτης και ο εξυπηρετητής.

```
<owl:Class rdf:about="#AtomicProcess">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasClient"/>
      <owl:hasValue rdf:resource="#TheClient"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#performedBy"/>
      <owl:hasValue rdf:resource="#TheServer"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

3.1.6.4 Σύνθετες Επεξεργασίες

Οι σύνθετες επεξεργασίες μπορούν να αποσυντεθούν σε άλλες (μη σύνθετες ή σύνθετες) επεξεργασίες, η αποσύνθεσή τους μπορεί να καθοριστεί χρησιμοποιώντας δομές ελέγχου όπως η Sequence (συνέχεια) και το If-Then-Else (αν-τότε-αλλιώς), τα οποία θα αναφέρουμε παρακάτω. Επειδή πολλές δομές ελέγχου έχουν ονόματα που χρησιμοποιούνταν παλιότερα για τις δομές ελέγχου στις γλώσσες προγραμματισμού, είναι εύκολο να παραβλέψουμε μια βασική διαφορά, ότι μια σύνθετη επεξεργασία δεν είναι η συμπεριφορά που θα ακολουθήσει η υπηρεσία, αλλά η συμπεριφορά ή ένα σύνολο από συμπεριφορές που ο πελάτης μπορεί να έχει

στέλνοντας και λαμβάνοντας μια σειρά μηνυμάτων. Αν η σύνθετη επεξεργασία έχει μια συνολική επίδραση, ο πελάτης θα πρέπει να κάνει ολόκληρη την επεξεργασία με σκοπό να πετύχει αυτή την επίδραση. Δεν έχουμε ακόμα δώσει ακριβή καθορισμό του τι σημαίνει να ολοκληρωθεί μια επεξεργασία, αλλά αυτό που εννοούμε είναι π.χ αν μια σύνθετη επεξεργασία είναι μια σειρά (sequence), τότε ο πελάτης στέλνει μια σειρά μηνυμάτων που επικαλούνται κάθε βήμα στη σειρά.

Ένα βασικό χαρακτηριστικό της σύνθετης επεξεργασίας είναι η προδιαγραφή της, δηλαδή το πως τα δεδομένα γίνονται δεκτά από συγκεκριμένες υποεπεξεργασίες και πως τα διάφορα αποτελέσματα παράγονται από συγκεκριμένες υποεπεξεργασίες.

```
<owl:Class rdf:ID="CompositeProcess">
  <rdfs:subClassOf rdf:resource="#Process"/>
  <owl:disjointWith rdf:resource="#AtomicProcess"/>
  <owl:disjointWith rdf:resource="#SimpleProcess"/>
  <rdfs:comment>
```

Μια σύνθετη επεξεργασία πρέπει να έχει ακριβώς μια composedOf ιδιότητα.

```
</rdfs:comment>
<owl:intersectionOf rdf:parseType="Collection">
  <owl:Class rdf:about="#Process"/>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#composedOf"/>
    <owl:cardinality rdf:datatype="&xsd;#nonNegativeInteger">
      1</owl:cardinality>
  </owl:Restriction>
</owl:intersectionOf>
</owl:Class>
```

Μια σύνθετη επεξεργασία πρέπει να έχει την ιδιότητα composedOf, η οποία υποδεικνύει τη δομή ελέγχου της σύνθετης με τη δομή ControlConstruct.

```
<owl:ObjectProperty rdf:ID="composedOf">
```

```
<rdfs:domain rdf:resource="#CompositeProcess"/>
<rdfs:range rdf:resource="#ControlConstruct"/>
</owl:ObjectProperty>
```

```
<owl:Class rdf:ID="ControlConstruct">
</owl:Class>
```

Κάθε δομή ελέγχου με τη σειρά της σχετίζεται με μια πρόσθετη ιδιότητα που ονομάζεται `components` για να υποδείξει από ποιες δομές ελέγχου δημιουργείται και σε κάποιες περιπτώσεις τη σειρά τους.

```
<owl:ObjectProperty rdf:ID="components">
<rdfs:domain rdf:resource="#ControlConstruct"/>
</owl:ObjectProperty>
```

Παραδείγματος χάριν, κάθε στιγμιότυπο της δομής ελέγχου `Sequence` έχει την ιδιότητα `components`, η οποία λαμβάνει τιμές από μία λίστα με δομές ελέγχου, `ControlConstructList`. Δίνεται παρακάτω ένας πλήρης πίνακας δομών ελέγχου.

Κάθε σύνθετη επεξεργασία μπορεί να θεωρηθεί ως ένα δέντρο του οποίου οι μη-τελικοί κόμβοι χαρακτηρίζονται από δομές ελέγχου, κάθε μία από τις οποίες έχει παιδιά που ορίζονται ως παράγοντες (`components`). Τα φύλλα του δέντρου είναι οι επικλήσεις άλλων επεξεργασιών, και υποδεικνύονται ως στιγμιότυπα της τάξης `Perform`.

```
<owl:Class rdf:ID="Perform">
<rdfs:subClassOf rdf:resource="#ControlConstruct"/>
<rdfs:subClassOf>
<owl:Restriction>
<owl:onProperty rdf:resource="#process"/>
<owl:cardinality rdf:datatype="&xsd;#nonNegativeInteger">
1</owl:cardinality>
</owl:Restriction>
</rdfs:subClassOf>
</owl:Class>
```

Η ιδιότητα της επεξεργασίας (process) μιας απόδοσης υποδεικνύει την επεξεργασία που θα εκτελεστεί.

```
<owl:ObjectProperty rdf:ID="process">  
<rdfs:domain rdf:resource="#Perform"/>  
<rdfs:range rdf:resource="#Process"/>  
</owl:ObjectProperty>
```

Όταν μια επεξεργασία εκτελείται ως βήμα σε μια μεγαλύτερη επεξεργασία, πρέπει να υπάρχει μια περιγραφή για το από που θα προέρχονται τα δεδομένα της επεξεργασίας που εκτελείται και που θα καταλήγουν τα αποτελέσματα.

Μια επεξεργασία μπορεί να παρακολουθηθεί σε διαφορετικά επίπεδα της επεξεργασίας, είτε ως αρχική, μη αποσυντεθημένη επεξεργασία ή σαν μια σύνθετη επεξεργασία. Αυτά αναφέρονται κάποιες φορές ως «μαύρο κουτί» και «γυάλινο κουτί». Κάθε προοπτική μπορεί να είναι πιο χρήσιμη σε κάποιο δεδομένο περιεχόμενο. Όταν μια σύνθετη επεξεργασία παρατηρείται ως μαύρο κουτί, μια απλή επεξεργασία μπορεί να χρησιμοποιηθεί για να την αναπαραστήσει. Στην περίπτωση αυτή, η σχέση μεταξύ της απλής και σύνθετης παρουσιάζεται χρησιμοποιώντας την ιδιότητα `expandsTo` (εκτείνεται) και την αντίστροφη της ιδιότητα `collapsesTo` (καταρρέει).

```
<owl:ObjectProperty rdf:ID="expandsTo">  
<rdfs:domain rdf:resource="#SimpleProcess"/>  
<rdfs:range rdf:resource="#CompositeProcess"/>  
<owl:inverseOf rdf:resource="#collapsesTo"/>  
</owl:ObjectProperty>
```

```
<owl:ObjectProperty rdf:ID="collapsesTo">  
<rdfs:domain rdf:resource="#CompositeProcess"/>  
<rdfs:range rdf:resource="#SimpleProcess"/>  
<owl:inverseOf rdf:resource="#expandsTo"/>  
</owl:ObjectProperty>
```

Συμπεριέχουμε σε αυτόν το τομέα μια περίληψη των δομών ελέγχου του OWL-S: Sequence, Split, Split + Join, Choice, Any-Order, Condition, If-Then-Else, Iterate, Repeat-While, and Repeat-Until.

Sequence: Μια λίστα δομών ελέγχου που πρέπει να γίνει στη σειρά.

```
<owl:Class rdf:ID="Sequence">
  <rdfs:subClassOf rdf:resource="#ControlConstruct"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#components"/>
      <owl:allValuesFrom rdf:resource="#ControlConstructList"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:ID="ControlConstructList">
  <rdfs:comment> A list of control constructs </rdfs:comment>
  <rdfs:subClassOf rdf:resource="#shadow-rdf;#List"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#shadow-rdf;#first"/>
      <owl:allValuesFrom rdf:resource="#ControlConstruct"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#shadow-rdf;#rest"/>
      <owl:allValuesFrom rdf:resource="#ControlConstructList"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

Split: Οι παράγοντες της επεξεργασίας split είναι ένα σύνολο παραγόντων επεξεργασίας που πρέπει να εκτελεστούν παράλληλα. Το split ολοκληρώνεται όταν όλα τα μέρη της διαδικασίας έχουν σχεδιαστεί να εκτελεστούν.

```
<owl:Class rdf:ID="Split">
  <rdfs:subClassOf rdf:resource="#ControlConstruct"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#components"/>
      <owl:allValuesFrom rdf:resource="#ControlConstructBag"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="ControlConstructBag">
  <rdfs:comment> A multiset of control constructs </rdfs:comment>
  <rdfs:subClassOf rdf:resource="#shadow-rdf;#List"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#shadow-rdf;#first"/>
      <owl:allValuesFrom rdf:resource="#ControlConstruct"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#shadow-rdf;#rest"/>
      <owl:allValuesFrom rdf:resource="#ControlConstructBag"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

Split+Join: Εδώ η επεξεργασία περιέχει την παράλληλη εκτέλεση ενός συνόλου παραγόντων επεξεργασίας με συγκεκριμένο συγχρονισμό. Έτσι το Split+Join ολοκληρώνεται όταν όλοι οι παράγοντες επεξεργασίας του έχουν ολοκληρωθεί. Με το

Split και το Split+Join, μπορούμε να ορίσουμε τις επεξεργασίες που έχουν μερικό συγχρονισμό.

```
<owl:Class rdf:ID="Split-Join">
  <rdfs:subClassOf rdf:resource="#ControlConstruct"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#components"/>
      <owl:allValuesFrom rdf:resource="#ControlConstructBag"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

Any-Order: Επιτρέπει στους παράγοντες της επεξεργασίας να εκτελεστούν σε κάποια ακαθόριστη σειρά, αλλά όχι παράλληλα. Η εκτέλεση και η ολοκλήρωση όλων των παραγόντων, απαιτείται. Η εκτέλεση της επεξεργασίας με τη δομή Any-Order δεν μπορεί να επικαλυφθεί. Παραδείγματος χάριν η ατομική επεξεργασία δεν μπορεί να εκτελεστεί παράλληλα και οι σύνθετες επεξεργασίες δεν μπορούν να διαστρωματωθούν. Όλοι οι παράγοντες πρέπει να εκτελεστούν. Όπως και με το Split+Join, η ολοκλήρωση όλων των παραγόντων απαιτείται.

```
<owl:Class rdf:ID="Any-Order">
  <rdfs:subClassOf rdf:resource="#ControlConstruct"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#components"/>
      <owl:allValuesFrom rdf:resource="#ControlConstructBag"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

Choice: Η επιλογή επικαλείται την εκτέλεση μιας δομής ελέγχου από ένα σύνολο δοθέντων δομών ελέγχου. Οποιαδήποτε από τις δεδομένες δομές ελέγχου μπορεί να επιλεγεί για εκτέλεση.

```

<owl:Class rdf:ID="Choice">
  <rdfs:subClassOf rdf:resource="#ControlConstruct"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#components"/>
      <owl:allValuesFrom rdf:resource="#ControlConstructBag"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

```

If-Then-Else: Η τάξη If-Then-Else είναι μια δομή ελέγχου που έχει ιδιότητες ifCondition, then, else συγκρατώντας διαφορετικές οπτικές του If-Then-Else.

```

<owl:Class rdf:ID="If-Then-Else">
  <rdfs:subClassOf rdf:resource="#ControlConstruct"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#components"/>
      <owl:allValuesFrom rdf:resource="#ControlConstructBag"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
<owl:ObjectProperty rdf:ID="ifCondition">
  <rdfs:comment> The if condition of an if-then-else</rdfs:comment>
  <rdfs:domain rdf:resource="#If-Then-Else"/>
  <rdfs:range rdf:resource="#Condition"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="then">
  <rdfs:domain rdf:resource="#If-Then-Else"/>
  <rdfs:range rdf:resource="#ControlConstruct"/>
</owl:ObjectProperty>

```

```

<owl:ObjectProperty rdf:ID="else">
  <rdfs:domain rdf:resource="#If-Then-Else"/>
  <rdfs:range rdf:resource="#ControlConstruct"/>
</owl:ObjectProperty>

```

Iterate: Η δομή Iterate δεν κάνει καμία υπόθεση σχετικά με το πόσες επαναλήψεις έγιναν ή για το πότε θα αρχίσουν, θα τερματιστούν ή θα ξαναρχίσουν. Η αρχή, ο τερματισμός ή η διατήρηση μιας υπόθεσης μπορεί να καθοριστεί με μια συνθήκη `whileCondition` ή με μια `untilCondition`.

Η επανάληψη είναι μια αφηρημένη τάξη, με την έννοια ότι δεν είναι λεπτομερής αρκετά για ένα μοντέλο επεξεργασίας. Είναι καθορισμένη να εξυπηρετεί όπως μια κοινή υπερτάξη του `Repeat-While`, `Repeat-Until` και άλλων πιθανών δομών επανάληψης που μπορεί να χρειαστούν στο μέλλον.

```

<owl:Class rdf:ID="Iterate">
  <rdfs:subClassOf rdf:resource="#ControlConstruct"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#components"/>
      <owl:allValuesFrom rdf:resource="#ControlConstructBag"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

```

Repeat-While και Repeat-Until: Και οι δύο επαναλαμβάνονται μέχρι μια συνθήκη να αποδειχθεί αληθής ή ψευδής, ακολουθώντας τις γνωστές συμβατικές γλώσσες προγραμματισμού. Τα `Repeat-While` τερματίζει για τη συνθήκη δεν υπάρχουν όταν είναι ψευδής και κάνουν την λειτουργία όταν είναι αληθής. Το `Repeat-Until` κάνει την λειτουργία και εξετάζει τη συνθήκη αν είναι αληθής. Επίσης το `Repeat-While` μπορεί ποτέ να μην λειτουργήσει, ενώ το `Repeat-Until` λειτουργεί τουλάχιστον μια φορά.

```

<owl:ObjectProperty rdf:ID="whileCondition">
  <rdfs:domain rdf:resource="#Repeat-While"/>
  <rdfs:range rdf:resource="#&expr;#Condition"/>

```



```
</owl:ObjectProperty>
```

```
<owl:ObjectProperty rdf:ID="whileProcess">  
  <rdfs:domain rdf:resource="#Repeat-While"/>  
  <rdfs:range rdf:resource="#ControlConstruct"/>  
</owl:ObjectProperty>
```

```
<owl:Class rdf:ID="Repeat-While">  
  <rdfs:comment> The repeat while construct</rdfs:comment>  
  <rdfs:subClassOf rdf:resource="#Iterate"/>  
</owl:Class>
```

```
<owl:ObjectProperty rdf:ID="untilCondition">  
  <rdfs:domain rdf:resource="#Repeat-Until"/>  
  <rdfs:range rdf:resource="#&expr;#Condition"/>  
</owl:ObjectProperty>
```

```
<owl:ObjectProperty rdf:ID="untilProcess">  
  <rdfs:domain rdf:resource="#Repeat-Until"/>  
  <rdfs:range rdf:resource="#ControlConstruct"/>  
</owl:ObjectProperty>
```

```
<owl:Class rdf:ID="Repeat-Until">  
  <rdfs:comment> The repeat until process</rdfs:comment>  
  <rdfs:subClassOf rdf:resource="#Iterate"/>  
</owl:Class>
```

3.1.6.5 Καθορίζοντας τη Ροή Δεδομένων και τις Συνδέσεις των Παραμέτρων

Όταν καθορίζουμε επεξεργασίες χρησιμοποιώντας το OWL-S, υπάρχουν πολλά σημεία όπου τα δεδομένα ενός παράγοντα επεξεργασίας, λαμβάνονται ως ένα από τα αποτελέσματα ενός προηγούμενου βήματος. Η φυσική μεταφορά των δεδομένων από την υπηρεσία στον πελάτη και πίσω είναι ένας τύπος ροής δεδομένων από ένα βήμα επεξεργασίας στο άλλο. Υπάρχουν και άλλα μοτίβα.

Συγκεκριμένα, τα αποτελέσματα μιας σύνθετης επεξεργασίας μπορεί να απορρέουν από τα αποτελέσματα ορισμένων παραγόντων και ο καθορισμός κάποιων παραγόντων αποτελεσμάτων προκύπτουν ως ένα αποτέλεσμα X της σύνθετης επεξεργασίας, αποτελεί άλλη μία επίσης προδιαγραφή ροής δεδομένων.

Υιοθετούμε το πρότυπο ότι η πηγή δεδομένων αναγνωρίζεται όταν ο χρήστης των δεδομένων δηλώνεται. Αν το βήμα 1 τροφοδοτεί το βήμα 3, καθορίζουμε το γεγονός αυτό στην περιγραφή του βήματος 3 και όχι στην περιγραφή του βήματος 1. Το αποκαλούμε consumer-pull πρότυπο σε αντίθεση με το εναλλακτικό producer-push. Εφαρμόζουμε αυτό το πρότυπο παρέχοντας μια σήμανση για αυθαίρετους όρους όπως οι τιμές των παραμέτρων των δεδομένων και των αποτελεσμάτων ενός βήματος της επεξεργασίας. Επιπλέον παρέχουμε μια σήμανση για υποόρους που υποδηλώνουν τις παραμέτρους των δεδομένων και των αποτελεσμάτων για προηγούμενα βήματα της επεξεργασίας.

Θα πρέπει να λάβουμε υπόψη μας το παρακάτω πίνακα:

I1 input of: { Composite Process CP }: with output O1

composed of

Step 1: Perform S1 →

Step 2: Perform S2

where S1 has inputs I11 and I12, and output O11
and S2 has input I21 and output O21

Πίνακας 2

Το δεξί βέλος δείχνει ότι το βήμα 2 πρέπει να ακολουθεί το βήμα 1, αλλά όχι απαραίτητα άμεσα.

Ας υποθέσουμε ότι θέλουμε μια ευθεία ροή δεδομένων. Τα δεδομένα εισαγωγής I1 από την όλη διαδικασία CP χρησιμοποιούνται ως δεδομένα εισαγωγής I11, αφού προσθέσουμε το 1. Τα δεδομένα εισαγωγής I12 της S1 είναι μία σταθερά, string «Academic». Τα αποτελέσματα O11 της S1 χρησιμοποιούνται ως δεδομένα εισαγωγής I21 της S2. Το μέγιστο του O και τα αποτελέσματα O21 του S2 επί π, χρησιμοποιούνται ως αποτέλεσμα O1 του CP. Χρησιμοποιώντας τη συνθήκη consumer-pull, δηλώνουμε απλά τις παραμέτρους I1, O11, και O21, αλλά για τις

παραμέτρους I11, I22, και O1 παρέχουμε, μαζί με μία δήλωση, δεσμεύσεις που ορίζουν ότι:

```
I11(Step1) comes from incr(I1(CP))
I12(Step1) = "Academic"
I21(Step2) comes from O11(Step1)
O1(CP) comes from  $\pi \times \max 0$ , O21(Step2))
```

Κάθε μια από αυτές τις ιδιότητες αναπαρίσταται στο OWL-S σαν μια σύνδεση, σαν ένα αφηρημένο αντικείμενο με 2 ιδιότητες, το param, (π.χ. I21 (S2)) και valuespecifier, μια περιγραφή της τιμής της. Σε μια προσπάθεια να παρέχουμε προδιαγραφές τιμής με έναν πιο ακριβή τρόπο σε ένα μεγάλο αριθμό περιπτώσεων, παρέχουμε 4 διαφορετικούς τύπους: valueSource, valueType, valueData, and valueFunction.

Δηλώνουμε την ιδιότητα, το Param, με τον συνήθη τρόπο:

```
<owl:Class rdf:ID="Binding">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#toParam"/>
      <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">
        1</owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

<owl:ObjectProperty rdf:ID="toParam">
  <rdfs:domain rdf:resource="#Binding"/>
  <rdfs:range rdf:resource="#Parameter"/>
</owl:ObjectProperty>
```

Ο πιο απλός τύπος ροής δεδομένων είναι το valueSource:

```
<owl:ObjectProperty rdf:ID="valueSource">
```

```

<rdfs:label>valueSource</rdfs:label>
<rdfs:domain rdf:resource="#Binding"/>
<rdfs:range rdf:resource="#ValueOf"/>
<rdfs:subPropertyOf rdf:resource="#valueSpecifier"/>
</owl:ObjectProperty>

```

Τα εύρος του valueSource είναι ένα απλό αντικείμενο της τάξης ValueOf, ορίζεται ολοκληρωτικά από τις ιδιότητες theVar και fromProcess. Αν η δέσμευση με το toParam= p έχει valueSource = s με ιδιότητες theVar= v και fromProcess = R , αυτό σημαίνει ότι η parameter p της διαδικασίας εξαρτάται από την παράμετρο v της διαδικασίας R .

```

<owl:Class rdf:ID="ValueOf">
<rdfs:label>ValueOf</rdfs:label>
<rdfs:subClassOf>
<owl:Restriction>
<owl:onProperty rdf:resource="#theVar"/>
<owl:cardinality rdf:datatype="&xsd;#nonNegativeInteger">
1</owl:cardinality>
</owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
<owl:Restriction>
<owl:onProperty rdf:resource="#fromProcess"/>
<owl:maxCardinality rdf:datatype="&xsd;#nonNegativeInteger">
1</owl:maxCardinality>
</owl:Restriction>
</rdfs:subClassOf>
</owl:Class>

```

```

<owl:ObjectProperty rdf:ID="theVar">
<rdfs:domain rdf:resource="#ValueOf"/>
<rdfs:range rdf:resource="#Parameter"/>
</owl:ObjectProperty>

```

```

<owl:ObjectProperty rdf:ID="fromProcess">
  <rdfs:domain rdf:resource="#ValueOf"/>
  <rdfs:range rdf:resource="#Perform"/>
</owl:ObjectProperty>

```

Για παράδειγμα ο απλός αυτός πίνακας έχει ένα σημείο όπου η έκφραση `valueSource` μπορεί να χρησιμοποιηθεί. Εδώ φαίνεται το πώς αυτό το σημείο του πίνακα θα εκφραζόταν.

```

<process:Sequence rdf:ID="CP">
  <process:components rdf:parseType="Collection">
    <process:Perform rdf:ID="Step1">
      <process:process rdf:resource="&aux;#S1"/>
      ...
    </process:Perform>
    <process:Perform rdf:ID="Step2">
      <process:process rdf:resource="&aux;#S2"/>
      <process:hasDataFrom> <!-- see below -->
        <process:Binding>
          <process:theParam rdf:resource="&aux;#I21"/>
          <process:valueSource>
            <process:ValueOf>
              <process:theParam rdf:resource="#O11"/>
              <process:fromProcess rdf:resource="#Step1"/>
            </process:ValueOf>
          </process:valueSource>
        </process:Binding>
      </process:hasDataFrom>
    </process:Perform>
  </process:components>
  ...
</process:Sequence>

```

Σε αυτό το παράδειγμα χρησιμοποιούμε την ιδιότητα `has dataform` που συνδέει τα δεδομένα του `Performs` με κάποιες συνδέσεις.

```
<owl:ObjectProperty rdf:ID="hasDataFrom">
  <rdfs:domain rdf:resource="#ProcessComponent"/>
  <rdfs:range rdf:resource="#Binding"/>
</owl:ObjectProperty>
```

Ο ολοκληρωμένος πίνακας φαίνεται παρακάτω, ύστερα από την περαιτέρω αναφορά στις συνδέσεις.

Προκύπτει ένα λεπτό ζήτημα από το γεγονός ότι το εύρος του `fromProcess` είναι `Perform`. Στον ανεπίσημο πίνακα παραπάνω, χρησιμοποιήσαμε εκφράσεις όπως `incr(I1(CP))` για να ονομάσουμε τα δεδομένα `I1` της ολοκληρωμένης διαδικασίας `CP`. Ωστόσο δεν μπορούμε πραγματικά να αναφερθούμε σε μια δέσμευση με `valueSource` που είναι `ValueOf` με `fromProcess=CP` γιατί το `CP` δεν είναι `Perform` αλλά διαδικασία. Αν το σκεφτούμε καλύτερα, δεν έχει νόημα να αναφερόμαστε σε μία τιμή η οποία έχει βγει από μια διαδικασία, γιατί κάθε φορά που επικαλούμαστε την υπηρεσία, θα εμπλέκονται διαφορετικές τιμές. Χρειαζόμαστε μια έκφραση που θα αναφέρεται στη τωρινή τιμή μιας παραμέτρου, η οποία θα είναι η τιμή που λαμβάνει κατά τη διάρκεια ενός πραγματικού `Perform`. Πιο συγκεκριμένα: Κατά τη διάρκεια οποιουδήποτε `Perform P` της `CP`, η τιμή των εισαχθέντων δεδομένων `I1` του βήματος `S1` είναι η τιμή των εισαχθέντων δεδομένων `I1` του `P`.

Επομένως, εισάγουμε μια σταθερή μεταβλητή, για να παίξει το ρόλο του `P`, και δίνουμε το όνομα `TheParentPerform`.

```
<swrl:Variable rdf:ID="TheParentPerform">
  <rdfs:comment>
```

Μία ειδική μεταβλητή, χρησιμοποιείται, κατά το τρέχον χρόνο στο στιγμιότυπο εκτέλεσης, στον ενσωματωμένο ορισμό της σύνθετης διαδικασίας.

```
</rdfs:comment>
</swrl:Variable>
```

Θα δείξουμε πως αυτό χρησιμοποιείται παρακάτω. Τα valueFunction και valueData χρησιμοποιούν XML Literal για να κωδικοποιήσουν τις αυθαίρετες εκφράσεις.

```
<owl:DatatypeProperty rdf:ID="valueFunction">
  <rdfs:label>valueFunction</rdfs:label>
  <rdfs:subPropertyOf rdf:resource="#valueSpecifier"/>
  <rdfs:domain rdf:resource="#Binding"/>
  <rdfs:range rdf:resource="#&rdf;#XMLLiteral"/>
</owl:DatatypeProperty>
```

```
<owl:DatatypeProperty rdf:ID="valueData">
  <rdfs:label>valueData</rdfs:label>
  <rdfs:domain rdf:resource="#Binding"/>
</owl:DatatypeProperty>
```

Το valuefunction μιας σύνδεσης είναι ένα XMLliteral που μπορεί να διαβαστεί ως λειτουργικός όρος. Κάποιοι από τους υποόρους μπορεί να είναι valueof καθορίζοντας τα αποτελέσματα προηγούμενων όρων. Μαζί με τις συνθήκες και τις επιδράσεις, οι επισημάνσεις της έκφρασης valuefunction δεν μπορούν να γίνουν γνωστές μέχρι να συνδεθούν οι τιμές των μεταβλητών. Το valuedata της σύνδεσης είναι ένα XMLliteral που μπορεί να μεταφραστεί σαν σταθερό δεδομένο.

Το valuetype της σύνδεσης είναι ένα URI που αναφέρεται σε μία OWL τάξη ορισμού C. Ένα στιγμιότυπο του valuetype βεβαιώνει ότι η τιμή μιας δεδομένης παραμέτρου θα ανήκει στο c. Το c, πρέπει να είναι ένα υποεπίπεδο του συνολικού τύπου της παραμέτρου, το οποίο καθορίζεται χρησιμοποιώντας parameterType.

```
<owl:DatatypeProperty rdf:ID="valueType">
  <rdfs:label>valueType</rdfs:label>
  <rdfs:subPropertyOf rdf:resource="#valueSpecifier"/>
  <rdfs:domain rdf:resource="#Binding"/>
  <rdfs:range rdf:resource="#&xsd;#anyURI"/>
</owl:DatatypeProperty>
```

Παρακάτω υπάρχει ο ολοκληρωμένος πίνακας του παραδείγματος, με όλες τις ροές δεδομένων να εκφράζονται χρησιμοποιώντας μια από τις τρεις προδιαγραφές πηγής δεδομένων.

```
<process:CompositeProcess rdf:ID="CP">
  <process:hasInput rdf:ID="I1"/>
  <process:hasOutput rdf:ID="O1"/>
  <process:composedOf>
    <process:Sequence rdf:ID="CP">
      <process:components rdf:parseType="Collection">
        <process:Perform rdf:ID="Step1">
          <process:process rdf:resource="&aux;#S1"/>
          <process:hasDataFrom>
            <process:InputBinding>
              <process:theParam rdf:resource="&aux;#I1"/>
              <process:valueFunction expressionLanguage="&drs;"
                rdf:parseType="Literal">
                <drs:Functional_term>
                  <drs:term_function rdf:resource="&arith;#incr"/>
                  <drs:term_args rdf:parseType="Collection">
                    <process:valueOf>
                      <process:theParam rdf:resource="#I1"/>
                      <process:fromProcess
                        rdf:resource="#TheParentPerform"/>
                    </process:valueOf>
                  </drs:term_args>
                </drs:Functional_term>
              </process:valueFunction>
            </process:InputBinding>
          <process:InputBinding>
            <process:theParam rdf:resource="&aux;#I2"/>
            <process:valueData
              xsd:datatype="&xsd;#string"
              >Academic</process:valueData>
```



```

    </process:InputBinding>
  </process:hasDataFrom>
</process:Perform>
<process:Perform rdf:ID="Step2">
  <process:process rdf:resource="&aux;#S2"/>
  <process:hasDataFrom>
    <process:Binding>
      <process:theParam rdf:resource="&aux;#I21"/>
      <process:valueSource>
        <process:ValueOf>
          <process:theParam rdf:resource="#O11"/>
          <process:fromProcess rdf:resource="#Step1"/>
        </process:ValueOf>
      </process:valueSource>
    </process:Binding>
  </process:hasDataFrom>
</process:Perform>
</process:components>

<process:Produce>
  <process:producedBinding>
    <process:OutputBinding>
      <process:theParam rdf:resource="#O1"/>
      <process:valueFunction expressionLanguage="&drs;"
        rdf:parseType="Literal">
        <drs:Functional_term>
          <drs:term_function rdf:resource="&arith;#times"/>
          <drs:term_args rdf:parseType="Collection">
            <xsd:Integer rdf:datatype="&xsd;#Float"
              >3.14159</xsd:Float>
          <drs:Functional_term>
            <drs:term_function
              rdf:resource="&arith;#max"/>
            <drs:term_args rdf:parseType="Collection">

```

```

        <xsd:Integer rdf:datatype="&xsd;#Integer"
            >0</xsd:Integer>
        <process:valueOf>
            <process:theParam
                rdf:resource="#O21"/>
            <process:fromProcess
                rdf:resource="#S2"/>
        </process:valueOf>
    </drs:term_args>
</drs:Functional_term>
</drs:term_args>
</drs:Functional_term>
</process:valueFunction>
</process:OutputBinding>
</process:producedBinding>
</process:Sequence>
</process:composedOf>
</process:CompositeProcess>

```

Το Produce είναι μια νέα τάξη που χρησιμοποιείται για να αιχμαλωτίζει τη ροή δεδομένων στα αποτελέσματα του ParentPerform. Τα αποτελέσματα δεν μπορούν να δηλωθούν μια και για πάντα, επειδή στην παρουσία του If-Then-Else, αυτά τα αποτελέσματα θα εξαρτώνται από ποιο μέρος του υποθετικού παίρνει ο μεσάζων. Έτσι στο κομμάτι αυτό όπου τα δεδομένα υπολογίζονται το αποτέλεσμα είναι γνωστό. Εισάγουμε το Produce, ένα κίβδηλο βήμα για να δούμε ποιο θα είναι το αποτέλεσμα.

```

<owl:Class rdf:ID="Produce">
    <rdfs:subClassOf rdf:resource="#ControlConstruct"/>
</owl:Class>

```

```

<owl:Property rdf:ID="producedBinding">
    <rdfs:domain rdf:resource="#Produce"/>
    <rdfs:range rdf:resource="#OutputBinding"/>

```

</owl:Property>

3.1.7 Υποστηρίζοντας μια Υπηρεσία σε μια Σταθερή Υλοποίηση

Η υποστήριξη μιας υπηρεσίας καθορίζει τις λεπτομέρειες για τον τρόπο πρόσβασης σε μια υπηρεσία, λεπτομέρειες που αφορούν κυρίως το πρωτόκολλο και τη τυποποίηση των μηνυμάτων, τη σειριακή διάταξη, τη μεταφορά και τη διεύθυνση. Η υποστήριξη μπορεί να θεωρηθεί ως μία αντιστοίχιση από μια αφηρημένη σε μια συγκεκριμένη προδιαγραφή των στοιχείων της περιγραφής της υπηρεσίας που απαιτούνται για να αλληλεπιδράσει με την υπηρεσία. Συγκεκριμένα, για τους σκοπούς μας, τα δεδομένα και τα αποτελέσματα των ατομικών επεξεργασιών. Θα πρέπει να σημειώσουμε ότι στο OWL-S, το Προφίλ Υπηρεσίας και το Μοντέλο Υπηρεσίας θεωρούνται ως αφηρημένες αναπαραστάσεις, μόνο η Υποστήριξη της Υπηρεσίας ασχολείται με το ολοκληρωμένο επίπεδο της προδιαγραφής.

Το OWL-S δεν περιέχει μια αφηρημένη δομή για σαφή περιγραφόμενα μηνύματα. Το αφηρημένο περιεχόμενο ενός μηνύματος καθορίζεται από τις ιδιότητες των δεδομένων και του αποτελέσματος ορισμένων ατομικών επεξεργασιών. Επίσης οι ατομικές επεξεργασίες, μαζί με το καθορισμό των βασικών ενεργειών από τις οποίες μεγαλύτερες επεξεργασίες συνθέτονται, μπορούν να θεωρηθούν ως πρωταρχικά στοιχεία επικοινωνίας μιας αφηρημένης προδιαγραφής επεξεργασίας.

Τα ολοκληρωμένα μηνύματα καθορίζονται λεπτομερώς στην υποστήριξη. Η κεντρική λειτουργία της υποστήριξης του OWL-S είναι να δείξει πως τα (αφηρημένα) δεδομένα και αποτελέσματα των ατομικών επεξεργασιών πραγματοποιούνται ολοκληρωτικά ως μηνύματα, τα οποία μεταφέρουν αυτά τα δεδομένα και τα αποτελέσματα σε μια συγκεκριμένη δομή μετάδοσης. Λόγω του ότι υπάρχει ένας σημαντικός όγκος εργασίας πάνω στο τομέα της προδιαγραφής των ολοκληρωμένων μηνυμάτων, επιλέξαμε να κάνουμε χρήση της WSDL για τη δημιουργία ενός αρχικού μηχανισμού υποστήριξης του OWL-S. Όπως αναφέραμε παραπάνω ο σκοπός μας δεν είναι να καθορίσουμε μία πιθανή προσέγγιση της υποστήριξης για να χρησιμοποιηθεί με όλες τις υπηρεσίες, αλλά περισσότερο να παρέχουμε μια γενική, κανονική και ευρέως εφαρμόσιμη προσέγγιση, η οποία θα είναι χρήσιμη για το μεγαλύτερο μέρος των περιπτώσεων.

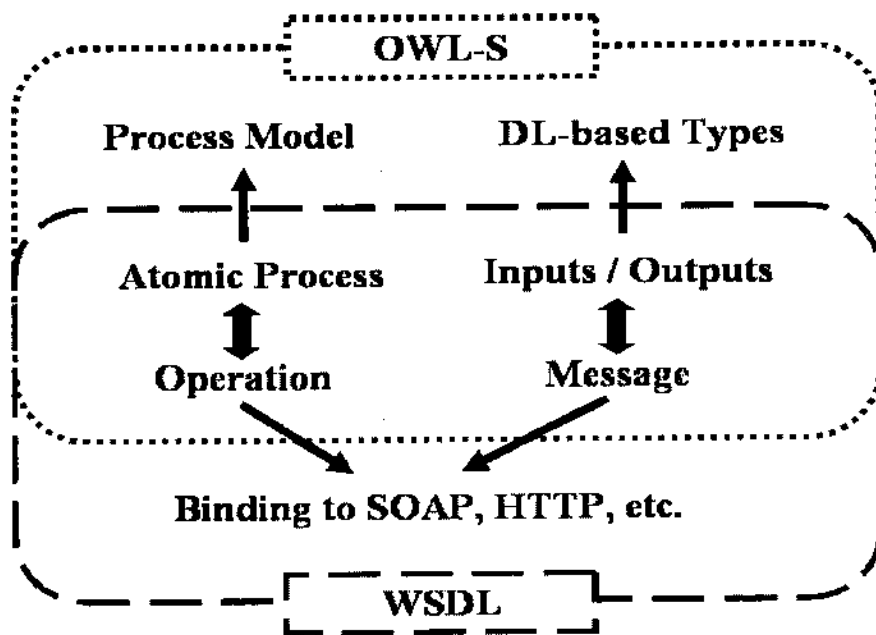
Η WSDL είναι ένας τύπος XML για την περιγραφή διαδικτυακών υπηρεσιών ως ένα σύνολο από τελικά σημεία το οποίο λειτουργεί πάνω στα μηνύματα,

περιέχοντας πληροφορίες που είτε θα προσανατολίζονται στο κείμενο είτε θα προσανατολίζονται στην διαδικασία. Οι λειτουργίες και τα μηνύματα περιγράφονται αφηρημένα και μετά οριοθετούνται σε ένα ολοκληρωμένο διαδικτυακό πρωτόκολλο και σε ένα τύπο μηνύματος για να ορίσουν ένα τελικό σημείο. Τα σχετικά ολοκληρωμένα τελικά σημεία συνδυάζονται σε αφηρημένα τελικά σημεία (υπηρεσίες). Η WSDL είναι επεκτάσιμη και έτσι επιτρέπει την περιγραφή των τελικών σημείων και των μηνυμάτων τους ανεξάρτητα από τη δομή των μηνυμάτων ή από τα διαδικτυακά πρωτόκολλα που χρησιμοποιούνται.

Μπορούμε εύκολα να διαπιστώσουμε ότι η έννοια της υποστήριξης του OWL-S συμφωνεί σε γενικές γραμμές με την έννοια της δέσμευσης της WSDL. Όντως, χρησιμοποιώντας τα στοιχεία επεκτασιμότητας που παρέχονται από την WSDL, μαζί με νέα στοιχεία που προτείνονται εδώ, είναι εύκολο να υποστηρίξουμε μία ατομική επεξεργασία του OWL-S. Δείχνουμε πως μπορεί να γίνει αν βασιστούμε στην προδιαγραφή της WSDL 1.1.

3.1.7.1 Η Σχέση μεταξύ του OWL-S και της WSDL

Η προσέγγιση που περιγράψαμε επιτρέπει στον δημιουργό της υπηρεσίας, που πρόκειται να παρέχει κάποιες περιγραφές της υπηρεσίας για χρήση από τους ενδεχομένους πελάτες, να εκμεταλλευτεί τις συμπληρωματικές δυνατότητες των προδιαγραφών των δύο γλωσσών. Από τη μια πλευρά (το αφηρημένο κομμάτι της προδιαγραφής μιας υπηρεσίας) ο δημιουργός επωφελείται χρησιμοποιώντας το μοντέλο επεξεργασίας του OWL-S και την εκφραστικότητα της τάξης του OWL-S με τους τυπογραφικούς μηχανισμούς. Από την άλλη πλευρά (το ολοκληρωμένο κομμάτι), ο δημιουργός κερδίζει από την ευκαιρία να επαναχρησιμοποιήσει την εκτεταμένη εργασία που έγινε στο WSDL (και στις σχετικές γλώσσες όπως η SOAP) και στην λογισμική υποστήριξη για ανταλλαγές μηνυμάτων που βασίζονται σε αυτές τις δηλώσεις.



Σχήμα 10 - Αντιστοίχιση ανάμεσα στο OWL-S και τη WSDL

Θα πρέπει να δώσουμε έμφαση στο ότι η υποστήριξη του OWL-S/WSDL αφορά μια συμπληρωματική χρήση των δυο γλωσσών, με ένα σύμφωνο τρόπο με τους συγγραφείς της WSDL. Και οι δύο γλώσσες απαιτούνται για το πλήρη καθορισμό μιας υποστήριξης, γιατί και οι δυο γλώσσες δεν καλύπτουν το ίδιο εννοιολογικό πεδίο. Όπως φαίνεται και στο παραπάνω σχήμα και οι δύο γλώσσες επικαλύπτουν την παροχή για τον καθορισμό της προδιαγραφής του τι καλεί η WSDL αφηρημένους τύπους, οι οποίοι με τη σειρά τους χρησιμοποιούνται για να χαρακτηρίσουν τα δεδομένα και τα αποτελέσματα των υπηρεσιών. Η WSDL εξ' ορισμού καθορίζει τους αφηρημένους τύπους χρησιμοποιώντας την XML Schema, όπου το OWL-S επιτρέπει τον καθορισμό των αφηρημένων τύπων ως OWL τάξεις. Ωστόσο, το WSDL/XSD δεν είναι ικανό να εκφράσει τη σημασιολογία μιας OWL τάξης. Παρομοίως, το OWL-S δεν έχει τα μέσα να εκφράσει τις πληροφορίες σύνδεσης, που η WSDL αιχμαλωτίζει. Επομένως είναι φυσικό, η υποστήριξη του OWL-S/WSDL, να χρησιμοποιεί τάξεις του OWL ως αφηρημένους τύπους του τμήματος ενός μηνύματος που δηλώνονται στη WSDL, και μετά βασίζεται στις δομές σύνδεσης της WSDL για να καθορίσει την τυποποίηση των μηνυμάτων.

Μια υποστήριξη του OWL-S/WSDL βασίζεται στις ακόλουθες τρεις αντιστοιχίες ανάμεσα στο OWL-S και στο WSDL.

1. Μία ατομική επεξεργασία του OWL-S αντιστοιχεί σε μια λειτουργία WSDL. Διαφορετικοί τύποι λειτουργιών σχετίζονται με τις επεξεργασίες OWL-S ως ακολούθως:

- Μία ατομική επεξεργασία μαζί με τα δεδομένα και τα αποτελέσματα αντιστοιχεί σε μια λειτουργία ερωταπόκρισης της WSDL.
- Μια ατομική επεξεργασία με δεδομένα, αλλά όχι και αποτελέσματα αντιστοιχεί στη λειτουργία μιας κατεύθυνσης της WSDL.
- Η ατομική επεξεργασία με αποτελέσματα, αλλά χωρίς δεδομένα αντιστοιχεί σε μια λειτουργία γνωστοποίησης της WSDL.
- Η σύνθετη λειτουργία μαζί με αποτελέσματα και δεδομένα και την αποστολή των αποτελεσμάτων να ορίζεται ως εισερχόμενη πριν την αποδοχή των δεδομένων, αντιστοιχεί στη λειτουργία αίτηση-απάντηση της WSDL.

Η υποστήριξη του OWL-S δεν απαιτεί μια ένα προς ένα αντιστοιχία ανάμεσα σε μία ατομική επεξεργασία και σε μια μοναδική WSDL λειτουργία. Για να προσαρμόσουμε τη πρακτική υποστήριξης της WSDL, να παρέχει πολλαπλούς προσδιορισμούς της ίδιας λειτουργίας, το OWL-S επιτρέπει την ένα προς πολλά αντιστοιχία ανάμεσα στην ατομική επεξεργασία και τις πολλαπλές λειτουργίες του WSDL. Είναι επίσης πιθανό, σε αυτές τις περιπτώσεις να διατηρήσει μια ένα προς ένα αντιστοιχία χρησιμοποιώντας πολλαπλές ατομικές επεξεργασίες.

2. Κάθε ένα από τα σύνολα των δεδομένων και των αποτελεσμάτων της ατομικής επεξεργασίας του OWL-S αντιστοιχεί με την έννοια του μηνύματος της WSDL. Τα δεδομένα του OWL-S αντιστοιχούν με τα μέρη ενός εισερχόμενου μηνύματος δεδομένων, της λειτουργίας του WSDL. Τα αποτελέσματα του OWL-S αντιστοιχούν με τα μέρη του μηνύματος αποτελεσμάτων της λειτουργίας του WSDL.

3. Οι τύποι των δεδομένων και των αποτελεσμάτων της ατομικής επεξεργασίας του OWL-S αντιστοιχούν με την εκτεταμένη έννοια του αφηρημένου τύπου της WSDL.

Για να δημιουργήσουμε την υποστήριξη OWL-S/WSDL πρώτα πρέπει να αναγνωρίσουμε, στην WSDL, τα μηνύματα και τις λειτουργίες μέσω των οποίων υπάρχει πρόσβαση σε μια ατομική επεξεργασία, και έπειτα να οριστούν οι αντιστοιχίες (1)-(3).

Πριν από την έκδοση 0.9 του OWL-S, οι αντιστοιχίες (2) και (3) απαιτούνταν για να είναι άμεσες οι αντιστοιχίες. Για αυτό τα δεδομένα και τα αποτελέσματα του OWL-S έπρεπε άμεσα να ταιριάζουν με ένα συγκεκριμένο μέρος μηνύματος της WSDL. Ξεκινώντας με την έκδοση 0.9, αυτός ο περιορισμός δεν ισχύει πια, επιτρέπει στη προδιαγραφή των XSLT μετατροπών να δείξει πως τα δεδομένα του WSDL πηγάζουν από (μία ή περισσότερες) ιδιότητες των δεδομένων του OWL-S και πως τα αποτελέσματα πηγάζουν από ένα ή περισσότερα μέρη μηνυμάτων των αποτελεσμάτων του WSDL.

Παρότι δεν είναι απαραίτητο, είναι χρήσιμο να καθορίσουμε τις αντιστοιχίες στο WSDL και στο OWL-S. Επίσης επιτρέπουμε τις δομές και στις δύο γλώσσες για αυτό το σκοπό.

3.1.7.2 Υποστήριξη στις Υπηρεσίες του OWL-S με το WSDL και το SOAP

Επειδή το OWL-S είναι μια γλώσσα βασισμένη στην XML και η δήλωση των ατομικών επεξεργασιών και των τύπων των δεδομένων και των αποτελεσμάτων ταιριάζουν με την WSDL, είναι εύκολο να επεκτείνουμε τις υπάρχουσες WSDL συνδέσεις για χρήση με το OWL-S, όπως στην SOAP σύνδεση. Εδώ παρουσιάζουμε σύντομα πως σε μια αυθαίρετη ατομική επεξεργασία, καθορισμένη στο OWL-S, μπορεί να δοθεί υποστήριξη χρησιμοποιώντας το WSDL και το SOAP, υποθέτοντας ότι το HTTP θα είναι ο επιλεγμένος μηχανισμός μεταφοράς.

3.1.7.2.1 Η Προδιαγραφή της WSDL

Η υποστήριξη του OWL-S με το WSDL και το SOAP εμπεριέχουν τη δημιουργία μιας περιγραφής της υπηρεσίας του WSDL με όλα τα συνήθη μέρη (τύποι, μηνύματα, λειτουργία, τύποι θύρας, σύνδεση και δομές υπηρεσίας).

Με σεβασμό στους τύπους των μερών των μηνυμάτων της WSDL, θα ήταν χρήσιμο να διακρίνουμε δύο περιπτώσεις: αυτές των οποίων ο τύπος είναι ένας OWL τύπος και όλους τους άλλους. Στην πρώτη περίπτωση, η OWL τάξη μπορεί είτε να καθοριστεί στον τομέα τύπων της WSDL, είτε να καθορίζεται σε ένα χωριστό κείμενο

που αναφέρεται από την περιγραφή της WSDL, χρησιμοποιώντας μια παράμετρο του OWL-S.

Οι επεκτάσεις του OWL-S παρουσιάζονται παρακάτω:

1. Ένα μέρος του καθορισμού των μηνυμάτων της WSDL, το χαρακτηριστικό της παραμέτρου του OWL-S, μπορεί να χρησιμοποιηθεί για να υποδείξει το συνολικό απαιτούμενο όνομα του αντικείμενου των δεδομένων ή των αποτελεσμάτων του OWL-S, στο οποίο αντιστοιχεί αυτό το μέρος του μηνύματος. Το χαρακτηριστικό αυτό είναι ιδιαίτερα χρήσιμο για τις περιπτώσεις που τύπος του μέρους του μηνύματος είναι τύπος OWL. Σε αυτές τις περιπτώσεις, ο καθορισμός του τύπου του OWL μπορεί να αποκομιστεί από την προδιαγραφή της επεξεργασίας του OWL-S παρακολουθώντας την ιδιότητα `parameterType`, που αναφέρεται στο αντικείμενο των δεδομένων ή των αποτελεσμάτων.
2. Γι' αυτές τις περιπτώσεις, μέσα από τις οποίες ένα μέρος του μηνύματος χρησιμοποιεί ένα τύπο OWL, στο χαρακτηριστικό `encodingStyle`, μέσα στο στοιχείο σύνδεσης της WSDL, μπορεί να δοθεί μια τιμή όπως `"http://www.w3.org/2002/07/owl"` για να δείξει ότι τα μέρη των μηνυμάτων θα μπουν σε σειριακή μορφή για τα στιγμιότυπα της τάξης των δεδομένων τύπων για την καθορισμένη έκδοση του OWL.
3. Σε κάθε στοιχείο λειτουργίας του WSDL, το χαρακτηριστικό της επεξεργασίας του OWL-S μπορεί να χρησιμοποιηθεί για να υποδείξει το όνομα της ατομικής επεξεργασίας του OWL-S, με το οποίο η λειτουργία αντιστοιχεί.

Θα πρέπει να σημειώσουμε ότι η WSDL επιτρέπει τη χρήση αυθαίρετων νέων χαρακτηριστικών στα στοιχεία μέρους των μηνυμάτων και για τη χρήση αυθαίρετων τιμών των χαρακτηριστικών `encodingStyle`. Επομένως η επέκταση (3) είναι το μόνο σημείο όπου εντοπίζεται μια αλλαγή στη προδιαγραφή WSDL 1.1.

3.1.7.2.2 Η Τάξη Υποστήριξης του OWL-S

Μέχρι στιγμής έχουμε δείξει πως οι προσδιορισμοί της WSDL μπορούν να αναφέρονται στις αντίστοιχες δηλώσεις του OWL-S. Αυτό που απομένει είναι να καθιερώσουμε ένα μηχανισμό με τον οποίο οι σχετικές δομές του WSDL, θα

μπορούν να αναφέρονται στο OWL-S. Η τάξη `WsdlGrounding` στο OWL-S, μια υποτάξη της υποστήριξης, εξυπηρετεί αυτό το σκοπό. Κάθε στιγμιότυπο της `WsdlGrounding` περιέχει μια λίστα `WsdlAtomicProcessGrounding` στιγμιότυπων. Ένα στιγμιότυπο `WsdlAtomicProcessGrounding` αναφέρεται σε συγκεκριμένα στοιχεία μέσα στην WSDL προδιαγραφή, χρησιμοποιώντας τις ακόλουθες ιδιότητες:

- `WsdlVersion`: Ένα URI που υποδεικνύει ποια έκδοση του WSDL χρησιμοποιείται.
- `WsdlDocument`: Ένα URI ενός εγγράφου του WSDL στο οποίο αναφέρεται αυτή η υποστήριξη.
- `WsdlOperation`: Το URI της WSDL λειτουργίας αντιστοιχεί με την δεδομένη ατομική επεξεργασία.
- `WsdlService`, `WsdlPort` (προαιρετικά): Τα URI της υπηρεσίας WSDL που παρέχει τη δεδομένη λειτουργία. Η τιμή του `WsdlOperation` μπορεί να αναγνωρίσει ή να μην αναγνωρίσει ολοκληρωτικά μία συγκεκριμένη WSDL με την οποία να αλληλεπιδρά. Αν υπάρχουν πολλές εισοδοί που παρέχουν την συγκεκριμένη λειτουργία, ένας μηχανισμός θεσμοθέτησης του OWL-S είναι ελεύθερος για να επιλέξει οποιαδήποτε από αυτές τις θύρες. Αν θέλουμε να περιορίσουμε περαιτέρω την επιλογή θυρών, η `WsdlAtomicProcessGrounding` μπορεί να ορίσει ένα οποιοδήποτε αριθμό θυρών για την `WsdlService` και/ή την `WsdlPort`.
- `WsdlInputMessage`: Ένα αντικείμενο που περιέχει το URI του ορισμού του μηνύματος του WSDL που περιέχει τα δεδομένα της δοθείσας ατομικής επεξεργασίας.
- `WsdlInput`: Ένα αντικείμενο περιέχει ένα ζευγάρι αντιστοίχισης, για ένα μέρος των δεδομένων του WSDL μηνύματος. Κάθε ζευγάρι αναπαρίσταται χρησιμοποιώντας ένα στιγμιότυπο του `WsdlInputMessageMap`. Ένα στοιχείο του ζευγαριού αναγνωρίζει το μέρος του μηνύματος, χρησιμοποιώντας το URI. Το άλλο στοιχείο μας δείχνει πως να αποκομίσουμε αυτό το μέρος του μηνύματος από ένα ή περισσότερα δεδομένα της ατομικής επεξεργασίας OWL-S. Στις απλές περιπτώσεις στις οποίες το μέρος του μηνύματος αντιστοιχεί άμεσα σε ένα δεδομένα του OWL-S και ο τύπος του δεδομένου χρησιμοποιείται άμεσα από την προδιαγραφή της WSDL. Σε όλες τις άλλες

περιπτώσεις η ιδιότητα `XsltTransformation` δίνει ένα σενάριο XSLT που δημιουργεί το μέρος του μηνύματος από ένα στιγμιότυπο της ατομικής επεξεργασίας.

- `WsdlOutputMessage`: Ισχύει το ίδιο όπως και με το `WsdlInputMessage`, αλλά για αποτελέσματα.
- `WsdlOutput`: Παρόμοια με το `WsdlInput`, αλλά για αποτελέσματα. Σ' αυτήν την περίπτωση έχουμε ένα ζευγάρι αντιστοίχισης για ένα αποτέλεσμα της ατομικής επεξεργασίας του OWL-S, το οποίο παρουσιάζεται χρησιμοποιώντας ένα στιγμιότυπο του `WsdlOutputMessageMap`. Κάθε τέτοιο ζευγάρι περιέχει ένα στιγμιότυπο του `OwlsParameter` που καθορίζει τ' αποτέλεσμα. Το άλλο στοιχείο του ζευγαριού μπορεί να είναι είτε το `WsdlMessagePart`, όταν υπάρχει άμεση αντιστοιχία μ' ένα συγκεκριμένο μέρος του μηνύματος, ή το `XsltTransformation` για όλες τις άλλες περιπτώσεις.

3.1.8 Περίληψη και Τρέχουσα Κατάσταση

Το OWL-S είναι μια οντολογία, βασισμένη στο πλαίσιο του OWL του σημασιολογικού δικτύου, που χρησιμοποιείται για την περιγραφή των υπηρεσιών δικτύου. Θα δίνει τη δυνατότητα στους χρήστες και στους μεσάζοντες λογισμικού να βρίσκουν, να ζητούν βοήθεια, να συνθέτουν και να παρακολουθούν αυτόματα τις πηγές του δικτύου που παρέχουν υπηρεσίες, κάτω από καθορισμένους περιορισμούς.

Ευχόμαστε να εμπλουτίσουμε το OWL-S στο μέλλον με καθορισμένους τρόπους που έχουμε συμπεριλάβει σε αυτή την τεχνική περίληψη και σε σχέση με την εμπειρία του χρήστη με αυτό.

Πιστεύουμε ότι το OWL-S θα βοηθήσει το σημασιολογικό δίκτυο να γίνει ένα μέρος όπου οι άνθρωποι δεν θα μπορούν μόνο να βρίσκουν πληροφορίες, αλλά και να εκτελούν εργασίες.

3.1.9 Παραρτήματα

- **Παράρτημα Α- Σημείωση πάνω στη Συμβατότητα των Τύπων των Δεδομένων και των Αποτελεσμάτων**

Δύο επεξεργασίες είναι συμβατοί τύποι αν κάθε αποτέλεσμα της μίας γίνεται δεδομένο της άλλης, το `parameterType` του αποτελέσματος είναι ένας υποτύπος του `parameterType` του δεδομένου. Συχνά χρησιμοποιούμε τις τάξεις του OWL ως τύπους παραμέτρων στις περιγραφές του OWL-S. Γι' αυτό, το URI που είναι μια τιμή της ιδιότητας `parameterType`, υποδηλώνει μία OWL τάξη, δηλαδή ένα σύνολο χαρακτηριστικών που αναφέρονται στη κωδικοποιημένη περιγραφή του OWL. Ένας μηχανισμός αιτιολόγησης του OWL μπορεί να καθορίσει το τύπο συμβατότητας σε τέτοιες περιπτώσεις, εξετάζοντας αν ο σχετικός τύπος αποτελέσματος υπάγεται στον τύπο δεδομένων. Σε περίπτωση όπου οι τύποι είναι τύποι XML Schema, η σχέση των υποτύπων καθορίζεται από την προδιαγραφή XML Schema (και μπορεί να εξεταστεί από το μηχανισμό επικύρωσης της XML Schema).

Οι τύποι της XML Schema και οι τάξεις του OWL είναι τύποι ασύμβατοι. Αν μια υπηρεσία χρησιμοποιεί τους τύπους της XML Schema σε δηλώσεις WSDL και ένας σχεδιαστής υπηρεσίας επιθυμεί να χρησιμοποιήσει την OWL στην επεξεργασία ή στο επίπεδο του προφίλ, ο σχεδιαστής θα πρέπει να εκτελέσει τη διαδικασία της μετάφρασης από την XML Schema στο OWL.

- **Παράρτημα Β- Σημείωση πάνω στις Λίστες Σήμανσης του OWL-S**

Στην OWL DL, το λεξιλόγιο της RDF List (`rdf:List`, `rdf:first`, `rdf:rest`, `rdf:nil`) μπορεί να χρησιμοποιηθεί σε συνδυασμό με τα κατηγορήματα δόμησης όπως: `owl:unionOf`, `owl:intersectionOf`, κ.τ.λ. Στην OWL Full, επιτρέπεται να χρησιμοποιήσουμε τις λίστες RDF για να δηλώσουμε μια λίστα χαρακτηριστικών. Το λεξιλόγιο της λίστας μπορεί να χρησιμοποιηθεί ως συντακτικό στην OWL DL, και όχι για την κύρια σχεδίαση.

Για να κάνουμε τις οντολογίες της OWL προσβάσιμες στους υπάρχοντες μηχανισμούς αιτιολόγησης της OWL DL, το OWL-S Coalition έχει αποφασιστεί να υπακούει σ' αυτούς τους περιορισμούς. Δυστυχώς, οι δομές ελέγχου στην οντολογία

επεξεργασίας, χρησιμοποιούν τη συλλογή λεξιλογίου. Στο OWL-S 1.0 οι δομές μοντελοποιούνται όπως οι λίστες RDF και το χαρακτηριστικό `rdf:parsetype="collection"` χρησιμοποιούνταν για να γράψουμε τη λίστα. Παραδείγματος χάριν:

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:process-1.0="http://www.daml.org/services/owl-s/1.0/ Process.owl#">
<process-1.0:CompositeProcess rdf:ID="BravoAir_Process">
  <process-1.0:composedOf>
    <process-1.0:Sequence>
      <process-1.0:components rdf:parseType="Collection">
        <process-1.0:AtomicProcess rdf:about="#GetDesiredFlightDetails"/>
        <process-1.0:AtomicProcess rdf:about="#SelectAvailableFlight"/>
        <process-1.0:CompositeProcess rdf:about="#BookFlight"/>
      </process-1.0:components>
    </process-1.0:Sequence>
  </process-1.0:composedOf>
</process-1.0:CompositeProcess>
</rdf:RDF>
```

Το παραπάνω είναι ακριβώς ίδιο με την ακόλουθη αναλυτική παρουσίαση.

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:process-1.0="http://www.daml.org/services/owl-s/1.0/ Process.owl#">
<process-1.0:CompositeProcess rdf:ID="BravoAir_Process">
  <process-1.0:composedOf>
    <process-1.0:Sequence>
      <process-1.0:components>
        <rdf:Description>
          <rdf:first>
            <process-1.0:AtomicProcess rdf:about="#GetDesiredFlightDetails"/>
          </rdf:first>
          <rdf:rest>
            <rdf:Description>
```

```

<rdf:first>
  <process-1.0:AtomicProcess rdf:about="#SelectAvailableFlight"/>
</rdf:first>
<rdf:rest>
  <rdf:Description>
    <rdf:first>
      <process-1.0:CompositeProcess rdf:about="#BookFlight"/>
    </rdf:first>
    <rdf:rest rdf:resource="&rdf;#nil"/>
  </rdf:Description>
</rdf:rest>
</rdf:Description>
</rdf:rest>
</rdf:Description>
</process-1.0:components>
</process-1.0:Sequence>
</process-1.0:composedOf>
</process-1.0:CompositeProcess>
</rdf:RDF>

```

Στο OWL-S 1.1 η λύση του να μείνει στην OWL DL, επιτυγχάνεται χρησιμοποιώντας ένα «συνοδευτικό» λεξιλόγιο για να επαναπροσδιορίσουμε τις έννοιες List, First, Rest και Nil με διαφορετικές ετικέτες ονομασίας. Η οντολογία της συνοδευτικής λίστας εντοπίζεται στο <http://www.daml.org/services/owl-s/1.1/generic/ObjectList.owl>. Η χρήση αυτού του λεξιλογίου στο OWL-S είναι παρόμοια με το παραπάνω παράδειγμα:

```

<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:shadow-rdf="http://www.daml.org/services/owl-s/1.1/generic/ ObjectList.owl"
  xmlns:process-1.1="http://www.daml.org/services/owl-s/1.1/ Process.owl#">
<process-1.1:CompositeProcess shadow-rdf:ID="BravoAir_Process">
  <process-1.1:composedOf>
    <process-1.1:Sequence>
      <process-1.1:components>

```

```

<rdf:Description>
  <shadow-rdf:first>
    <process-1.1:AtomicProcess
      rdf:about="#GetDesiredFlightDetails"/>
  </shadow-rdf:first>
  <shadow-rdf:rest>
    <rdf:Description>
      <shadow-rdf:first>
        <process-1.1:AtomicProcess
          rdf:about="#SelectAvailableFlight"/>
        </shadow-rdf:first>
        <shadow-rdf:rest>
          <rdf:Description>
            <shadow-rdf:first>
              <process-1.1:CompositeProcess
                rdf:about="#BookFlight"/>
            </shadow-rdf:first>
            <shadow-rdf:rest
              rdf:resource="&shadow-rdf;#nil"/>
          </rdf:Description>
        </shadow-rdf:rest>
      </rdf:Description>
    </shadow-rdf:rest>
  </rdf:Description>
</shadow-rdf:rest>
</process-1.1:components>
</process-1.1:Sequence>
</process-1.1:composedOf>
</process-1.1:CompositeProcess>
</rdf:RDF>

```

Σε αυτή τη περίπτωση δεν μπορούμε να χρησιμοποιήσουμε την επισήμανση `rdf:parsetype=collection` από τη στιγμή που χρησιμοποιούμε ένα τελείως διαφορετικό σύνολο ιδιοτήτων για να σημάνουμε τις λίστες. Η υπάρχουσα συνοδευτική λίστα λεξιλογίου είναι περισσότερο μια ρηχή παρουσίαση των λιστών, αλλά δεν είναι

περισσότερο «φτωχή» από το αυθεντικό λεξιλόγιο της RDF. Στο OWL Full, δεν υπάρχει κανένας περιορισμός στη χρήση της RDF λίστας λεξιλογίου και είναι επίσης δυνατό να καθορίσουμε σχέσεις ισότητας ανάμεσα στη λίστα λεξιλογίου της RDF και στη συνοδευτική λίστα λεξιλογίου. Το ακόλουθο σύνολο ισχυρισμών είναι ένας τρόπος για να καθιερώσουμε αυτή την ισότητα:

```
<rdf:Description rdf:ID="&rdf;#List">
  <owl:equivalentClass rdf:resource="&shadow-rdf;#List"/>
</rdf:Description>
<rdf:Property rdf:about="&rdf;#first">
  <owl:equivalentProperty rdf:resource="&shadow-rdf;#first"/>
</rdf:Property>
<rdf:Property rdf:about="&rdf;#rest">
  <owl:equivalentProperty rdf:resource="&shadow-rdf;#rest"/>
</rdf:Property>
<rdf:Description rdf:ID="&rdf;#nil">
  <owl:sameAs rdf:resource="&shadow-rdf;#nil"/>
</rdf:Description>
```

Ο προμηθευτής της υπηρεσίας μπορεί απλά να εισάγει μια οντολογία, όπου περιέχει αυτούς τους ισχυρισμούς και να συνεχίσει να χρησιμοποιεί το ενσωματωμένο λεξιλόγιο της RDF. Τέτοιες περιγραφές υπηρεσίας είναι ακόμα σύμφωνες με τις οντολογίες του OWL, αλλά ταυτόχρονα αυτά εμπίπτουν στην OWL Full. Είναι πιο δύσκολο να χρησιμοποιήσουμε κάποιον άλλο τρόπο, από τη στιγμή που ένας θα πρέπει να διατηρήσει τη διάκριση ανάμεσα στη συντακτική χρήση του λεξιλογίου της RDF και τις χρήσεις για τη μοντελοποίηση της κυριότητας. Απαιτεί μια προηγούμενη κίνηση του `owlparser` όπως στο OWL API, για να αποσπαστούν πρώτα τα τριπλά της σύνταξης και μετά αυτά που παραμένουν μπορούν να αντικατασταθούν από το συνοδευτικό λεξιλόγιο. [34]

3.2 Περιγραφή των υπηρεσιών δικτύου χρησιμοποιώντας το OWL-S και την WSDL

3.2.1 Σύνοψη

Η ομάδα του σημασιολογικού δικτύου που ανήκει στο πρόγραμμα DARPA Agent Markup Language αναπτύσσει μία οντολογία για τις υπηρεσίες δικτύου που ονομάζεται OWL-S. Παράλληλα αναπτύσσει μία σειρά από εργαλεία υποστήριξης και τεχνολογίας πρακτόρων για να αυτοματοποιήσει τις υπηρεσίες στο σημασιολογικό δίκτυο. Το OWL-S παρέχει στους προμηθευτές των υπηρεσιών δικτύου ένα βασικό σύνολο δομών της γλώσσας σήμανσης για την περιγραφή των ιδιοτήτων και των δυνατοτήτων των υπηρεσιών τους, με τρόπο σαφές που ο υπολογιστής μπορεί να κατανοήσει και να μεταφράσει. Η σήμανση του OWL-S θα αυτοματοποιήσει τις εργασίες των υπηρεσιών δικτύου συμπεριλαμβάνοντας την εύρεση, την εκτέλεση, την αλληλεπίδραση, τη σύνθεση και τη παρακολούθηση της εκτέλεσης.

Η WSDL είναι τυποποίηση της XML και περιγράφει τις υπηρεσίες δικτύου σαν ένα σύνολο από τελικά σημεία. Χειρίζεται μηνύματα που περιέχουν πληροφορίες εγγράφων ή διαδικασιών. Οι λειτουργίες και τα μηνύματα περιγράφονται αρχικά αποσπασματικά και έπειτα μεταβαίνουν σε ένα πιο ολοκληρωμένο δικτυακό πρωτόκολλο και τύπο μηνύματος, για να ορίσουν ένα τελικό σημείο (υπηρεσίες). Η WSDL είναι επεκτάσιμη, ώστε να επιτρέπει την περιγραφή των τελικών σημείων και των μηνυμάτων τους ανεξάρτητα από το τύπο των μηνυμάτων και το δικτυακό πρωτόκολλο που χρησιμοποιείται στην επικοινωνία.

Τα στοιχεία επέκτασης της WSDL επιτρέπουν τη παράλληλη χρήση της με το OWL-S. Έτσι αυτοί που δημιουργούν υπηρεσίες μπορούν να εκμεταλλευτούν τα πλεονεκτήματα και των δύο προδιαγραφών. Παρακάτω θα δούμε πως πραγματοποιείται ο συνδυασμός του OWL-S και της WSDL εστιάζοντας στο τρόπο που χρησιμοποιείται η WSDL σε αυτό το συνδυασμό.

3.2.2 Εισαγωγή

Η WSDL (Web Services Description Language) ορίζει ένα πρωτόκολλο και ένα μηχανισμό ανεξάρτητης κωδικοποίησης για να μπορούν οι προμηθευτές των υπηρεσιών δικτύου να περιγράφουν τους τρόπους αλληλεπίδρασης με τις προσφερόμενες υπηρεσίες. Η WSDL είναι ένα λεξικό της XML το οποίο περιγράφει τις υπηρεσίες δικτύου και τις χαρτογραφεί ως μια συλλογή τελικών σημείων που επικοινωνούν. Διαχωρίζει τον αφηρημένο ορισμό της υπηρεσίας και των μηνυμάτων από τη ολοκληρωτική τους σύνδεση σε μία δικτυακή θύρα και σε ένα τύπο μηνύματος. Η πρόσφατη προδιαγραφή της WSDL περιγράφει ολοκληρωμένες δεσμεύσεις για το SOAP, HTTP GET/POST, και MIME.

Η OWL-S περιγράφει τις υπηρεσίες δικτύου χρησιμοποιώντας κάποιους όρους. Ο όρος Προφίλ (Profile) περιγράφει τι ακριβώς κάνει η υπηρεσία, το Μοντέλο Διαδικασίας (Process Model) εξηγεί πως λειτουργεί η υπηρεσία και η Υποστήριξη (Grounding) αναλύει τον τρόπο πρόσβασης στην υπηρεσία. Το προφίλ και το μοντέλο διαδικασίας θεωρούνται αφηρημένες προδιαγραφές, με την έννοια ότι δεν ορίζουν τις λεπτομέρειες συγκεκριμένων τύπων μηνυμάτων, πρωτοκόλλων και διευθύνσεων στο δίκτυο. Ο ρόλος της υποστήριξης είναι να παρέχει πιο ολοκληρωμένες λεπτομέρειες. Η WSDL μπορεί να ορίσει αυτές τις λεπτομέρειες και για αυτό έχει κερδίσει το ενδιαφέρον της εμπορικής κοινότητας των υπηρεσιών δικτύου. Έτσι οι δημιουργοί της OWL-S έχουν ορίσει συμφωνίες για τη χρήση της WSDL στην υποστήριξη των υπηρεσιών OWL-S. Αυτές οι συμφωνίες βασίζονται στη παρατήρηση ότι η έννοια της υποστήριξης της OWL-S είναι γενικά σύμφωνη με την έννοια της σύνδεσης της WSDL. Συνεπώς η υποστήριξη μιας ατομικής διαδικασίας της OWL-S γίνεται σχετικά απευθείας. Θα δούμε παρακάτω πως μπορεί αυτό να γίνει βασιζόμενοι στη προδιαγραφή της WSDL 1.1.

Δεδομένου ότι η προκαθορισμένη WSDL προδιαγραφή αναφέρεται στους αρχικούς τύπους δεδομένων της XSD, και οι σύνθετοι τύποι δεδομένων ορίζονται χρησιμοποιώντας την XSD, μία OWL-S/WSDL προδιαγραφή μπορεί επίσης να αναφέρεται σε OWL τάξεις. Αυτές οι τάξεις μπορούν απευθείας να χρησιμοποιηθούν από υπηρεσίες που χρησιμοποιούν WSDL, εφόσον υποστηρίζεται από τους WSDL typing μηχανισμούς. Σε αυτή τη περίπτωση οι τάξεις της OWL μπορούν είτε να οριστούν μέσα στη προδιαγραφή της WSDL, ή σε ένα ξεχωριστό έγγραφο στο οποίο θα αναφέρεται μέσα από την WSDL προδιαγραφή. Παρόλα αυτά δεν είναι

απαραίτητο να δημιουργηθούν WSDL υπηρεσίες που χρησιμοποιούν απευθείας τους τύπους της OWL με αυτό τον τρόπο. Όταν δεν χρησιμοποιούνται απευθείας, οι XSLT μετασχηματισμοί, μπορούν να χρησιμοποιηθούν για να αναλύσουν λεπτομερώς τις σχέσεις ανάμεσα στους τύπους της OWL και τους αντίστοιχους τύπους XML Schema που χρησιμοποιούνται στους ορισμούς της WSDL.

Οι δύο γλώσσες είναι εξίσου απαραίτητες για το πλήρη προσδιορισμό μιας υποστήριξης, επειδή δεν καλύπτουν τον ίδιο εννοιολογικό χώρο. Η WSDL, εξ ορισμού, ορίζει αφηρημένους τύπους χρησιμοποιώντας την XML Schema, ενώ η OWL-S επιτρέπει τον προσδιορισμό τους σαν OWL-S τάξεις. Στο μεταξύ η WSDL/XSD δεν μπορεί να εκφράσει τα σημασιολογικά στοιχεία μιας OWL τάξης. Παρομοίως η OWL δεν μπορεί να εκφράσει τις πληροφορίες σύνδεσης της WSDL. Για αυτό είναι λογικό μία OWL-S/WSDL υποστήριξη να χρησιμοποιεί τάξεις της OWL-S ως αφηρημένους τύπους που αναφέρονται στη WSDL, έπειτα βασίζεται στις δομές σύνδεσης της WSDL για να ορίσει τη διαμόρφωση των μηνυμάτων.

Ένα έγγραφο της WSDL, ορίζει τις υπηρεσίες σαν μία συλλογή από δικτυακά τελικά σημεία ή θύρες. Στην WSDL ο αποσπασματικός ορισμός των τελικών σημείων και των μηνυμάτων διαχωρίζεται από το σταθερό δικτυακό κομμάτι ή από τη διαμόρφωση δεσμευτικών δεδομένων. Αυτό επιτρέπει την επαναχρησιμοποίηση των αφηρημένων ορισμών: τα μηνύματα, που είναι συνοπτικοί ορισμοί των μηνυμάτων που έχουν ανταλλαγή και τις κατηγορίες θύρας που είναι συνοπτικοί ορισμοί των λειτουργιών. Το σταθερό πρωτόκολλο και οι προδιαγραφές διαμόρφωσης δεδομένων για ένα συγκεκριμένο τύπο θύρας συνιστούν μια επαναχρησιμοποιούμενη σύνδεση. Μια θύρα ορίζεται ως η ένωση μιας διεύθυνσης δικτύου με μια επαναχρησιμοποιούμενη σύνδεση, μια συλλογή από θύρες ορίζουν μια υπηρεσία. Για αυτό το λόγο, ένα έγγραφο της WSDL χρησιμοποιεί τα παρακάτω στοιχεία για να ορίσει τις υπηρεσίες δικτύου.

- Τύποι- το σύνολο των ορισμών για τους τύπους δεδομένων χρησιμοποιώντας ένα σύστημα τύπων όπως το XSD.
- Μήνυμα- ένας αποσπασματικός ορισμός των δεδομένων που ανταλλάσσονται.
- Λειτουργία- η αποσπασματική περιγραφή μιας ενέργειας που υποστηρίζεται από την υπηρεσία.
- Τύπος θύρας- ένα αποσπασματικό σύνολο λειτουργιών που υποστηρίζεται από ένα ή περισσότερα τελικά σημεία.

- Σύνδεση- ένα σταθερό πρωτόκολλο και μια προδιαγραφή τυποποίησης δεδομένων για ένα συγκεκριμένο τύπο θύρας.
- Θύρα- ένα μοναδικό τελικό σημείο που ορίζεται ως ο συνδυασμός μιας σύνδεσης και μιας διεύθυνσης δικτύου.
- Υπηρεσία- ένα σύνολο από τελικά σημεία τα οποία σχετίζονται μεταξύ τους.

Πρέπει να σημειώσουμε ότι η WSDL δεν εισάγει ένα καινούργιο τύπο γλώσσας ορισμού. Αναγνωρίζει την ανάγκη για πλούσια συστήματα τύπων που περιγράφουν τη δομή των μηνυμάτων και υποστηρίζουν τη προδιαγραφή XML Schemas specification (XSD) ως κανονικό σύστημα. Ωστόσο είναι παράλογο να περιμένουμε ότι ένα μόνο σύστημα γραμματικής θα περιγράψει όλους τους τύπους μηνυμάτων, για αυτό η WSDL επιτρέπει τη χρήση και άλλων τύπων γλώσσας ορισμού μέσω της επεκτασιμότητας.

Επιπρόσθετα η WSDL ορίζει ένα κοινό μηχανισμό σύνδεσης. Ο μηχανισμός αυτός χρησιμοποιείται για την επισύναψη ενός συγκεκριμένου πρωτοκόλλου ή δομής δεδομένων ή σχεδιαγράμματος σε ένα αφηρημένο μήνυμα, λειτουργία, ή τελικό σημείο. Επιτρέπει την επαναχρησιμοποίηση των αφηρημένων ορισμών.

Μπορούμε να κρατήσουμε την ουσία μιας OWL-S/WSDL βάσης, με τη δημιουργία ενός στιγμιότυπου του OWL-S που περιλαμβάνει όλες τις απαραίτητες πληροφορίες που αφορούν τις σχέσεις ανάμεσα στις σχετικές δομές του OWL-S και της WSDL.

Η χρήση του OWL-S μαζί με τη WSDL περιλαμβάνει την επέκταση του OWL-S στα ακόλουθα στοιχεία της WSDL: τύποι, μηνύματα, λειτουργία και σύνδεση. Συνοπτικά, στους τύπους επιτρέπουμε αλλά δεν απαιτούμε, τον συνυπολογισμό των αυθαίρετων δηλώσεων της OWL. Στο μήνυμα επιτρέπουμε αλλά δεν απαιτούμε την προδιαγραφή των τμημάτων των μηνυμάτων που έχουν τάξεις του OWL-S σαν αποσπασματικούς τύπους. Για το στοιχείο της λειτουργίας προτείνουμε μια νέα ιδιότητα για να φαίνεται ότι υπάρχει επικοινωνία ανάμεσα στη δεδομένη λειτουργία και στην ατομική διαδικασία της OWL-S. Στη σύνδεση δεν επεκτείνουμε ουσιαστικά την WSDL αλλά παρέχουμε μόνο ένα νέο τρόπο κωδικοποίησης που χρησιμοποιείται με τη σύνδεση SOAP της WSDL.

3.2.3 Παράδειγμα για το OWL-S/WSDL

Το παρακάτω παράδειγμα, το οποίο είναι χωρισμένο σε δύο μέρη, μας δείχνει πως ορίζεται η OWLS/ WSDL σε μια απλή on-line αγορά βιβλίου. Η υπηρεσία υποστηρίζει μια λειτουργία η οποία ονομάζεται Congo Buy και η οποία αναπτύσσεται χρησιμοποιώντας το πρωτόκολλο SOAP πάνω στο HTTP.

1^ο Μέρος

Το πρώτο μέρος δίνει τον OWL-S κωδικό για μια απλή ατομική διαδικασία και το συνοδευτικό στιγμιότυπο στήριξης, που σχετίζει αυτό τον κωδικό με τη προδιαγραφή της WSDL, μέσω της οποίας μπορεί να επικοινωνήσει με την υπηρεσία. Πρέπει να σημειώσουμε ότι αυτή η υποστήριξη είναι αρκετά απλή. Πιο συγκεκριμένα δεν επεξηγεί τη χρήση του OWL-S property `xsltTransformation`, που χρησιμοποιείται για να εκφράσει πιο περίπλοκες αντιστοιχίσεις ανάμεσα σε κομμάτια μηνυμάτων WSDL και σε ατομικές διαδικασίες εισαχθέντων και εξαχθέντων δεδομένων.

2^ο Μέρος

Το δεύτερο μέρος παρέχει την προδιαγραφή της WSDL, περιλαμβάνοντας τις δομές που τη σχετίζουν με το κώδικα της OWL-S. Τα σημεία αναφοράς στα δύο έγγραφα έχουν επισημανθεί πιο έντονα.

Παράδειγμα 1^{ου} Μέρους: CongoBuy, Μία Ατομική Διαδικασία Της OWL-S

<http://example.com/congo/CongoBuy.owl>

```
<!DOCTYPE uridef[
  <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns">
  <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema">
  <!ENTITY owl "http://www.w3.org/2002/07/owl">
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema">
  <!ENTITY grounding "http://www.daml.org/services/owl-s/1.1/Grounding.owl">
  <!ENTITY process "http://www.daml.org/services/owl-s/1.1/Process.owl">
  <!ENTITY DEFAULT "http://example.com/congo/CongoBuy.owl">]>
```

```

<rdf:RDF
  xmlns:rdf=      "&rdf;#"
  xmlns:rdfs=    "&rdfs;#"
  xmlns:owl =    "&owl;#"
  xmlns:xsd=     "&xsd;#"
  xmlns:process= "&process;#"
  xmlns:grounding= "&process;#"
  xmlns="&DEFAULT;#">
<owl:Class rdf:ID="SignInData">
  <!-- details omitted -->
</owl:Class>
<!-- The CongoBuy atomic process -->
<process:AtomicProcess rdf:ID="CongoBuy">
  <process:hasInput>
    <process:Input ref:ID="In-BookName">
      <process:parameterType rdf:about="&xsd;#string">
        </process:Input>
    </process:hasInput>
  <process:hasInput>
    <process:Input ref:ID="In-SignInData">
      <process:parameterType rdf:resource="#SignInData">
        </process:Input>
    </process:hasInput>
  <process:hasOutput>
    <process:Output ref:ID="Out-Confirmation">
      <process:parameterType rdf:resource="&xsd;#string">
        </process:Output>
    </process:hasOutput>
  </process:AtomicProcess>
<!-- OWL-S Grounding Instance -->
<grounding:WsdIGrounding rdf:ID="FullCongoBuyGrounding">
  <grounding:hasAtomicProcessGrounding rdf:resource="#CongoBuyGrounding"/>
</grounding:WsdIGrounding>

```

```

<grounding:WsdAtomicProcessGrounding rdf:ID="CongoBuyGrounding">
  <grounding:owlsProcess rdf:resource="#congoBuy">
    <grounding:wSDLOperation>
      <grounding:WsdOperationRef>
        <grounding:portType>
          <xsd:uriReference
rdf:value="http://example.com/congo/congobuy.wsdl#CongoBuyPortType"/>
        </grounding:portType>
        <grounding:operation>
          <xsd:uriReference
rdf:value="http://example.com/congo/congobuy.wsdl#BuyBook"/>
        </grounding:operation>
      </grounding:WsdOperationRef>
    </grounding:wSDLOperation>

    <grounding:wSDLInputMessage
rdf:resource="http://example.com/congo/congobuy.wsdl#CongoBuyInput"/>
    <grounding:wSDLInput>
      <grounding:wSDLInputMessageMap>
        <grounding:owlsParameter rdf:resource="#In-BookName">
          <grounding:wSDLMessagePart>
            <xsd:uriReference
rdf:value="http://example.com/congo/congobuy.wsdl#BookName">
          </grounding:wSDLMessagePart>
        </grounding:wSDLInputMessageMap>
      </grounding:wSDLInput>
      <grounding:wSDLInput>
        <grounding:wSDLInputMessageMap>
          <grounding:owlsParameter rdf:resource="#In-SignInInfo">
            <grounding:wSDLMessagePart>
              <xsd:uriReference
rdf:value="http://example.com/congo/congobuy.wsdl#SignInInfo">
            </grounding:wSDLMessagePart>
          </grounding:wSDLInputMessageMap>
        </grounding:wSDLInput>
      </grounding:wSDLInputMessageMap>
    </grounding:wSDLInput>
  </grounding:owlsProcess>
</grounding:WsdAtomicProcessGrounding>

```

```

</grounding:wSDLInput>
<grounding:wSDLOutputMessage
rdf:resource="http://example.com/congo/congobuy.wSDL#CongoBuyOutput"/>
<grounding:wSDLOutput>
  <grounding:wSDLOutputMessageMap>
    <grounding:owlsParameter rdf:resource="#Out-Confirmation">
      <grounding:wSDLMessagePart>
        <xsd:uriReference
rdf:value="http://example.com/congo/congobuy.wSDL#Confirmation">
          </grounding:wSDLMessagePart>
        </grounding:wSDLOutputMessageMap>
      </grounding:wSDLOutput>

    <grounding:wSDLVersion rdf:resource="http://www.w3.org/TR/2001/NOTE-wSDL-
0010315">
      <grounding:wSDLDocument>
        "http://example.com/congo/congobuy.wSDL"
      </grounding:wSDLDocument>
    </grounding:WSDLGrounding>

```

Παράδειγμα 2^ο μέρους: Η Προδιαγραφή της WSDL παρέχει μια σύνδεση SOAP για το CongoBuy.

<http://example.com/congo/congobuy.wSDL>

```

<?xml version="1.0"?>
<definitions name="CongoBuy"
targetNamespace="http://example.com/congo/congobuy.wSDL"
  xmlns:tns="http://example.com/congo/congobuy.wSDL"
  xmlns:congoOwl="http://example.com/congo/CongoBuy.owl#"
  xmlns:soap="http://schemas.xmlsoap.org/wSDL/soap/"
  xmlns:owl-s-wSDL="http://www.daml.org/services/owl-s/wSDL/"
  xmlns="http://schemas.xmlsoap.org/wSDL/">

```

```

<message name="CongoBuyInput">
  <part name="BookName" owl-s-wsdl:owl-s-parameter="congoOwl:In-
BookName"/>process="congoOwl:CongoBuy"/>

```



```

    <soap:body parts="Confirmation" use="encoded"
      namespace="http://example.com/congo/"
      encodingStyle="http://www.daml.org/2001/03"/>
  </output>
</operation>
</binding>

<service name="CongoBuyService">
  <documentation>My first OWL-S/WSDL service</documentation>
  <port name="CongoBuyPort" binding="tns:CongoBuySoapBinding">
    <soap:address location="http://example.com/congo/"/>
  </port>
</service>
</definitions>

```

3.2.4 Υποστήριξη Υπηρεσιών με την WSDL

3.2.4.1 Η Δομή των WSDL Εγγράφων και τα Στοιχεία Επέκτασιμότητας για το OWL-S

Σε αυτή την ενότητα αναπαράγουμε την WSDL γραμματική προσθέτοντας ένα νέο στοιχείο και δύο καινούργιες ιδιότητες για την χρήση της μαζί με το OWL-S. Τα στοιχεία που έχουμε προσθέσει αναφέρονται με πιο έντονο χρώμα.

```

<wsdl:definitions name="nmtoken"? targetNamespace="uri"?
  xmlns:owl-s-wsdl="http://www.daml.org/services/owl-s/wsdl/">

  <import namespace="uri" location="uri"/>* ...

  <wsdl:documentation.... /> ?

  <wsdl:types> ?
    <wsdl:documentation.... />?

```

```

<xsd:schema.... />*
<!-- extensibility element --> *
<rdf:RDF namespace-declarations...>... </rdf:RDF/>*
</wsdl:types>

<wsdl:message name="nmtoken"> *
  <wsdl:documentation.... />?
  <!-- This spec adds attribute owl-s-parameter -->
  <part name="nmtoken" element="qname"? type="qname"? owl-s-wsdl:owl-s-
parameter="qname"?/> *
</wsdl:message>

<wsdl:portType name="nmtoken">*
  <wsdl:documentation.... />?
  <!-- This spec proposes attribute owl-s-process -->
  <wsdl:operation name="nmtoken" owl-s-wsdl:owl-s-process="qname"?>*
    <wsdl:documentation.... /> ?
    <wsdl:input name="nmtoken"? message="qname">?
      <wsdl:documentation.... /> ?
    </wsdl:input>
    <wsdl:output name="nmtoken"? message="qname">?
      <wsdl:documentation.... /> ?
    </wsdl:output>
    <wsdl:fault name="nmtoken" message="qname"> *
      <wsdl:documentation.... /> ?
    </wsdl:fault>
  </wsdl:operation>
</wsdl:portType>

<wsdl:binding name="nmtoken" type="qname">*
  <wsdl:documentation.... />?
  <!-- extensibility element --> *
  <wsdl:operation name="nmtoken">*
    <wsdl:documentation.... /> ?

```

```

<-- extensibility element --> *
<wsdl:input> ?
  <wsdl:documentation.... /> ?
  <-- extensibility element -->
</wsdl:input>
<wsdl:output> ?
  <wsdl:documentation.... /> ?
  <-- extensibility element --> *
</wsdl:output>
<wsdl:fault name="nmtoken"> *
  <wsdl:documentation.... /> ?
  <-- extensibility element --> *
</wsdl:fault>
</wsdl:operation>
</wsdl:binding>

```

```

<wsdl:service name="nmtoken"> *
  <wsdl:documentation.... />?
  <wsdl:port name="nmtoken" binding="qname"> *
    <wsdl:documentation.... /> ?
    <-- extensibility element -->
  </wsdl:port>
  <-- extensibility element -->
</wsdl:service>

```

```

<-- extensibility element --> *

```

```

</wsdl:definitions>

```

Οι υπηρεσίες ορίζονται χρησιμοποιώντας 6 στοιχεία:

- Τους τύπους, οι οποίοι παρέχουν τους ορισμούς των τύπων των δεδομένων που χρησιμοποιούνται για να περιγράψουν τα μηνύματα που ανταλλάσσονται.

- Το μήνυμα, το οποίο αναπαριστά έναν αφηρημένο ορισμό των δεδομένων που μεταδίδονται. Ένα μήνυμα συνίσταται από λογικά κομμάτια κάθε ένα από το οποία σχετίζεται με ένα λογικό ορισμό μέσα σε κάποιο σύστημα τύπων.
- Το τύπο της θύρας, που είναι ένα σύνολο από αποσπασματικές λειτουργίες. Κάθε λειτουργία αναφέρεται σε ένα μήνυμα που εισέρχεται και στα μηνύματα που εξέρχονται.
- Τη σύνδεση, η οποία ορίζει το σταθερό πρωτόκολλο και τη προδιαγραφή της δομής των δεδομένων για τις λειτουργίες και τα μηνύματα που ορίζονται από ένα συγκεκριμένο port type.
- Τη θύρα, που ορίζει τη διεύθυνση για μια σύνδεση και για αυτό το λόγο ορίζει ένα μοναδικό τελικό σημείο επικοινωνίας.
- Την υπηρεσία, η οποία συναθροίζει το σύνολο των σχετιζόμενων σημείων.

Στην WSDL ο όρος σύνδεση αναφέρεται στο σχετικό πρωτόκολλο διαδικασίας ή στις πληροφορίες δομής δεδομένων με μια αφηρημένη οντότητα όπως αυτή του μηνύματος, της λειτουργίας ή του τύπου της θύρας. Η WSDL επιτρέπει στα στοιχεία να αναπαριστούν μια συγκεκριμένη τεχνολογία (τα οποία αναφέρονται ως επεκτάσιμα στοιχεία) κάτω από διάφορα στοιχεία τα οποία ορίζονται από την WSDL. Αυτά τα σημεία επεκτασιμότητας χρησιμοποιούνται για να ορίσουμε τις πληροφορίες σύνδεσης για ένα συγκεκριμένο πρωτόκολλο ή διαμόρφωση μηνύματος, χωρίς όμως να περιορίζονται για τέτοια χρήση μόνο. Τα στοιχεία επεκτασιμότητας πρέπει να χρησιμοποιούν ένα XML namespace διαφορετικό από της WSDL.

Τα στοιχεία επεκτασιμότητας συνήθως χρησιμοποιούνται για να ορίσουν κάποια τεχνολογία συγκεκριμένης σύνδεσης. Για να μπορούμε να διακρίνουμε κατά πόσο η τεχνολογία της συγκεκριμένης σύνδεσης απαιτείται στην επικοινωνία ή είναι προαιρετική, τα στοιχεία επεκτασιμότητας μπορούν να τοποθετήσουν μια ιδιότητα `wsdl:required`, που θα λαμβάνει λογικές τιμές 0 ή 1, πάνω στο στοιχείο. Η εξ' ορισμού τιμή για το `required` είναι `false`. Η ιδιότητα `required` ορίζεται στο namespace `http://schemas.xmlsoap.org/wsdl/`.

Τα στοιχεία επεκτασιμότητας επιτρέπουν τη καινοτομία στο τομέα των πρωτοκόλλων δικτύου και μηνυμάτων χωρίς να χρειάζεται να επανεξεταστεί η βασική προδιαγραφή της WSDL. Η WSDL θεωρεί ότι οι προδιαγραφές που ορίζουν τέτοια πρωτόκολλα ορίζουν επίσης κάθε απαραίτητη επέκταση της WSDL που χρησιμοποιείται για να περιγράψει αυτά τα πρωτόκολλα ή αυτές τις τυποποιήσεις.

3.2.4.2 Τύποι

Το στοιχείο των τύπων περιλαμβάνει τους ορισμούς των τύπων των δεδομένων που σχετίζονται με τα μηνύματα που ανταλλάσσονται. Για μέγιστη ενδολειτουργικότητα και ουδετερότητα, η WSDL προτιμά τη χρήση της XSD ως σύστημα κανονικών τύπων και το οποίο λαμβάνει ως ουσιαστικό σύστημα τύπων.

```
<definitions.... >  
  <types>  
    <xsd:schema.... /> *  
  </types>  
</definitions>
```

Παρόλα αυτά, εφόσον είναι παράλογο να περιμένουμε από ένα μόνο σύστημα τύπων γραμματικής να μπορεί να χρησιμοποιηθεί στη περιγραφή όλων των αφηρημένων τύπων, τωρινών και μελλοντικών, η WSDL επιτρέπει στα συστήματα τύπων να προστεθούν μέσω των στοιχείων επεκτασιμότητας. Ένα στοιχείο επεκτασιμότητας μπορεί να εμφανίζεται κάτω από τα στοιχεία τύπων για να αναγνωρίσει το σύστημα ορισμού τύπων που χρησιμοποιείται και για να δίνει ένα XML στοιχείο για τους ορισμούς των τύπων. Ο ρόλος αυτού του στοιχείου μπορεί να συγκριθεί με το στοιχείο `schema` της γλώσσας XML Schema.

```
<definitions.... >  
  <types>  
    <!-- type-system extensibility element --> *  
  </types>  
</definitions>
```

Όταν χρησιμοποιούμε το OWL-S με την WSDL, είναι πιθανό να δηλωθούν OWL τάξεις και ιδιότητες μέσα στα τμήματα των τύπων. Όταν ένα τμήμα τύπων χρησιμοποιείται με αυτό τον τρόπο τα στοιχεία επεκτασιμότητας φαίνονται κάπως έτσι:

```

<definitions.... >
  <types>
    <rdf:RDF namespace-declarations... >.... </rdf:RDF>*
  </types>
</definitions>

```

Πιθανότατα η πιο σύνηθες περίπτωση θα ήταν να δηλωθούν όλες οι σχετικές τάξεις σε ένα ξεχωριστό OWL φάκελο όπως φαίνεται στο παράδειγμα του μέρους 1. Αν γίνει αυτό, το τμήμα τύπων της WSDL μπορεί ολοκληρωτικά να παραβλεφθεί.

3.2.4.3 Μηνύματα

Τα μηνύματα περιέχουν ένα ή περισσότερα λογικά κομμάτια. Κάθε κομμάτι σχετίζεται με έναν τύπο από κάποιο σύστημα τύπων χρησιμοποιώντας μία ιδιότητα `message-typing`. Το σύνολο των ιδιοτήτων `message-typing` είναι επεκτάσιμο. Η WSDL ορίζει διάφορες ιδιότητες `message-typing`, οι οποίες μπορούν να χρησιμοποιηθούν μαζί με την XSD.

- Στοιχείο (`element`). Αναφέρεται σε ένα XSD στοιχείο χρησιμοποιώντας ένα QName.
- Τύποι (`type`). Αναφέρεται σε ένα XSD simple Type ή σε ένα complex Type χρησιμοποιώντας ένα QName.

Οι υπόλοιπες `message-typing` ιδιότητες μπορούν να οριστούν εφόσον χρησιμοποιούν ένα namespace διαφορετικό από αυτό της WSDL. Τα στοιχεία επεκτασιμότητας της σύνδεσης μπορούν επίσης να χρησιμοποιήσουν τις ιδιότητες `message-typing`.

Σε αυτή την ενότητα χρησιμοποιούμε την `message-typing` ιδιότητα `owl-s-parameter`, για χρήση με την OWL-S. Αυτό συνήθως θα χρησιμοποιούνταν για να μας δείξει μια υποκατηγορία της κυριότητας του `owl-s:input` ή του `owl-s:output`. Εφόσον γνωρίζουμε τη κυριότητα του `input` και του `output`, ο τύπος του μπορεί εύκολα να αποκτηθεί από το κώδικα της OWL-S.

Το συντακτικό για τον ορισμό ενός μηνύματος είναι ως εξής. Τα message-typing χαρακτηριστικά αναγράφονται πιο έντονα.

```
<definitions.... >  
  <message name="nmtoken"> *  
    <part name="nmtoken" element="qname"? type="qname"?/> *  
  </message>  
</definitions>
```

Η ιδιότητα message name παρέχει ένα μοναδικό όνομα ανάμεσα σε όλα τα μηνύματα που ορίζονται μέσα σε ένα WSDL έγγραφο.

Το κομμάτι name attribute παρέχει ένα μοναδικό όνομα ανάμεσα σε όλα τα κομμάτια του μηνύματος.

Παραδείγματος χάριν, για να ορίσουμε το εισερχόμενο μήνυμα στο Congo Buy γράφουμε αυτό (από το παράδειγμα στο μέρος 1):

```
<definitions.... >  
  <message name="CongoBuyInput">  
    <part name="BookName" owl-s-parameter="congoOwl:In-BookName"/>  
    <part name="SignInInfo" owl-s-parameter="congoDaml:In-SignInInfo"/>  
  </message>  
</definitions>
```

Σημείωση: Το <http://www.daml.org/services/owl-s/wsd/> μπορεί να χρησιμοποιηθεί ως πεδίο ονόματος για την owl-s παράμετρο.

3.2.4.4 Τύποι θύρας

Ο τύπος θύρας είναι ένα ονομαζόμενο σύνολο από αφηρημένες λειτουργίες και αφηρημένα μηνύματα.

```
<wsdl:definitions.... >  
  <wsdl:portType name="nmtoken">  
    <wsdl:operation name="nmtoken".... /> *
```

```
</wsdl:portType>
</wsdl:definitions>
```

Η ιδιότητα `name` του τύπου θύρας παρέχει ένα μοναδικό όνομα ανάμεσα σε όλους τους τύπους θύρας που ορίζονται μέσα σε ένα WSDL έγγραφο. Μια λειτουργία παίρνει το όνομα της μέσω της ιδιότητας `name`.

Η WSDL έχει τέσσερις τρόπους μετάδοσης που μπορεί ένα τελικό σημείο να υποστηρίξει.

A) Μιας κατεύθυνσης. Το τελικό σημείο λαμβάνει ένα μήνυμα.

B) Ερώτηση- απάντηση. Το τελικό σημείο λαμβάνει ένα μήνυμα και στέλνει ένα σχετικό.

Γ) Αίτηση – απάντηση. Το τελικό σημείο στέλνει ένα μήνυμα και λαμβάνει ένα σχετικό.

Δ) Ειδοποίηση. Το τελικό σημείο στέλνει ένα μήνυμα. Η WSDL αναφέρεται σε αυτά τα πρωταρχικά στοιχεία σαν λειτουργίες.

A) Λειτουργία μιας κατεύθυνσης

Μια ατομική διαδικασία που δέχεται εισερχόμενα αλλά δεν έχει εξερχόμενα δεδομένα, θεωρείται λειτουργία της WSDL μιας κατεύθυνσης.

B) Λειτουργία ερωταπόκρισης

Μία OWL-S ατομική διαδικασία με δεδομένα που εισάγονται και εξάγονται, θεωρείται λειτουργία ερωταπόκρισης.

Γ) Λειτουργία αίτησης απόκρισης

Μία σύνθετη διαδικασία με εισερχόμενα και εξερχόμενα δεδομένα. Τα εξερχόμενα δεδομένα που στέλνονται ορίζονται σαν εισερχόμενα πριν από τη παραλαβή των εισερχόμενων. Αυτή η διαδικασία ορίζεται ως λειτουργία αίτησης απόκρισης.

Δ) Λειτουργία ειδοποίησης

Μια ατομική διαδικασία με εξερχόμενα δεδομένα, χωρίς όμως εισερχόμενα, θεωρείται λειτουργία ειδοποίησης της WSDL.

3.2.4.5 Από τις διαδικασίες του OWL-S στις λειτουργίες της WSDL

Χρειάζεται να βρούμε ένα τρόπο όπου θα μπορούμε να δείξουμε την ανταλλαγή που πραγματοποιείται ανάμεσα σε μια WSDL λειτουργία και σε μια ατομική διαδικασία OWL-S. Η WSDL δεν προσφέρει στοιχεία επεκτασιμότητας στις λειτουργίες. Ενώ υπάρχουν τρόποι μέσα στα στοιχεία σύνδεσης της WSDL για να δείξουμε την επικοινωνία, χρησιμοποιώντας για παράδειγμα την ιδιότητα του soapAction, θα ήταν επίσης χρήσιμο να δείξουμε αυτή την επικοινωνία σε αντίθεση με τον ορισμό της λειτουργίας. Για αυτό το λόγο θεωρούμε ότι η WSDL εγκρίνει το προαιρετικό χαρακτηριστικό owl-s-process για το στοιχείο λειτουργίας της WSDL όπως φαίνεται στο παράδειγμα 1:

```
<operation name="BuyBook" owl-s-process="congoOwl: CongoBuy">
```

Σημείωση: η διεύθυνση <http://www.daml.org/services/owl-s/wsdll/> μπορεί να χρησιμοποιηθεί ως πεδίο ονόματος για την διαδικασία OWL-S.

3.2.4.6 Σύνδεση

Όπως με κάθε χρήση της WSDL, υπάρχουν πολλές επιλογές για τη σύνδεση μιας OWL-S/WSDL- υπηρεσίας. Για τους παρόντες στόχους, υλοποιούμε δεσμεύσεις της SOAP, όπως χρησιμοποιούνται στα υπάρχοντα παραδείγματα της WSDL, μόνο που χρησιμοποιούμε την "<http://www.daml.org/2001/03/>" για να δείξουμε το είδος της κωδικοποίησης που έχει κάθε κομμάτι του μηνύματος.[35]

3.3 Παραδείγματα Οντολογιών

3.3.1 Η Ιεραρχία του Προφίλ στο Owl

Αυτό το έγγραφο παρέχει μια βασική οντολογία του προφίλ υπηρεσιών και των προϊόντων για να υποστηρίξει τα παραδείγματα του OWL-S, για την BravoAir και την υπηρεσία πώλησης βιβλίων, Congo. Η οντολογία αυτή δεν θεωρείται ούτε ως «Η οντολογία των υπηρεσιών», ούτε ως «Η επίσημη οντολογία OWL-S των υπηρεσιών». Περισσότερο παρέχει ένα πολύ απλό παράδειγμα στο πώς μπορεί να δημιουργηθεί μια οντολογία και πώς μπορεί να χρησιμοποιηθεί με το προφίλ μιας συγκεκριμένης υπηρεσίας (βλέπε CongoProfile.owl).

Για κάποιες επιπρόσθετες πληροφορίες σχετικά με αυτή τη προσέγγιση μπορούμε να βρούμε στην:

<http://www.daml.org/services/owl-s/1.1/ProfileHierarchy.html>.

Το OWL-S δεν επιβάλλει ότι κάθε προφίλ μπορεί να κατηγοριοποιηθεί σε μία τάξη ιεραρχίας. Οι τάξεις ιεραρχίας του προφίλ παρέχουν τα μέσα για τον καθορισμό πληροφοριών σχετικά με μια υπηρεσία, που μπορεί να είναι χρήσιμες κατά τη διάρκεια της ανάκτησης της υπηρεσίας. Η οντολογία παρακάτω δείχνει πώς η ιεραρχία του προφίλ υπηρεσίας μπορεί να δομηθεί και πώς μπορεί να ενοποιηθεί με την υπόλοιπη εργασία που αναπτύσσεται στο OWL-S. Τελικά, είναι στη δικαιοδοσία των οργανισμών επιχειρήσεων να περιγράψουν την οντολογία των υπηρεσιών που είναι σημαντική για αυτούς. Το OWL-S παρέχει απλά τα εργαλεία για τη κατασκευή και τη χρήση μιας οντολογίας. Στο παρακάτω παράδειγμα, οι υπηρεσίες κατηγοριοποιούνται χρησιμοποιώντας υποκατηγορίες της τάξης του προφίλ www.daml.org/services/owl-s/1.1/Profile.owl, καθορίζοντας κάποιες ιδιόμορφες ιδιότητες τέτοιων υπηρεσιών.

Ειδικότερα περιγράφουμε δύο μεγάλες κατηγορίες υπηρεσιών: α) τις υπηρεσίες πληροφοριών και β) τις υπηρεσίες Ηλεκτρονικού Εμπορίου που πωλούν προϊόντα. Οι υπηρεσίες Πληροφοριών έχουν τρεις ιδιότητες: μια πηγή της πληροφορίας, ένα θέμα και μια χρονική στιγμή της πληροφορίας.

Information_Service

+ source:InformationSource

- + topic:Thing
- + information_date:sri-time-ontology date

Οι υπηρεσίες Ηλεκτρονικού Εμπορίου έχουν μια εμπορική ιδιότητα όσον αφορά το τι πωλούν και το deliveryMode για να καθορίζεται πως θα παραδοθεί το εμπόρευμα.

E_Commerce

- + merchandise:product
- + deliveryMode:Transportation

Καθορίζουμε δύο υποκατηγορίες τάξης των υπηρεσιών του Ηλεκτρονικού Εμπορίου, η μια για την πώληση βιβλίων η οποία μας παρέχει μια ανώτερη τάξη για το παράδειγμα Congo, και μία για τα αεροπορικά εισιτήρια, που παρέχει ένα τρόπο ταξινόμησης του παραδείγματος για την BravoAir.

BookSelling

- + merchandise:Book

AirlineTicketing

- + merchandise:CommercialAirlineTravel

Ο καθορισμός πολλών υπηρεσιών εξαρτάται από τις οντολογίες των προϊόντων. Εδώ καθορίζουμε επίσης το Προϊόν ως το υψηλότερο επίπεδο τάξης ενός απλού προϊόντος. Δεν αποτελεί σκοπό, ο καθορισμός μιας γενικής οντολογίας προϊόντων, αλλά η παροχή των μέσων για την δημιουργία οντολογιών και η χρησιμοποίηση τους για τη διαφήμιση και την παροχή υπηρεσιών. Είναι στη δικαιοδοσία της κοινότητας των επιχειρήσεων να δημιουργήσει τέτοιες οντολογίες ανάλογα με τις ανάγκες και τους στόχους της. Ο δικός μας καθορισμός του προϊόντος αναφέρεται, για λόγους απλοποίησης, στην UNSPSC ταξινόμηση προϊόντων. Το προϊόν είναι το υψηλότερο επίπεδο της οντολογίας Product, μέχρι στιγμής είναι οι μόνες ιδιότητες που αναφέρονται στην UNSPSC οντολογία προϊόντων. Σε αυτή τη τάξη, δείχνουμε πως μια πληροφορία που είναι έξω από το

OWL, μπορεί να χρησιμοποιηθεί για τον καθορισμό πληροφορίας μέσα στις οντολογίες OWL.

Product

+ material:Material

+ UNSPSCcode:number

+ UNSPSCclassification:string

Επίσης καθορίζουμε δύο υποκατηγορίες τάξης του προϊόντος, μία για τα βιβλία και μία CommercialAirTravel, που χρησιμοποιούνται για να ορίσουμε την υπηρεσία πώλησης βιβλίων (BookSelling) και για την υπηρεσία αεροπορικών εισιτηρίων (AirTicketing).

Τέλος, καθορίζουμε τρεις επιπλέον τάξεις: την InformationSource, την ManufacturingProcess, και την Transportation. Αυτές οι τάξεις παρέχονται για να ολοκληρώσουν τις οντολογίες, αλλά όπως υπάρχουν τώρα, είναι απλώς στοιχεία που δεν έχουν ακόμα επεξεργαστεί.

3.3.2 Προφίλ Βασισμένο στις Ιεραρχίες Τάξης

3.3.2.1 Επεξηγηματικές Επισημάνσεις για την Ιεραρχία του Προφίλ του OWL-S

1.1

Όπως αναφέρθηκε στην Τεχνική Περίληψη, ένα Προφίλ Υπηρεσίας χρησιμοποιείται για να χαρακτηρίσει μια υπηρεσία για σκοπούς όπως η διαφήμιση, η εύρεση και η επιλογή. Τα Προφίλ υπηρεσιών δημοσιοποιούνται με μια ποικιλία ειδών εγγραφής, βρίσκονται χρησιμοποιώντας διάφορα εργαλεία και επιλέγονται χρησιμοποιώντας διάφορες τεχνικές ταυτοποίησης. Το OWL-S δεν καθορίζει αλλά ούτε και περιορίζει τους τρόπους με τους οποίους τα προφίλ χρησιμοποιούνται, αλλά περισσότερο παρέχει μια βάση για τη δομή τους, που είναι αρκετά ελαστική ούτως ώστε να προσαρμόσει πολλά διαφορετικά περιεχόμενα και μεθόδους χρήσης.

Γενικά, αυτός ο τύπος χαρακτηρισμού υπηρεσιών πρέπει δραστικά να τοποθετεί την υπηρεσία μέσα σε ένα ευρύ σύνολο υπηρεσιών που υπάρχουν μέσα

σε κάποια κυριότητα ή ακόμα και μέσα στο κόσμο. Μια πολύ φυσιολογική τεχνική για αυτό το είδος τοποθέτησης είναι η δημιουργία μιας ιεραρχικής τάξης, μαζί με την διαδοχή των ιδιοτήτων από τις κατώτερες τάξεις. Αυτή η θεμελιώδης τεχνική, η οποία αποτελεί γνωστό κομμάτι του σχεδιασμού και προγραμματισμού που είναι προσανατολισμένος στο αντικείμενο, υποστηρίζεται αρκετά καλά από το OWL καθώς και από άλλες βασισμένες στη λογική, γλώσσες σήμανσης. Αυτή η τεχνική όταν χρησιμοποιείται, για τη δημιουργία ιεραρχιών στις κατώτερες τάξεις, της τάξης του Προφίλ, παρέχει χρήσιμα μέσα για την δημιουργία ενός στυλ εύρεσης στην κατηγοριοποίηση των υπηρεσιών, αλλά με μια πιο επίσημη δομή από αυτή που χρησιμοποιείται στις «κίτρινες σελίδες», και έτσι υποστηρίζει πιο δυναμικές μορφές ερωτημάτων.

Στην ιεραρχία του προφίλ του OWL δίνεται ένα εξαιρετικά απλό παράδειγμα για αυτού του είδους ιεραρχικής τάξης. Θα πρέπει να τονίσουμε ότι είναι ένα εξαιρετικά περιορισμένο παράδειγμα, που έχει σαν στόχο μόνο να παρέχει τις υποκατηγορίες τάξης του προφίλ στο OWL κάτω από τις οποίες τα παραδείγματα του Congo και της BravoAir μπορούν να κατηγοριοποιηθούν. Ένας άλλος λόγος είναι για να επεξηγήσουν την διαδοχή των ιδιοτήτων και των πολλαπλών διαδοχών. Η ιεραρχία του προφίλ δεν παρέχει τις κανονικές τάξεις του Προφίλ που χρησιμοποιούνται στον Ιστό, αλλά έχει σαν σκοπό να παρουσιάσει μερικώς τα τεχνικά χαρακτηριστικά αυτής της προσέγγισης.

Τελικά, ο σημασιολογικός ιστός πιθανότατα, θα υποστηρίζει ένα σύνολο διαφορετικών ιεραρχιών, βασισμένων στο Προφίλ και αυτές που χρησιμοποιούνται ευρέως θα επιλέγονται από δυνάμεις της αγοράς. Υπάρχουσες ταξινομήσεις των κατηγοριών μιας υπηρεσίας, όπως το NAICS και το UNSPSC, πιθανότατα θα εξελιχθούν σε σχήματα κατηγοριοποίησης των ιεραρχιών τάξης, μαζί με τις κληροδοτημένες ιδιότητες, χρησιμοποιώντας την προσέγγιση που παρουσιάζεται εδώ.

Μαζί με την ιεραρχία τάξης του προφίλ υπηρεσίας, προσδοκούμε ότι πολλά σχήματα κατηγοριοποίησης της ιεραρχίας τάξεων, θα χρησιμοποιηθούν σε σχέση με τα προφίλ υπηρεσιών του OWL-S. Παραδείγματος χάριν ιεραρχίες προϊόντων, γεωγραφικά περιοχές, M.M.E ή άλλες κατηγορίες που σχετίζονται με διαφορετικούς τύπους υπηρεσιών. Τυπικά, σε πολλές ιδιότητες ενός στιγμιότυπου του Προφίλ, θα δίνονται διάφορες τιμές που θα προκύπτουν από αυτές τις ιεραρχίες. Παραδείγματος χάριν, μια ιστοσελίδα B2C μπορεί να χαρακτηρίσει τις προσφορές των προϊόντων

χρησιμοποιώντας όρους (τάξεις ή στιγμιότυπα) από μια ιεραρχία προϊόντος και τη περιοχή αποστολής χρησιμοποιώντας όρους από μια γεωγραφική ιεραρχία. Οι βοηθητικές τάξεις ιεραρχίας βρίσκονται έξω από το αντικείμενο του OWL-S, αλλά θα καθορίζονται χρησιμοποιώντας το OWL, και έτσι θα είναι αυτομάτως συμβατές με μία OWL-S προδιαγραφή. Δεν καθορίζουμε καμία τάξη ιεραρχίας εδώ, εκτός από ένα μικρό απόσπασμα της ιεραρχίας ενός προϊόντος, που χρησιμοποιείται σε αντιδιαστολή με το παράδειγμα του Congo.

Παρόλο που η προσέγγιση του προφίλ, που βασίζεται στην ιεραρχική τάξη του επιπέδου, όπως παρουσιάζονται εδώ είναι ένας ελπιδοφόρος και με πολλά πλεονεκτήματα τρόπος για τη χρησιμοποίηση του προφίλ, δεν είναι η μόνη χρήσιμη προσέγγιση και το OWL-S δεν επιτάσσει ότι κάθε προφίλ θα πρέπει να ταξινομείται βάσει της ιεραρχίας τάξεων. Είναι όμως πιθανό, να καθορίζεται το προφίλ μιας υπηρεσίας ως στιγμιότυπο της τάξης του προφίλ, αποφεύγοντας τη χρήση της ιεραρχικής κατηγοριοποίησης. Η τελευταία προσέγγιση είναι κατάλληλη όταν μια υπηρεσία δεν μπορεί να κατηγοριοποιηθεί από κανένα οντολογικό σχήμα, ή όταν η ιεραρχική ταξινόμηση θα μεταβίβαζε ακατάλληλες συμπεράσματα που ο δημιουργός του προφίλ θα προτιμούσε να αποφύγει. Τελικά, η ιεραρχικός ταξινόμηση του προφίλ παρέχει ένα πρόσθετο εργαλείο για την πιο ακριβή περιγραφή μιας υπηρεσίας και των λειτουργιών της. Είναι στη δικαιοδοσία των προμηθευτών της υπηρεσίας που δημιουργούν το προφίλ, να αποφασίσουν αν θα το εκμεταλλευτούν αυτό και πως.[36]

3.4 Η Σχέση του OWL-S με Άλλες Τεχνολογίες

3.4.1 Εισαγωγή

Σε αυτό το κομμάτι θα σχολιάσουμε συνοπτικά τη σχέση του OWL-S με επιλεγμένες τεχνολογίες υπηρεσιών δικτύου καθώς και με τεχνολογίες του σημασιολογικού δικτύου. Υπάρχουν πολλές τεχνολογίες που σχετίζονται με το OWL-S, εμείς θα αναφερθούμε στις εξής:

- WSDL
- SOAP
- UDDI

- BPEL4WS
- CDL
- ebXML
- Grid services
- OWL and SWRL
- WSMO

Το OWL-S, αναπτύσσεται από το 2001 και έχει χρησιμοποιηθεί σε αρκετές ερευνητικές προσπάθειες. Σε αρκετές περιπτώσεις έχουν προταθεί διάφορα πρωτότυπα συστήματα ή συστήματα σχεδιασμού ή ένα σύνολο από εργαλεία, με σκοπό την υποστήριξή του.

3.4.2 WSDL

Η WSDL, περιγράφει τις υπηρεσίες σαν μια συλλογή από τελικά σημεία τα οποία επικοινωνούν μεταξύ τους ή σαν μια συλλογή από θύρες που στέλνουν και λαμβάνουν μηνύματα με βάση συγκεκριμένα πρωτόκολλα, όπως το HTTP, ή το SOAP-RPC. Η WSDL στοχεύει στην αυτοματοποίηση της επικοινωνίας ανάμεσα στις υπηρεσίες δικτύου διαχωρίζοντας τις αφηρημένες περιγραφές των υπηρεσιών από την ολοκληρωμένη μορφή των δεδομένων και των πρωτοκόλλων, που χρησιμοποιούνται για την υλοποίηση των υπηρεσιών δικτύου.

Οι υπηρεσίες δικτύου ορίζονται σαν σύνολο από θύρες, που ουσιαστικά είναι διευθύνσεις στο δίκτυο, οι οποίες σχετίζονται με συγκεκριμένα πρωτόκολλα και συγκεκριμένες προδιαγραφές τυποποίησης δεδομένων. Κάθε θύρα σχετίζεται με ένα τύπο θύρας, που περιγράφει τις λειτουργίες ανταλλαγής μηνυμάτων στις οποίες λαμβάνει μέρος η θύρα. Υπάρχουν 4 βασικά είδη λειτουργιών, τα οποία είναι πιθανά: μήνυμα προς μια κατεύθυνση, μία δύο κατευθύνσεων ερώτηση- απάντηση, μία δύο κατευθύνσεων αίτηση- απάντηση και ένα μιας κατεύθυνσης μήνυμα ειδοποίησης. Για να ορίσουμε τα μηνύματα, χρησιμοποιούνται συνήθως οι τύποι της XML Schema έτσι καλύπτεται ένα μεγάλο εύρος από ορισμούς τύπων. Αντιθέτως το OWL-S ορίζει τους τύπους των μηνυμάτων σε όρους τάξεων του OWL-S, επιτρέποντας μια πλουσιότερη σημασιολογικά υποδομή, ιεραρχημένη με τάξεις, υπογραμμίζοντας τις προδιαγραφές

των τύπων. Οι τύποι θύρας είναι επαναχρησιμοποιήσιμοι και μπορούν να δεσμευτούν με πολλαπλές θύρες.

Η WSDL έχει ως δεδομένο ένα μοντέλο πελάτη- εξυπηρετητή, με συγχρονισμένες ή με ασυσχέτιστες ασύγχρονες αλληλεπιδράσεις. Αντιθέτως το OWL-S υποστηρίζει μια πλουσιότερη σημασιολογικά περιγραφή των υπηρεσιών δικτύου πέρα από την επιφάνεια επικοινωνίας τους, και προϋποθέτει μια μακροπρόθεσμη συμπεριφορά των υπηρεσιών δικτύου ενταγμένη σε αυτό. Π.χ ορίζοντας λογικούς περιορισμούς ανάμεσα στις παραμέτρους υπηρεσιών που εισάγονται και εξάγονται.

Το OWL-S ουσιαστικά συμπληρώνει την WSDL και η οντολογία υποστήριξης της OWL-S, παρέχει τα μέσα για να μπορούν να χρησιμοποιηθούν και οι δύο μαζί. Οι διαδικασίες του OWL-S φαίνονται ως αφηρημένες προδιαγραφές υπηρεσιών, με την έννοια ότι δεν περιέχουν συνδέσεις σχετικά με τις λεπτομέρειες της αλληλεπίδρασης, όπως η διαμόρφωση μηνύματος και το πρωτόκολλο επικοινωνίας. Η οντολογία υποστήριξης χρησιμοποιείται για να ορίσει αυτές τις δεσμεύσεις, αντιστοιχίζοντας επιλεγμένα στοιχεία της οντολογίας OWL-S με επιλεγμένα στοιχεία μιας WSDL προδιαγραφής.

Εν συντομία αυτές οι αντιστοιχίσεις βασίζονται στη φυσική επικοινωνία που υπάρχει ανάμεσα στο OWL-S και τη WSDL:

- Μία ατομική διαδικασία OWL-S αντιστοιχεί σε μια WSDL λειτουργία
- Έχουμε το σύνολο των δεδομένων που εισάγονται και το σύνολο των δεδομένων που εξάγονται από μια OWL-S ατομική διαδικασία. Κάθε ένα αντιστοιχεί στην WSDL έννοια ενός μηνύματος.
- Οι τύποι (OWL-S τάξεις) των δεδομένων που εισάγονται και εξάγονται από μια OWL-S ατομική διαδικασία αντιστοιχούν στην εκτεταμένη έννοια της WSDL των αφηρημένων τύπων.

3.4.3 SOAP (Simple Object Access Protocol)

Το SOAP παρέχει ένα βασικό πλαίσιο εργασίας μηνυμάτων, για να μπορούν οι υπηρεσίες δικτύου να ανταλλάσσουν μηνύματα. Χρησιμοποιεί τη δομή της XML για τα μηνύματα που ανταλλάσσουν οι υπηρεσίες δικτύου. Ο μηχανισμός ανταλλαγής μηνυμάτων είναι αρχικά μιας κατεύθυνσης, παρόλο που μπορούν να προστεθούν

πιο περίπλοκοι μηχανισμοί πάνω σε αυτόν. Επιπρόσθετα με το SOAP μπορούμε να περιγράψουμε τις ενέργειες που πρέπει να πραγματοποιήσει ένα τελικό σημείο για να είναι σε θέση να λάβει ένα συγκεκριμένο μήνυμα SOAP.

Το SOAP αποτελεί ένα βασικό στοιχείο των υπηρεσιών δικτύου και για αυτό το λόγο αποτελεί κοινή βάση για διάφορα πρότυπα υπηρεσιών δικτύου, πιο συγκεκριμένα για το WSDL. Η OWL-S υποστηρίζει την ανταλλαγή SOAP μηνυμάτων ανάμεσα στις υπηρεσίες δικτύου, χρησιμοποιώντας την WSDL για τον ορισμό της δομής των μηνυμάτων που ανταλλάσσονται.

3.4.4 UDDI (Universal Description, Discovery and Integration)

Το UDDI αποτελεί ένα πρωτόκολλο καταχώρησης υπηρεσιών δικτύου, που επιτρέπει στους προγραμματιστές και στους υπόλοιπους αντιπροσώπους μιας επιχείρησης να εντοπίζουν επιχειρησιακούς συνεργάτες, έχοντας σαν βάση τις υπηρεσίες που παρέχουν.

Κάθε επιχειρησιακή περιγραφή σε UDDI περιλαμβάνει ένα στοιχείο `businessEntity`. Το στοιχείο αυτό περιγράφει μια υπηρεσία αναφέροντας το όνομα, μια τιμή κλειδί και πολλαπλά στοιχεία `bindingTemplate`. Το στοιχείο αυτό, με τη σειρά του, περιγράφει το είδος της πρόσβασης που απαιτεί μια υπηρεσία (τηλέφωνο, mail, http, fax κ.τ.λ.), τιμές-κλειδιά και τα στοιχεία `tModelInstances`. Τα τελευταία χρησιμοποιούνται για να περιγράψουν τις δομές των πρωτοκόλλων ανταλλαγής που κατανοεί η υπηρεσία, δηλαδή η τεχνική πληροφορία που απαιτείται για την πρόσβαση σε μια υπηρεσία. Χρησιμοποιούνται επίσης στη περιγραφή των namespaces για τις ταξινομήσεις που χρησιμοποιούνται στην κατηγοριοποίηση. Πολλά από τα στοιχεία είναι προαιρετικά συμπεριλαμβανομένων και των στοιχείων που θα απαιτούνταν στη ταυτοποίηση ή στην εύρεση υπηρεσιών.

Χρησιμοποιώντας το UDDI ο παροχέας μιας υπηρεσίας δικτύου καταχωρεί διαφημίσεις καθώς και λέξεις κλειδιά που χρησιμοποιούνται στην κατηγοριοποίηση. Ο χρήστης υπηρεσιών δικτύου μπορεί να κάνει αναζήτηση χρησιμοποιώντας λέξεις-κλειδιά. Η χρήση του OWL αναμένεται ότι θα διευκολύνει την ανάκτηση υπηρεσιών μέσω των καταχωρήσεων.

Ενώ η παρουσίαση των υπηρεσιών δικτύου που παρέχονται από το UDDI και το OWL-S διαφέρει αρκετά και οι δύο αυτές προσπάθειες στοχεύουν στη

διευκόλυνση της εύρεσης υπηρεσιών δικτύου. Έχει προταθεί ένας τρόπος για να συνδυαστούν οι δύο προσπάθειες, ο οποίος ορίζει την αντιστοίχιση της OWL-S με τη UDDI.

3.4.5 BPEL4WS

Η BPEL4WS (Business Process Execution Language for Web Services) ενεργοποιεί τις προδιαγραφές των εκτελέσιμων επιχειρησιακών διαδικασιών, συμπεριλαμβάνοντας και τις υπηρεσίες δικτύου, καθώς και τα πρωτόκολλα επιχειρησιακών διαδικασιών, σε όρους της δικής τους λογικής εκτέλεσης ή ροής ελέγχου. Οι επιχειρησιακές διαδικασίες που εκτελούνται ορίζουν τη συμπεριφορά όσων συμμετέχουν και αλληλεπιδρούν με τις υπηρεσίες. Επιπλέον η επίκληση των υπηρεσιών είναι άμεση, εφόσον τα πρωτόκολλα επιχειρηματικών διαδικασιών αποσπούν στοιχεία από την εσωτερική συμπεριφορά για να περιγράψουν τα μηνύματα που ανταλλάσσονται από τις διάφορες υπηρεσίες δικτύου κατά τη διάρκεια μιας αλληλεπίδρασης.

Οι αφηρημένες διαδικασίες λαμβάνουν υπόψη μόνο τα δεδομένα που έχουν σχέση με το πρωτόκολλο ενώ αγνοούν τα δεδομένα που σχετίζονται με τις εσωτερικές διαδικασίες και το προγραμματισμό. Οι επιδράσεις του προγραμματισμού πάνω στο επιχειρησιακό πρωτόκολλο περιγράφονται μετά χρησιμοποιώντας μη ντετερμινιστικές τιμές δεδομένων. Από την άλλη πλευρά οι εκτελέσιμες διαδικασίες περιγράφονται με τη χρήση μιας πλούσιας γλώσσας περιγραφής διαδικασιών, η οποία ασχολείται τόσο με τα δεδομένα που σχετίζονται με το πρωτόκολλο όσο και με τα δεδομένα των εσωτερικών διαδικασιών. Η BPEL4WS ορίζει επίσης διάφορους μηχανισμούς για την ανάνηψη από λάθη καθώς και χειρισμούς επανόρθωσης, που ορίζουν τις επανορθωτικές κινήσεις σε ενέργειες που δεν μπορούν με σαφήνεια να αναιρεθούν.

Πρόσφατες εργασίες έχουν δείξει ότι η BPEL4WS μπορεί, χρησιμοποιώντας τις περιγραφές υπηρεσιών του OWL-S, να αυξήσει τη λειτουργικότητα της συμπεριλαμβάνοντας εργασίες όπως η σύνδεση με δυναμικούς συνεργάτες και η σημασιολογική ενοποίηση. Το OWL-S έχει σαν στόχο να κάνει τις υπηρεσίες δικτύου κατανοητές από τους Η/Υ, να ενεργοποιήσει τον αυτοματισμό καθώς και μια σειρά ενεργειών όπως η εύρεση, η επίκληση και η σύνθεση. Η BPEL4WS δεν έχει τέτοιο

στόχο. Παρέχει μια γλώσσα για την περιγραφή εκτελέσιμων διαδικασιών και πρωτοκόλλων.

Η BPEL4WS είναι ουσιαστικά μια γλώσσα μοντελοποίησης διαδικασιών. Συνεπώς υπάρχει μια έντονη συσχέτιση ανάμεσα σε αυτή και το μοντέλο διαδικασίας της OWL-S.

3.4.6 ebXML

Η ebXML αναφέρεται στο γενικό πρόβλημα της αλληλεπίδρασης των B2B εργασιών, από την προοπτική του πλαισίου εργασίας. Οι αλληλεπιδράσεις των επιχειρησιακών ενεργειών μπορούν να περιγραφούν με δύο τρόπους: από τη πλευρά των επιχειρησιακών λειτουργιών, Business Operational View (BOV) και από την πλευρά των λειτουργικών υπηρεσιών, Functional Service View (FSV)[37]. Η BOV ασχολείται με τα σημασιολογικά στοιχεία στις συναλλαγές των επιχειρησιακών δεδομένων, τα οποία περιέχουν συμβάσεις λειτουργιών, συμφωνίες, αμοιβαίες υποχρεώσεις και οτιδήποτε άλλο σχετικό με επιχειρήσεις. Η FSV ασχολείται με τις υποστηρικτικές επιχειρήσεις, πιο συγκεκριμένα με τις δυνατότητες τους, την επιφάνεια λειτουργίας τους και τα πρωτόκολλα που χρησιμοποιούν.

Το ebXML καθιστά ικανές τις υπηρεσίες δικτύου να περιγράφουν τις επιχειρηματικές διαδικασίες που υποστηρίζουν και τις υπηρεσίες που προσφέρουν χρησιμοποιώντας το Collaboration Protocol Profiles (CPP). Μια επιχειρηματική διαδικασία στο ebXML θεωρείται ως ένα σύνολο από ανταλλαγές επιχειρησιακών εγγράφων ανάμεσα σε ένα σύνολο από υπηρεσίες δικτύου. Μπορεί να θεωρηθεί παρόμοιο με την ανταλλαγή μηνυμάτων ανάμεσα στις υπηρεσίες, όπως συνήθως περιγράφεται στα πρότυπα των υπηρεσιών. Το CPP περιλαμβάνει την ταξινόμηση των επιχειρήσεων, τις πληροφορίες για τις επαφές, τις υποστηριζόμενες επιχειρηματικές διαδικασίες κ.α. Είναι καταχωρημένες με τέτοιο τρόπο ώστε να μπορούν να βρεθούν και από άλλες υπηρεσίες. Ωστόσο το πεδίο δράσης της ebXML δεν επεκτείνεται στο τομέα της μορφοποίησης, όπου ορίζονται τα επιχειρησιακά έγγραφα. Αυτό αφήνεται στις αλληλεπιδραστικές υπηρεσίες δικτύου για να έλθουν σε συμφωνία εκ των προτέρων με τη δημιουργία ενός Πρωτοκόλλου Συνεργασίας. Εφόσον η OWL-S παρέχει τη γλώσσα για την περιγραφή της συμπεριφοράς των υπηρεσιών δικτύου, σκοπός του Collaboration Protocol Agreement είναι να

συμπληρώσει την ebXML. Πιο συγκεκριμένα οι περιγραφές της OWL-S θα μπορούσαν να χρησιμοποιηθούν με την ebXML για να περιγράψουν τις επιχειρηματικές διαδικασίες των υπηρεσιών που αλληλεπιδρούν.

3.4.7 CDL (Choreography Description Language)

Η CDL περιγράφει τις αλληλεπιδράσεις των υπηρεσιών δικτύου σε όρους εξωτερικής συμπεριφοράς, η οποία εκτίθεται τυπικά μέσα από την ανταλλαγή μηνυμάτων. Κάθε ένας που συμμετέχει στην αλληλεπίδραση ορίζει μια κοινή επιφάνεια, περιγράφοντας τη προσωρινή σειρά και τη λογική εξάρτηση των μηνυμάτων που στέλνει και λαμβάνει. Επιπρόσθετα, μπορεί να οριστεί ένα παγκόσμιο μοντέλο, που θα περιγράφει τα συλλογικά μηνύματα που ανταλλάσσονται από την αλληλεπίδραση των υπηρεσιών δικτύου. Αντίθετα με τη BPEL4WS, η CDL δεν περιγράφει τις λεπτομέρειες εκτέλεσης των ατομικών υπηρεσιών δικτύου. Εστιάζει περισσότερο στο πρόβλημα της συνεργασίας (ανταλλαγή μηνυμάτων ανάμεσα σε διανεμημένα peers) παρά στην ενορχήστρωση (δημιουργία εκτελέσιμων υπηρεσιών δικτύου). Συνεπώς δεν θεωρεί δεδομένη τη παρουσία μιας κεντρικής διαχείρισης των αλληλεπιδράσεων ανάμεσα στις υπηρεσίες δικτύου. Για τον ίδιο λόγο οι δομές ελέγχου ροής είναι εξαιρετικά πιο απλές από την BPEL4WS. Τα έγγραφα της CDL είναι επίσης προδιαγραφές που στοχεύουν στη διασαφήνιση, στην αυτόματη επικύρωση, παρόλο που μπορούν να χρησιμοποιηθούν για την αυτόματη παραγωγή κωδικών σκελετού.

Η CDL βρίσκεται πιο κοντά στο Μοντέλο Διαδικασίας της OWL-S και οι δύο μοιράζονται το στόχο της περιγραφής των μηνυμάτων που ανταλλάσσονται ανάμεσα στις υπηρεσίες δικτύου που συμμετέχουν. Παρόλα αυτά αντίθετα με την CDL, η σήμανση της OWL-S στοχεύει στην υποστήριξη της εκτέλεσης και της δημιουργίας συνθέσεων, εκτελέσιμων υπηρεσιών. Επιπλέον η CDL αποφεύγει κάθε περιγραφή επιχειρηματικής σπουδαιότητας ενώ η OWL-S έχει ως σκοπό να υποστηρίξει την περιγραφή όλων των στοιχείων, από τις προϋποθέσεις του «πραγματικού κόσμου» και τα αποτελέσματα της κλήσης μιας υπηρεσίας έως και την ταξινόμηση των υπηρεσιών.

3.4.8 OWL και SWRL

Η OWL και SWRL είναι ομαδοποιημένες, λόγω του ότι είναι και οι δύο γλώσσες που χρησιμοποιεί η OWL-S ώστε να μπορεί να «χτίζει» πάνω σε ανερχόμενα πρότυπα γλώσσας. Η γλώσσα οντολογίας OWL αποτελεί σύσταση της W3C και είναι αντιπροσωπευτική γλώσσα για OWL-S οντολογίες. Ένας λόγος που επιλέχτηκε η OWL σαν αντιπροσωπευτική γλώσσα για την OWL-S, είναι ότι είναι συμβατή με την XML και την RDF, ενώ ταυτόχρονα προσφέρει επιπρόσθετη εκφραστικότητα επιτρέποντας στους χρήστες να περιγράψουν περισσότερες κατηγορίες τάξεων, ιδιότητες και σχέσεις από ότι η XML και η RDF. Η OWL προσφέρει επίσης μια τυπική σημασιολογία, για αυτό το λόγο στους όρους που ορίζονται με την OWL δίνεται μια ακριβής έννοια και μπορούν να χρησιμοποιηθούν αποτελεσματικά σε εφαρμογές που απαιτούν ενδολειτουργική ικανότητα. Είναι μια γλώσσα αναπαράστασης που δεν παρέχει πλούσιο λεξιλόγιο για εφαρμογές υπηρεσιών. Για αυτό όποιος χρειαζόταν να δημιουργήσει μια υπηρεσία θα έπρεπε να βρει μια οντολογία υπηρεσιών ή να φτιάξει μια δική του. Η OWL-S είναι ένα σύνολο από αλληλοσχετιζόμενες OWL οντολογίες και παρέχει ένα σύνολο από όρους που μπορούν να χρησιμοποιηθούν από εφαρμογές υπηρεσιών. Αυτές οι οντολογίες καθορίζουν όρους που είναι οικείοι στα προφίλ υπηρεσιών, στα μοντέλα διαδικασίας και στον ενδοχειρισμό των υπηρεσιών. Παρέχουν στους χρήστες του σημασιολογικού δικτύου το λεξικό των τάξεων, σχέσεις και στιγμιότυπα, που χρησιμοποιούνται σε συμβατές εφαρμογές της OWL, RDF και XML.

Η OWL περιλαμβάνει τρεις υποκατηγορίες γλωσσών: την OWL Lite, την OWL DL και την OWL Full. Η OWL Lite έχει μικρότερη εκφραστική δύναμη από την OWL DL και την OWL Full και σε γενικές γραμμές δεν κάλυπτε τις ανάγκες της OWL-S. Η OWL DL παρέχει μέγιστη εκφραστικότητα, ενώ παράλληλα διατηρεί ολοκληρωμένη υπολογιστική ικανότητα, για αυτό και χρησιμοποιείται σε εφαρμογές όπου απαιτείται αιτιολογική υποστήριξη. Η OWL DL και η OWL Full έχουν την ίδια δομή γλώσσας μόνο που η OWL DL βάζει κάποιους περιορισμούς στη χρήση. Οι οντολογίες της OWL-S είναι γραμμένες σε OWL-DL.

Οι χρήστες του OWL-S μπορούν να χρησιμοποιήσουν διάφορα εργαλεία όπως μηχανισμούς επικύρωσης, φυλλομετρητές, περιβάλλοντα ανάπτυξης οντολογιών κ.τ.λ. Πολλά από τα εργαλεία αυτά μπορούμε να βρούμε στην ιστοσελίδα της DAML και στη SemWebCentral. Η SWRL Semantic Web Rule Language είναι μια

γλώσσα κανόνων η οποία συνδυάζει το OWL με μια γλώσσα κανόνων συμβατή με το OWL. Η SWRL έχει υποβληθεί στο W3C ως μια πιθανή αρχή για νέες γλώσσες κανόνων για το σημασιολογικό δίκτυο. Οι εκφράσεις της SWRL μπορούν να χρησιμοποιηθούν ως προϋποθέσεις, ως όροι για τον έλεγχο διαδικασιών, καθώς και σε εκφράσεις αποτελεσμάτων. Το γεγονός ότι η SWRL συνδέεται με την OWL-S καθιστά τις οντολογίες της OWL-S ισχυρές γιατί χρησιμοποιείται η εκφραστική δύναμη των κανόνων σε ένα ανερχόμενο πρότυπο.

3.4.9 Grid Services

Το Grid είναι ένα σύστημα το οποίο είναι σχεδιασμένο για να συντονίζει τις πηγές που δεν υπάγονται σε ένα κεντρικό έλεγχο χρησιμοποιώντας πρότυπα, διάφορα πρωτόκολλα εξασφαλίζοντας έτσι την ποιότητα των υπηρεσιών. Οι πηγές που είναι διαθέσιμες σε ένα Grid είναι μοντελοποιημένες ως Grid υπηρεσίες. Το Open Grid Infrastructure (OGSI) [38], χρησιμοποιεί εκτεταμένα τις δομές της WSDL και την XML Schema για να εισάγει την αντίληψη των καταχωρημένων υπηρεσιών δικτύου μαζί με τα πρότυπα λειτουργίας για τη δημιουργία και τη διαγραφή υπηρεσιών δικτύου, για την παρουσίαση και την ενημέρωση των μεταδεδομένων που σχετίζονται με τις υπηρεσίες δικτύου. Επιπλέον η OGSI παρέχει μηχανισμούς αναφορών σε στιγμιότυπα των υπηρεσιών δικτύου καθώς και ασύγχρονες ενημερώσεις για αλλαγές στην κατάσταση των υπηρεσιών δικτύου. Επιπλέον ορίζει την XML Schema για τη περιγραφή λαθών που λαμβάνουν χώρα κατά τη διάρκεια λειτουργίας της WSDL.

Το πλαίσιο εργασίας OGSI έχει πρόσφατα ανακατασκευαστεί για να ορίσει ένα σύνολο από σχετικές προδιαγραφές που θα μπορούσαν να υιοθετηθούν σε αποσπασματική βάση. Αυτή η ανακατασκευή μαζί με τις επεκτάσεις που γίνονται ώστε να συμβαδίζει με τις αλλαγές των προτύπων των υπηρεσιών δικτύου, δημιουργεί το WS-Resource Framework (WSRF). Παρόλο που το OGSI και το WSRF ξεπερνούν την WSDL με τη προδιαγραφή διαφόρων ειδών των ειδικών τύπων θύρας και των μηνυμάτων σφάλματος, σε αντίθεση με την OWL-S, δεν περιγράφουν τη συμπεριφορά του Grid ή των υπηρεσιών δικτύου. Με αυτή την έννοια η OWL-S συμπληρώνει και τις δύο προδιαγραφές.

3.4.10 WSMO

Η WSMO μοιράζεται με την OWL-S την ιδέα ότι οι οντολογίες είναι σημαντικές στη υποστήριξη της αυτόματης εύρεσης, σύνθεσης και ενδολειτουργίας των υπηρεσιών δικτύου. Παρά το ενιαίο όραμα που μοιράζονται διαφέρουν σημαντικά στις λεπτομέρειες και στο τρόπο προσέγγισης για την επίτευξη αυτών των αποτελεσμάτων. Ενώ η OWL-S ορίζει με σαφήνεια ένα σύνολο από οντολογίες που υποστηρίζουν την αιτιολόγηση για τις υπηρεσίες δικτύου, η WSMO ορίζει ένα θεμελιώδες πλαίσιο εργασίας μέσα στο οποίο θα πρέπει να δημιουργηθούν οι οντολογίες. Μια ακόμη διαφορά ανάμεσα στις δύο είναι ότι ενώ η OWL-S δεν κάνει καμία διάκριση ανάμεσα στους τύπους των υπηρεσιών δικτύου, η WSMO δίνει έμφαση στις προδιαγραφές των μεσολαβητών: στη χαρτογράφηση προγραμμάτων που λύνουν τα προβλήματα ενδολειτουργικότητας ανάμεσα στις υπηρεσίες δικτύου.

Στη WSMO οι μεσολαβητές εκτελούν εργασίες όπως η μετάφραση μεταξύ των οντολογιών ή ανάμεσα στα μηνύματα που παράγει η μια υπηρεσία και σε αυτά που μια άλλη αναμένει. Κατά τη διαδικασία ορισμού μεσολαβητών, η WSMO δίνει τους πιθανούς μεσολαβητές που βοηθούν στον ορισμό και τη ταξινόμηση των διαφορετικών εργασιών που οι μεσολαβητές πρέπει να λύσουν. Παρόλα αυτά είναι σχετικά δύσκολο να δημιουργηθεί μια ταξινόμηση πάνω στα κλασσικά προβλήματα του ενδοχειρισμού των υπηρεσιών δικτύου, όπως η εύρεση, η σύνθεση και η επίκληση. Για παράδειγμα δεν είναι ξεκάθαρο το πώς οι μεσολαβητές μπορούν να βοηθήσουν κατά τη διάρκεια της εύρεσης, εφόσον η εύρεση είναι ουσιαστικά πρόβλημα επιλογής, ενώ οι μεσολαβητές προσπαθούν να διευθετήσουν τις διαφορές ανάμεσα στους στόχους των υπηρεσιών δικτύου.

Ο ορισμός των μεσολαβητών στο WSMO απαιτεί προσοχή σε ορισμένες σημαντικές εργασίες μετάφρασης που αντιμετωπίζουν οι υπηρεσίες δικτύου. Αυτές οι εργασίες είναι απαραίτητες γιατί υποστηρίζουν την αλληλεπίδραση των υπηρεσιών δικτύου. Παρόλα αυτά η OWL-S παρέχει στις υπηρεσίες δικτύου και στους πελάτες της τις πληροφορίες που είναι απαραίτητες για να βρει εκείνους τους μεσολαβητές που μπορούν να διαρρυθμίσουν τις ανεπιτυχείς ταυτοποιήσεις, ή να δημιουργήσουν μεσολαβητές μέσα από τη διαδικασία της σύνθεσης.[39]

4 ΚΕΦΑΛΑΙΟ: ΕΡΕΥΝΑ ΚΑΙ ΕΦΑΡΜΟΓΕΣ ΤΟΥ OWL-S

4.1 Ασφάλεια Και Ακεραιότητα με το Πρότυπο OWL-S

4.1.1 Εισαγωγή

Η εισαγωγή των υπηρεσιών δικτύου έχει αλλάξει ριζικά τον τρόπο που διεξάγονται οι επιχειρηματικές συναλλαγές στη σημερινή αγορά. Οι περισσότερες χρηματοοικονομικές συναλλαγές βασίζονται πλέον στις εφαρμογές των υπηρεσιών δικτύου. Η κατάσταση στην οποία βρίσκεται τώρα η αρχιτεκτονική του τμήματος και των εφαρμογών των υπηρεσιών δικτύου, δεν προωθεί αρκετά τις δυνατότητες τους και ταυτόχρονα προωθεί ακόμη λιγότερο τις δυνατότητες των σημασιολογικών υπηρεσιών δικτύου. Οι ομάδες τυποποίησης όπως η OASIS και η W3C έχουν αρχικά εργαστεί πάνω σε θέματα σύνταξης της διαλειτουργικότητας και της ασφάλειας των υπηρεσιών δικτύου. Η έρευνα σχετικά με το πως τα σημασιολογικά πλούσια υπομνήματα εξυπηρετούν την εύρεση, τη σύνθεση και την επίκληση των υπηρεσιών δικτύου, ξεκινάει ουσιαστικά τώρα. Οι σημασιολογικές υπηρεσίες δικτύου υπόσχονται ότι μπορούν να δώσουν τις λύσεις για την πραγματοποίηση αυτού του μεγάλου οράματος.

Σε αυτό το κομμάτι θα εστιάσουμε στο θέμα της ασφάλειας των σημασιολογικών υπηρεσιών δικτύου. Θα δείξουμε πως μπορούμε να παρέχουμε πλούσια σημασιολογικά, ασφάλεια στις περιγραφές της υπηρεσίας OWL-S και πως μπορούμε να την ενισχύσουμε επεκτείνοντας την υπάρχουσα πολιτική που ακολουθείται για την ταυτοποίηση (OWL-S Matchmaker) και την OWL-S Virtual Machine, με μηχανισμούς ασφαλείας. Το OWL-S είναι ένα σύνολο από οντολογίες που χρησιμοποιούνται για τη περιγραφή των δυνατοτήτων, της διασύνδεσης και άλλων λεπτομερειών που αφορούν τις υπηρεσίες δικτύου και που έχουν σχεδιαστεί για να εξυπηρετούν την εύρεση, την επίκληση, και τη σύνθεση των αυτοματοποιημένων υπηρεσιών δικτύου. Το OWL-S αποτελείται από το προφίλ, το μοντέλο διαδικασίας και την υποστήριξη μιας υπηρεσίας δικτύου.

Σε διάφορες εργασίες που έχουν γίνει έχουν προταθεί οντολογίες για απαιτήσεις ασφαλείας υψηλού επιπέδου, που καθιστούν δυνατή την ταυτοποίηση

του αιτήματος ενός πελάτη με τις κατάλληλες υπηρεσίες. Ωστόσο η εύρεση των υπηρεσιών γίνεται όχι μόνο με βάση τις περιγραφές των λειτουργιών τους, αλλά και βάσει των κριτηρίων ασφαλείας. Παραδείγματος χάριν μια υπηρεσία δικτύου μπορεί να δηλώνει ικανή Open-PGP απόκρυψης και να απαιτεί από αυτόν που επικαλείται την υπηρεσία να πιστοποιείται και να επικοινωνεί σε XML. Προσθέσαμε επιπλέον λειτουργίες στην DAML-S Matchmaker θέτοντας την ικανή να επικυρώνει ότι οι απαιτήσεις ασφαλείας του πελάτη, ικανοποιούνται από τα χαρακτηριστικά της υπηρεσίας και ότι οι απαιτήσεις της υπηρεσίας ικανοποιούνται από τις δυνατότητες του πελάτη. Τα αποτελέσματα είναι χρήσιμα για γενικές αποφάσεις, όπως για το αν μια υπηρεσία παρέχει απόκρυψη ή τι είδους πιστοποίηση θα πρέπει κάποιος να έχει ώστε να κάνει πιστοποίηση του εαυτού του στην υπηρεσία.

Παρακάτω θα προτείνουμε μια πιο λεπτομερή σήμανση ασφαλείας στις παραμέτρους της υπηρεσίας σε ότι αφορά το προφίλ και το μοντέλο διαδικασίας. Επιπλέον επεκτείνουμε το πλαίσιο εργασίας προσθέτοντας υπομνήματα που αφορούν τις πολιτικές ασφαλείας, σε μια γλώσσα πλούσια και επεξηγηματική πάνω σε θέματα πολιτικής, τη Rei. Η Rei [40] είναι μια γλώσσα που επιτρέπει να ορίζονται κανόνες και περιορισμοί πάνω από τις κυριότητες ορισμένων οντολογιών. Αυτή η πολιτική πληροφόρησης χρησιμοποιείται στην επιλογή και επίκληση μιας υπηρεσίας. Τελευταία έχει γίνει μια σημαντική προσπάθεια τυποποίησης στην ασφάλεια που βασίζεται σε XML, όπως είναι η WS Security, η SAML της OASIS Security Services Technical Committee και οι προδιαγραφές ασφαλείας του Liberty Alliance Project. Η συγκεκριμένη εργασία δεν λαμβάνει υπόψη τη σημασιολογική πλευρά των υπηρεσιών δικτύου. Έρευνα που έχει γίνει πάνω σε θέματα πολιτικής ασφαλείας για το σημασιολογικό δίκτυο όπως είναι η [41] και η [42], σχετίζεται με την εργασία χωρίς να εστιάζει στις σημασιολογικές υπηρεσίες δικτύου. Έχει γίνει επίσης σημαντική έρευνα πάνω στη πολιτική ασφαλείας των διανεμημένων συστημάτων. Η KAos παρέχει μια αντιπροσωπευτική γλώσσα πολιτικής που βασίζεται στην DAML+OIL. Αν και είναι μια ενδιαφέρουσα προσέγγιση, η DAML+OIL δεν είναι ικανή από μόνη της να συλλάβει το συνολικό εύρος της πολιτικής των περιορισμών. Έχουν γίνει αρκετές προσπάθειες για την πρόσθεση συντακτικού για κανόνες στην DAML+OIL και στην OWL. Η Ponder είναι μια γλώσσα πολιτικής προδιαγραφών που αναπτύχθηκε στο Imperial College. Παρόλο που είναι ευέλικτη και εκφραστική αποδεικνύεται περισσότερο χρήσιμη σαν γλώσσα σύνταξης παρά στις υπηρεσίες δικτύου. Η Rei αναπαρίσταται σε RDFS και περιλαμβάνει υπομνήματα σε γλώσσα

προγραμματισμού prolog για την έκφραση κανόνων σε θέματα πολιτικής και προχωρά πέρα από τις δυνατότητες της DAML+OIL και OWL.

4.1.2 Εισαγωγή της Πολιτικής Ασφαλείας στις Περιγραφές των Υπηρεσιών Δικτύου

Οι πολιτικές (policies) θα πρέπει να είναι μέρος της αναπαράστασης των υπηρεσιών δικτύου και συγκεκριμένα των σημασιολογικών υπηρεσιών δικτύου. Οι πολιτικές μας παρέχουν τις διευκρινίσεις σχετικά με το ποιος μπορεί να χρησιμοποιήσει μια υπηρεσία, κάτω από ποιες συνθήκες, πως θα πρέπει να παρέχονται οι πληροφορίες στην υπηρεσία καθώς και το πώς οι παρεχόμενες πληροφορίες θα χρησιμοποιηθούν αργότερα. Θεωρούμε ότι το μοντέλο πελάτη-εξυπηρετητή είναι ένα μοντέλο στο οποίο ο πελάτης ζητά μια υπηρεσία δικτύου. Θεωρούμε επίσης ότι η χρήση πολιτικών είναι συμμετρική, με την έννοια ότι τόσο ο προμηθευτής όσο και ο αιτών περιορίζονται από ένα σύνολο πολιτικών που θα πρέπει να τηρηθούν κατά τη διάρκεια της αλληλεπίδρασής τους.

Θα αναφερθούμε σε δύο είδη πολιτικών: στις πολιτικές ασφαλείας και στις πολιτικές πιστοποίησης. Οι πολιτικές ασφαλείας ορίζουν κάτω από ποιες συνθήκες μπορεί να γίνει ανταλλαγή πληροφοριών και ποιες είναι οι θεμιτές χρήσεις αυτών των πληροφοριών. Θα πρέπει να τονίσουμε ότι ο προμηθευτής μπορεί να δώσει στον αιτών το δικαίωμα πρόσβασης σε απόρρητες πληροφορίες, αν και μόνο αν το κλειδί πρόσβασης παραμένει απόρρητο κατά τη διάρκεια της συναλλαγής. Ο αιτών θα πρέπει να αποφασίσει κατά πόσο μπορεί να ικανοποιήσει το συγκεκριμένο όρο ή όχι. Παρομοίως ο αιτών μπορεί να έχει μια δική του πολιτική ασφαλείας απαιτώντας ορισμένες πληροφορίες να παραμένουν απόρρητες και έτσι να μη μπορεί να μοιράζεται τις μη απόκρυφες πληροφορίες. Μια τέτοια πολιτική θα απέτρεπε στον αιτών να αλληλεπιδράσει με οποιαδήποτε υπηρεσία δικτύου που δεν εκτελεί την απαραίτητη απόκρυψη.

Οι πολιτικές ασφαλείας δεν είναι χρήσιμες μόνο όταν θέλουμε να προσδιορίσουμε ποιες πληροφορίες είναι απόρρητες κατά τη διάρκεια μιας συναλλαγής αλλά και μετά από τη λήψη δεδομένων. Για παράδειγμα όταν μια υπηρεσία δηλώνει ότι δεν μπορεί να διανείμει πληροφορίες που λαμβάνει ως εισαχθέντα δεδομένα, αυτό μπορεί να είναι σημαντικό κριτήριο για κάποιον που

απαιτεί ασφάλεια. Αυτή η πολιτική μεταφράζεται σαν υποχρέωση στην υπηρεσία δικτύου και τηρείται όπως ένα συμβόλαιο. Αν για παράδειγμα μετά την επίκληση της, δώσει πληροφορίες που αφορούν τον αιτών σε κάποιον που προωθεί προϊόντα, ο αιτών μπορεί να λάβει νομικά μέτρα ενάντια στην υπηρεσία. Όσο οι χρηματοοικονομικές συναλλαγές γίνονται σύνηθες φαινόμενο ανάμεσα στις υπηρεσίες δικτύου και όσο αυτές ασχολούνται με απόρρητες πληροφορίες όπως είναι τα προσωπικά δεδομένα, είναι λογικό να αναμένουμε ενίσχυση των πολιτικών ασφαλείας. Θα πρέπει επίσης να λάβουμε υπόψη τις πολιτικές πιστοποίησης, δηλαδή πολιτικές που περιορίζουν τον προμηθευτή να δέχεται αιτήσεις μόνο από συγκεκριμένους πελάτες. Για παράδειγμα μια πολιτική πιστοποίησης μπορεί να δηλώνει ότι ο αιτών πρέπει να δρα εκ μέρους ενός ατόμου που είναι μέλος μιας συγκεκριμένης ομάδας ενός οργανισμού και το γεγονός ότι είναι μέλος μπορεί να αποδειχθεί μέσω ψηφιακής πιστοποίησης. Παρομοίως ο αιτών μπορεί να περιορίσει τη ζήτηση υπηρεσιών σε ορισμένο αριθμό προμηθευτών.

4.1.3 Παράδειγμα

Θα χρησιμοποιήσουμε ένα παράδειγμα για να δούμε τις διάφορες πλευρές των πολιτικών ασφαλείας του OWL-S. Ας υποθέσουμε ότι ένας επιστήμονας ψάχνει μία on-line υπηρεσία υπολογισμού που θέλει να χρησιμοποιήσει για την επεξεργασία δεδομένων από πειράματα. Ο επιστήμονας επιθυμεί να παραμείνουν απόρρητες όλες οι προσωπικές πληροφορίες που θα γνωστοποιήσει στην υπηρεσία. Αυτό σημαίνει ότι ψάχνει μόνο για υπηρεσίες δικτύου που θα αποδέχονται αποκρυπτογραφημένα δεδομένα και δεν θα μοιραστούν τα προσωπικά του δεδομένα με άλλες υπηρεσίες ή πράκτορες. Αυτή είναι η πολιτική ασφαλείας του επιστήμονα.

Ας υποθέσουμε ότι υπάρχει μια υπηρεσία δικτύου την οποία παρέχει ένας οργανισμός και η οποία μπορεί να εκτελέσει τους απαραίτητους υπολογισμούς δεδομένων. Αυτή η υπηρεσία είναι προσβάσιμη μόνο από τα μέλη μιας συγκεκριμένης ομάδας επιλεγμένων οργανισμών και έτσι ο επιστήμονας μπορεί να εγγραφεί στην υπηρεσία μόνο με τρόπο που μπορεί να πιστοποιηθεί. Αυτή είναι η πολιτική έγκρισης της υπηρεσίας δικτύου.

Μπορούμε να προσεγγίσουμε τις πολιτικές ασφαλείας και έγκρισης σε δύο επίπεδα. Σε πραγματικό επίπεδο προτείνουμε οντολογίες που θα υπομνηματίζουν τις

παραμέτρους που εισάγονται και εξάγονται με χαρακτηριστικά ασφαλείας. Αυτά τα χαρακτηριστικά δηλώνουν αν οι παράμετροι είναι αποκρυπτογραφημένοι ή ψηφιακά υπογεγραμμένοι. Σε ένα πιο θεωρητικό επίπεδο δείχνουμε τον τρόπο τυποποίησης των πολιτικών ασφαλείας και προσωπικών δεδομένων με τη γλώσσα Reif. Η επιλογή των υπηρεσιών δικτύου που ικανοποιούν τον αιτών, είναι μέρος της εκτεταμένης διαδικασίας ταυτοποίησης της OWL-S.

Η ενίσχυση των μηχανισμών κρυπτογράφησης, όπως τα κρυπτογραφημένα ή υπογεγραμμένα μηνύματα θα επιτευχθεί μέσω της ενοποίησης με την OWL-S VM, μια γενική διαδικασία για το μοντέλο διαδικασίας της OWL-S και ένα εργαλείο για την αυτόματη επίκληση των υπηρεσιών OWL-S.

4.1.4 Ασφάλεια και Πιστοποίηση OWL-S

Θα προτείνουμε οντολογίες και σημάνσεις για να αποσπάσουμε τις πληροφορίες ασφαλείας από τις παραμέτρους που εισάγονται και εξάγονται στις υπηρεσίες δικτύου. Το πρόβλημα της αναπαράστασης απόρρητων δεδομένων στη σήμανση των σημασιολογικών υπηρεσιών όπως το OWL-S είναι ότι τα απόρρητα δεδομένα από τη φύση τους δεν φανερώνουν την εσωτερική τους δομή ή σημασία γιατί είναι μόλις ένα byte string. Για αυτό προτείνουμε μια σημασιολογική σήμανση που ορίζει τα χαρακτηριστικά ασφαλείας των εισαχθέντων και εξαχθέντων δεδομένων, των παραμέτρων, των υπηρεσιών δικτύου ενώ ταυτόχρονα διατηρεί τις πληροφορίες που αφορούν τη δομή των δεδομένων, χωρίς να αποκαλύπτει την σημασία τους. Οι μετα-πληροφορίες αυτές μπορούν να χρησιμοποιηθούν για την επιλογή υπηρεσίας.

Μια βασική οντολογία για τη διαχείριση των κρυπτογραφημένων λεπτομερειών των εισαχθέντων και εξαχθέντων δεδομένων των παραμέτρων των υπηρεσιών δικτύου μπορούμε να βρούμε στην διεύθυνση <http://www.csl.sri.com/users/denker/owl-sec/infObj.owl>. Προκειμένου να κρατήσουμε τα κρυπτογραφημένα ή προσημασμένα, εισαχθέντα ή εξαχθέντα δεδομένα, ορίζουμε μία InfObj τάξη (information object) και τις υποκατηγορίες της τάξης EncInfObj (encrypted information object), SigInfObj (signed information object). Η InfObj χρησιμοποιείται για να προσδιοριστεί το εύρος των εισαχθέντων και εξαχθέντων παραμέτρων της OWL-S. Τα αντικείμενα πληροφοριών (information object)

περιέχουν ένα `baseObject` που περιγράφει το τύπο ή τη δομή της πληροφορίας που είναι κωδικοποιημένη σε αυτό. Για παράδειγμα το `baseObject` μιας παραμέτρου I/O της τάξης `EnclInfObj` μπορεί να είναι μια τάξη όπως η `SSN`. Αυτή η κυριότητα παράγει γνώση σχετικά με το είδος των δεδομένων που ανταλλάσσονται και μπορεί να χρησιμοποιηθεί στην υποστήριξη. Για παράδειγμα για το αν μια παράμετρος της υπηρεσίας ταιριάζει με τις απαιτήσεις του πελάτη ή για το αν οι παράμετροι που εισάγονται / εξάγονται σε δύο υπηρεσίες δικτύου ταιριάζουν ώστε να μπορούν οι υπηρεσίες να συνδυαστούν.

Η βασική οντολογία είναι επαρκής για να περιγράψουμε τις λεπτομέρειες κρυπτογράφησης στο παράδειγμα. Το πρώτο βήμα είναι η εύρεση και η περιγραφή μιας υπηρεσίας που ικανοποιεί τις απαιτήσεις του επιστήμονα. Το δεύτερο βήμα είναι η επίκληση της συγκεκριμένης υπηρεσίας. Παρακάτω περιγράφουμε πως θα πρέπει να διαχειρίζονται τα απόρρητα δεδομένα σε αυτά τα δύο βήματα.

Το σύστημα ταυτοποίησης χρησιμοποιείται για την εύρεση μιας υπηρεσίας υπολογισμού δεδομένων, η οποία θα ικανοποιεί τις λειτουργικές απαιτήσεις του υπαλλήλου. Θα παραλείψουμε τις λεπτομέρειες που αφορούν αυτές τις απαιτήσεις και θα εστιάσουμε στις απαιτήσεις ασφάλειας του πελάτη και της υπηρεσίας. Οι παράμετροι που εισάγονται και εξάγονται θα περιγραφούν ως `information objects` τα οποία αναφέρονται στο τύπο της πληροφορίας και στο είδος της τεχνικής ασφάλειας που τους εφαρμόζεται. Την ίδια προσέγγιση θα εφαρμόσουμε για τις απαιτήσεις και τις ικανότητες των πελατών. Στο συγκεκριμένο παράδειγμα υποθέτουμε ότι ο επιστήμονας μπορεί να παρέχει ένα αποκρυπτογραφημένο στιγμιότυπο της τάξης `Person`. Χρησιμοποιούμε την οντολογία FOAF (βλέπε <http://xmlns.com/foaf/0.1>) για να ορίσουμε ένα συγκεκριμένο πεδίο ορισμού πληροφοριών όπως όνομα, οργανισμό κ.τ.λ. Υποθέτουμε το παρακάτω ορισμό στιγμιότυπου της τάξης `Person` για να περιγράψουμε τον επιστήμονα.

```
<foaf:Person rdf:ID="MarySmith">
<foaf:name xml:lang="en">Mary Smith</foaf:name>
<foaf:title>Dr.</foaf:title>
</foaf:Person>
```

Η επιστήμονας δεν θέλει να αποκαλύψει τις προσωπικές της πληροφορίες σε οποιονδήποτε. Ένας τρόπος να αντιληφθούμε αυτή τη πολιτική ατομικών στοιχείων

είναι να ψάξουμε για υπηρεσίες που διαχειρίζονται τις πληροφορίες με κρυπτογραφικό τρόπο. Οι περιγραφές της υπηρεσίας OWL-S περιέχουν περιγραφές που εισάγονται στην περιοχή `process:hasInput`, όπου `process` είναι η συντόμευση για την τελευταία έκδοση της OWL-S και μπορεί να βρεθεί στη διεύθυνση <http://www.daml.org/services/owl-s/1.0>. Συνεπώς το είδος της υπηρεσίας που ψάχνει η επιστήμονας θα πρέπει να έχει τις παρακάτω παραμέτρους εισαγωγής στο προφίλ της.

```
<process:hasInput rdf:ID="PersonInf">
<process:parameterType rdf:resource="EncPersonInfObj"/>
</process:hasInput>
where
<Class rdf:ID="EncPersonInfObj">
<SubClassOf rdf:resource="#EncInfObject"/>
<Restriction>
<onProperty rdf:resource="baseObject"/>
<allValuesFrom rdf:resource="&foaf:#Person"/>
</Restriction>
</Class>
```

Η υπηρεσία ταυτοποίησης χρησιμοποιεί αυτή τη πληροφορία για να επιλέξει τις υπηρεσίες με το κατάλληλο προφίλ ή τις κατάλληλες περιγραφές διαδικασίας.

Μέχρι στιγμής έχουμε ασχοληθεί με εκείνα τα κομμάτια της ασφάλειας που μπορούμε να διαχειριστούμε με τις τεχνικές κρυπτογράφησης.

Αν υποθέσουμε ότι η υπηρεσία υπολογισμού δεδομένων βασίζει την έγκριση χρήσης της υπηρεσίας σε δύο στοιχεία: 1) η επιστήμονας πρέπει να είναι μέλος μιας συγκεκριμένης ομάδας επιλεγμένων οργανισμών και 2) η επιστήμονας θα πρέπει να εγγραφεί στην υπηρεσία με τρόπο που μπορεί να πιστοποιηθεί. Η πρώτη πολιτική θα αντιμετωπιστεί σαν *Req* πολιτική. Ο δεύτερος όρος εκφράζεται από την υπηρεσία σαν απαίτηση προς την επιστήμονα να εγγραφεί χρησιμοποιώντας το όνομα της και άλλες προσωπικές πληροφορίες με έναν επαληθεύσιμο τρόπο ώστε να αποφευχθούν οι επικίνδυνες περιπτώσεις απομίμησης (χρήσης της υπηρεσίας από κάποιον που χρησιμοποιεί τα στοιχεία του επιστήμονα). Η ψηφιακή υπογραφή είναι μια τεχνική κρυπτογράφησης που μας βοηθάει να πετύχουμε επαληθεύσιμη

πιστοποίηση. Μια υπηρεσία δικτύου θα μπορούσε να εκφράσει τις απαιτήσεις της πάνω στο θέμα της πιστοποιημένης εγγραφής με τον ακόλουθο τρόπο.

```
<process:hasInput rdf:ID="RegInf">
<process:parameterType rdf:resource="SigRegInfObj"/>
</process:hasInput>
<Class rdf:ID="SigRegInfObj">
<SubClassOf rdf:resource="#SigInfObject"/>
<Restriction>
<onProperty rdf:resource="base"/>
<allValuesFrom rdf:resource="&foaf;#Person"/>
</Restriction>
</Class>
```

Από τα παραδείγματα γίνεται φανερό ότι οι απαιτήσεις του πελάτη μπορεί να μην ταυτίζονται με τις απαιτήσεις της υπηρεσίας. Ο πελάτης απαιτεί οι προσωπικές πληροφορίες να είναι απόρρητες όταν τις εισάγει σε μια υπηρεσία και η υπηρεσία απαιτεί από το πελάτη να δηλώνει αυτές τις πληροφορίες. Παρόλα αυτά ο μηχανισμός ταυτοποίησης μπορεί να συμπεράνει ότι και ο πελάτης και η υπηρεσία δικτύου συμφωνούν στο γεγονός ότι πρέπει οι πληροφορίες να υποβάλλονται.

4.1.5 Αναπαράσταση και Αιτιολόγηση των Πολιτικών

Θα προσπαθήσουμε να ενοποιήσουμε τις πολιτικές που αφορούν διάφορες πλευρές της ασφάλειας συμπεριλαμβάνοντας την εξουσιοδότηση και την ατομικότητα στις σημασιολογικές υπηρεσίες. Οι πολιτικές είναι αρχικά χρήσιμες κατά τη διάρκεια της φάσης της εύρεσης και για τη διαμόρφωση ανεπίσημων συμβολαίων.

4.1.5.1 Πολιτικές Αναπαράστασης Χρησιμοποιώντας τη Rei

Οι πολιτικές που ακολουθούνται αναπαρίστανται χρησιμοποιώντας τη Rei, μια γλώσσα βασισμένη σε RDFS που χρησιμοποιείται στο προσδιορισμό των πολιτικών.

Το μοντέλο Reī βασίζεται στις δεοντολογικές έννοιες των δικαιωμάτων, των απαγορεύσεων, των υποχρεώσεων και των απαλλαγών. Αυτές οι έννοιες έχουν τέσσερα χαρακτηριστικά: αυτού που εκτελεί δράση, της ενέργειας, της πρόβλεψης και των περιορισμών. Οι περιορισμοί ορίζουν τις συνθήκες κάτω από τις οποίες ο δράστης, η ενέργεια και οποιαδήποτε άλλη οντότητα περιεχομένου θα πρέπει να είναι πραγματικά στοιχεία τη στιγμή της επίκλησης. Η πρόβλεψη περιγράφει τους όρους που πρέπει να είναι πραγματικοί μετά την επίκληση και αποτελεί υποχρέωση από την πλευρά του δράστη.

Η τάξη Policy βρίσκεται στη ρίζα της οντολογίας Reī. Στη δική μας εφαρμογή η τάξη αποτελείται από το εύρος που λαμβάνει η κυριότητα policyEnforced, μια νέα κυριότητα της OWL-S που περιγράφεται παρακάτω. Ορίζουμε τρεις υποκατηγορίες της τάξης Policy ώστε να ορίσουμε τους διαφορετικούς τύπους της πολιτικής που μπορούμε να υποστηρίξουμε. Οι κατηγορίες αυτές είναι η PrivacyPolicy, η AuthorizationPolicy και η Confidentiality-Policy.

Η Reī μοντελοποιεί επίσης διάφορα speech acts τα οποία μετατρέπουν τα δικαιώματα και τις υποχρεώσεις του αποστολέα και του παραλήπτη. Τα speech act περιλαμβάνουν την ανάθεση, την ανάκληση, την αίτηση, την ακύρωση, την υπόσχεση και την εντολή. Η Reī έχει ένα μοναδικό χαρακτηριστικό που επιτρέπει στα speech act να είναι μέρος των περιορισμών και των προβλέψεων άλλων δεοντολογικών εννοιών.

Μπορούμε να ορίσουμε στη Reī μια πολιτική έγκρισης που μπορεί να περιγραφεί ως εξής: «Επέτρεψε σε όλους να έχουν πρόσβαση στην υπηρεσία υπολογισμού δεδομένων, οι οποίοι όμως ανήκουν στην ίδια ομάδα με το προμηθευτή της υπηρεσίας». Για να κάνουμε πιο συγκεκριμένη αυτή τη πολιτική αξιοποιούμε την κυριότητα της OWL-S contactInformation την οποία μπορούμε να εξειδικεύσουμε να έχει την έκταση foaf:Agent. Με αυτή την κυριότητα μπορούμε να περιγράψουμε τον προμηθευτή της υπηρεσίας. Επιπλέον υποθέτουμε ότι υπάρχουν πληροφορίες για τις ομάδες στις οποίες ανήκει η επιστήμονας καθώς και πληροφορίες σχετικά με τις ομάδες στις οποίες ανήκει ο προμηθευτής της υπηρεσίας. Ένα κομμάτι της πολιτικής έγκρισης ορίζεται παρακάτω σε Reī.

@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.

@prefix rei: <http://www.csee.umbc.edu/~lkagai1/rei#>.

@prefix foaf: <http://xmlns.com/foaf/0.1/#>.

@prefix process: <http://www.daml.org/services/owl-s/1.0#>.

@prefix dcs: <http://www.somenamespace.com/dcs#>.

@prefix: <#>.

:actorVar a rei:Variable.

:providerVar a rei:Variable.

:projectVar a rei:Variable.

:AuthPolicy1 a rei:Right;

rei:agent actorVar;

rei:constraint [a rei:AndCondition;

rei:first[a rei:AndCondition;

rei:first[a rei:SimpleCondition;

rei:subject dcs:profile;

rei:predicate process:contactInformation;

rei:object providerVar];

rei:second[a rei:SimpleCondition;

rei:subject providerVar;

rei:predicate foaf:currentProject;

rei:object projectVar]];

rei:second[a rei:SimpleCondition;

rei:subject actorVar;

rei:predicate foaf:currentProject;

rei:object projectVar]]

Από την άλλη πλευρά ο αιτών μπορεί να έχει σαν πολιτική να μη μοιράζεται ποτέ τις προσωπικές του πληροφορίες. Αυτή η πολιτική μπορεί να εκφραστεί σε Rei με τον ακόλουθο τρόπο.

:serviceVar a rei:Variable.

:outputVar a rei:Variable;

:PrivPolicy1 a rei:Prohibition;

rei:action serviceVar;

rei:constraint [a rei:AndCondition;

rei:first[a rei:SimpleCondition;

rei:subject serviceVar;

```

rei:predicate process:hasOutput
rei:object outputVar];
rei:second [a rei:SimpleCondition;
rei:subject outputVar;
rei:predicate rdfs:domain;
rei:object foaf:Person]]

```

Συγκεκριμένα αυτή η πολιτική ατομικών στοιχείων λαμβάνει ως δεδομένο ότι όλες οι προσωπικές πληροφορίες της επιστήμονα, ορίζονται από την οντολογία FOAF. Αυτή η πολιτική δηλώνει ότι οποιαδήποτε υπηρεσία που έχει σαν έξοδο οτιδήποτε περιγράφει ένα FOAF person θα πρέπει να απαγορεύεται. Αυτή η πολιτική λειτουργεί σαν πρότυπο για τις επιτρεπόμενες ή απαγορευμένες υπηρεσίες, βασισμένη στις παραμέτρους εξόδου. Επιπρόσθετα ο αιτών μπορεί να θέλει να διευκρινίσει ότι οποιαδήποτε προσωπική πληροφορία μοιράζεται, θα πρέπει να παραμένει απόρρητη.

```

:serviceVar a rei:Variable.
:someVar a rei:Variable.
: PrivPolicy2 a rei:Right;
rei:action serviceVar;
rei:constraint [a rei:AndCondition;
rei:first[a rei:AndCondition;
rei:subject serviceVar;
rei:predicate process:hasInput;
rei:object someVar];
rei:second[a rei:SimpleCondition;
rei:subject someVar;
rei:predicate process:parameterType;
rei:object inf: EncPersonInfObj]].

```

Τέλος το Rei παρέχει ένα μηχανισμό προτεραιότητας που βοηθάει στις περιπτώσεις που οι πολιτικές συγκρούονται. Για παράδειγμα ορίζουμε παραπάνω δύο πολιτικές που είναι αντιφατικές. Η πρώτη ορίζει ότι ο αιτών δεν επιθυμεί να μοιράζεται τις προσωπικές του πληροφορίες και η δεύτερη ότι οι προσωπικές

πληροφορίες μπορούν να διαμοιραστούν μόνο αν παραμείνουν απόρρητες. Ο αιτών μπορεί να δηλώσει ότι η δεύτερη πολιτική έχει προτεραιότητα σε σχέση με τη πρώτη, διασφαλίζοντας ότι οι υπηρεσίες που θα συναντήσουν τη δεύτερη πολιτική, θα ελεγχθούν.

4.1.5.2 Επεκτείνοντας το OWL-S

Οι υπηρεσίες δικτύου περιγράφονται στο OWL-S με τη βοήθεια τριών στοιχείων: του προφίλ που προσφέρει μια γενική περιγραφή της υπηρεσίας, του μοντέλου διαδικασίας που περιγράφει πως η υπηρεσία εκτελεί το ρόλο της και το πρωτόκολλο αλληλεπίδρασης υπηρεσιών δικτύου και τέλος το στοιχείο της υποστήριξης που ορίζει πως οι ατομικές διαδικασίες στο μοντέλο διαδικασίας αναγράφονται σε WSDL.

Οι πληροφορίες ασφάλειας είναι απαραίτητες και στα τρία στοιχεία. Στο προφίλ καθορίζονται οι απαιτήσεις για την επίκληση της υπηρεσίας, καθώς το μοντέλο διαδικασίας και η αιτιολόγηση χρειάζονται μια προδιαγραφή των απαιτήσεων ασφάλειας για τα μηνύματα που ανταλλάσσονται ανάμεσα στην υπηρεσία και τον αιτών.

Δεν υπάρχει μια σαφής τοποθέτηση για τις πολιτικές ασφαλείας στο OWL-S, αλλά η λογική επέκταση για την εξασφάλιση της ασφάλειας είναι να ενωθούν οι πολιτικές με το προφίλ. Το λογικό θα ήταν οι πολιτικές να περιγράφουν τις γενικές ιδιότητες της υπηρεσίας δικτύου, από το να είναι οι ιδιότητες δεδομένες σε κάθε διαδικασία.

Βασισμένη σε προηγούμενη εργασία [43] θα προτείναμε οι πολιτικές να ήταν επέκταση των απαιτήσεων ασφάλειας των υπηρεσιών και θα προτείναμε την πρόσθεση μιας κυριότητας η οποία θα καλείται `policyEnforce` και η οποία θα ορίζεται σαν υποδιαίστερη κυριότητα του `securityRequirement`. Το `Policyenforced` περιγράφει τις πολιτικές που πρέπει να ενισχυθούν προκειμένου να εκτελεστούν σωστά οι λειτουργίες της υπηρεσίας.

Η πολιτική έγκρισης για την απαίτηση της υπηρεσίας που περιγράψαμε παραπάνω θα μπορούσε να επισημανθεί ως εξής:

```
<profile:Profile rdf:ID="DataComputationService_Profile">
```

```
<profile:textDescription>
This data computation service requires authorization.
</profile:textDescription>
...
<policyEnforced: rdf:resource="#AuthPolicy1"/>
</profile:Profile>
```

4.1.5.3 Επιλογή Προμηθευτών Χρησιμοποιώντας τις Πολιτικές

Οι ιδιότητες παίζουν σημαντικό ρόλο στην εύρεση και στην επιλογή των προμηθευτών. Οι πολιτικές καθορίζουν τους περιορισμούς στην διαδικασία αλληλεπίδρασης με τον προμηθευτή καθώς και το τι πρέπει να γίνει με τις πληροφορίες που ανταλλάσσονται. Οποιαδήποτε παραβίαση των πολιτικών κατά τη διάρκεια της αλληλεπίδρασης θα προκαλούσε την αποτυχία της. Για αυτό θα πρέπει ο αιτών να επιλέξει προμηθευτή με συμβατικές πολιτικές. Αν ο αιτών δηλαδή έχει ως πολιτική να διατηρεί απόρρητες τις πληροφορίες που μεταδίδει, δεν θα μπορέσει να αλληλεπιδράσει με έναν προμηθευτή που έχει ως πολιτική να στέλνει τις πληροφορίες ανοιχτά.

Κατά τη διάρκεια της διαδικασίας εύρεσης ο αιτών έχει την ευθύνη της επιλογής του καταλληλότερου προμηθευτή και σαν μέρος της διαδικασίας θα πρέπει να επιβεβαιώσει την συμβατότητα των πολιτικών του με αυτές του προμηθευτή. Η Rei παρέχοντας ένα μηχανισμό υποστήριξης πάνω στις πολιτικές και στις κυριότητες μπορεί να εκτιμήσει τη συμβατότητα των δικαιωμάτων, των απαγορεύσεων και των υποχρεώσεων μέσα στην κυριότητα.

Η αρχική προσφορά όσων αναφέραμε είναι ουσιαστικά η ενοποίηση των αιτιολογήσεων που πραγματοποιεί η Rei πάνω σε θέματα πολιτικών, χρησιμοποιώντας ένα μηχανισμό ταυτοποίησης που καλείται MatchMaker [9]. Τα αποτελέσματα συνοψίζονται παρακάτω.

Για να ικανοποιήσει τους περιορισμούς ασφάλειας ατομικών στοιχείων, ο μηχανισμός ταυτοποίησης επιλέγει πρώτα εκείνους τους προμηθευτές που έχουν τις ικανότητες που αναμένει ο αιτών. (Βήμα 1). Έπειτα ο μηχανισμός ταυτοποίησης εξάγει τις πολιτικές ασφάλειας από τον αιτών και τον προμηθευτή. (Βήμα 2).

Χρησιμοποιεί τη πολιτική υποστήριξης Re_i για να εξακριβώσει τη συμβατότητα των πολιτικών. Αν οι πολιτικές προκύψουν ασύμβατες, τότε ο προμηθευτής εγκαταλείπεται αλλιώς επιλέγεται. (Βήμα 3,4).

1. Ο μηχανισμός ταυτοποίησης παίρνει την προδιαγραφή OWL-S της υπηρεσίας, που ταιριάζει με τις λειτουργικές απαιτήσεις του αιτών.
2. Ανακτά τη πολιτική ασφάλειας ατομικών στοιχείων, που έχει οριστεί από τον πελάτη/αιτών και εξάγει τις πολιτικές ατομικότητας από το προφίλ του προμηθευτή.
3. Ο μηχανισμός ταυτοποίησης στέλνει τη περιγραφή OWL-S μαζί με τις πολιτικές ασφάλειας ατομικών στοιχείων, στο τμήμα υποστήριξης της Re_i.
4. Καθώς η πολιτική ασφάλειας ατομικών στοιχείων, ορίζει ποια πρότυπα υπηρεσιών είναι απαγορευμένα, το τμήμα υποστήριξης της Re_i ελέγχει αν η υπηρεσία εξάγει πληροφορίες που πελάτης επιθυμεί να διατηρήσει απόρρητες. Ελέγχει επίσης αν έρχονται σε αντίθεση οι πολιτικές ασφάλειας ατομικών στοιχείων, του προμηθευτή και του αιτών.
5. Αν η πολιτική ασφάλειας ατομικών στοιχείων, δεν ικανοποιείται, ο μηχανισμός υποστήριξης της Re_i το επιστρέφει σαν σφάλμα και συνεχίζει την αναζήτηση υπηρεσιών μέχρι να βρει αυτή που θα είναι συμβατή.

Παρομοίως ο αλγόριθμος για τις πολιτικές έγκρισης είναι ο ακόλουθος:

1. Ο μηχανισμός ταυτοποίησης εξάγει την προϋπόθεση της υπηρεσίας που έχει τύπο AuthorizationPolicy.
2. Συγκεντρώνει όλες τις σχετικές πληροφορίες που αφορούν το χρήστη και τις στέλνει μαζί με τη πολιτική έγκρισης στο μηχανισμό υποστήριξης της Re_i.
3. Αν ο μηχανισμός το επιστρέψει σαν αληθές, τότε η πολιτική έγκρισης ικανοποιείται και ο πελάτης μπορεί να χρησιμοποιήσει την υπηρεσία. Αλλιώς ο μηχανισμός συνεχίζει να ελέγχει μέχρι να βρει αυτή που θα είναι συμβατή.

4.1.5.4 Εξακριβώνοντας την Παρουσία των Πολιτικών κατά τη Διάρκεια της Αλληλεπίδρασης

Οι πολιτικές μπορούν να δηλωθούν στο προφίλ, πρέπει όμως να ενισχυθούν στο μοντέλο διαδικασίας που είναι υπεύθυνο για την αλληλεπίδραση ανάμεσα στο προμηθευτή και τον αιτών. Στην προδιαγραφή OWL-S, το μοντέλο διαδικασίας εκφράζει το πρωτόκολλο αλληλεπίδρασης του προμηθευτή που θα πρέπει να

ακολουθήσει ο αιτών προκειμένου να αλληλεπιδράσουν επιτυχώς. Περαιτέρω το στοιχείο υποστήριξης μας δίνει την αντιστοίχιση του μοντέλου διαδικασίας με τη προδιαγραφή του μηνύματος και συγκεκριμένα με τη WSDL και το SOAP.

Οι ανερχόμενες προδιαγραφές ασφαλείας για τις υπηρεσίες, θεωρούν ότι η ασφάλεια των μηνυμάτων μπορεί να οριστεί στο επίπεδο WSDL και SOAP. Αν ο αιτών θέλει να επιβεβαιώσει κατά πόσο θα ενισχυθούν οι πολιτικές κατά τη διάρκεια της αλληλεπίδρασης, τότε χρειάζεται να ελέγξει τους περιορισμούς που έχουν τεθεί από τον προμηθευτή για την ανταλλαγή μηνυμάτων.

Αν ο αιτών θέλει να ελέγξει αν ο προμηθευτής εμμένει στις δημοσιοποιημένες πολιτικές, θα πρέπει να αναλύσει τις διαφορετικές προδιαγραφές της προώθησης μηνυμάτων. Αυτό είναι επίσης απαραίτητο επειδή ο προμηθευτής μπορεί να μην εκθέσει ολοκληρωτικά τις πολιτικές του.

Ο αλγόριθμος που περιγράφεται παρακάτω είναι μια πρώτη απόπειρα να καταστεί ικανός ο αιτών να ελέγξει τη προσκόλληση του προμηθευτή στις πολιτικές. Ο αιτών χρησιμοποιεί τη προδιαγραφή του πρωτοκόλλου αλληλεπίδρασης για να αντλήσει τους διαφορετικούς τύπους πληροφοριών που εμπεριέχονται κωδικοποιημένοι στα μηνύματα που λαμβάνει ή στέλνει ο προμηθευτής (Βήμα 1,2). Η ενέργεια αυτή επιτυγχάνεται μέσω της αξιοποίησης της υποστήριξης που περιλαμβάνει το μοντέλο διαδικασίας και η WSDL. Το μοντέλο διαδικασίας περιλαμβάνει το τύπο της οντολογίας που περιέχει η πληροφορία που πρόκειται να αποτελέσει αντικείμενο ανταλλαγής, ενώ η WSDL περιγράφει πως η πληροφορία κωδικοποιείται μέσα στο μήνυμα. Τέλος (Βήμα 3) ένα σύστημα υποστήριξης χρησιμοποιείται για να εξακριβώσει κατά πόσο η κωδικοποίηση που χρησιμοποιείται είναι με βάση τις πολιτικές του προμηθευτή και του αιτών.

Θα μπορούσαμε να υλοποιήσουμε αυτό τον αλγόριθμο και με το μηχανισμό υποστήριξης της Re1. Αν τα αποτελέσματα της υποστήριξης των πολιτικών δεν είναι συνεπή με τις πολιτικές του αιτών, τότε εκείνος θα ξέρει ότι πρόκειται να προβεί σε παραβίαση αν επιδιώξει αλληλεπίδραση με τον συγκεκριμένο προμηθευτή. Αλλιώς αν η αιτιολόγηση παρουσιάσει ασυνέπεια ανάμεσα στις πολιτικές ο αιτών θα μπορεί να αποφασίσει αν θέλει να επιλέξει τον προμηθευτή βασιζόμενος στις πρόσθετες απαιτήσεις του.

1. Ο αιτών συγκεντρώνει το μοντέλο διαδικασίας, την υποστήριξη, τις προδιαγραφές WSDL και SOAP από τον προμηθευτή και τέλος τις πολιτικές που αφορούν εκείνον και τον προμηθευτή.

2. Χρησιμοποιεί το μοντέλο διαδικασίας, την αιτιολόγηση, τις προδιαγραφές WSDL και SOAP του προμηθευτή για να εντοπίσει το τύπο της κωδικοποίησης που υιοθετείται σε κάθε τύπο πληροφορίας.
3. Ο μηχανισμός υποστήριξης χρησιμοποιείται για να εξακριβώσει ότι α) οι πολιτικές του αιτών ικανοποιούνται και ότι β) οι πολιτικές του προμηθευτή ενισχύονται
4. Αν το πρώτο τεστ αποτύχει ο αιτών δεν χρησιμοποιεί τον προμηθευτή. Αν αποτύχει και το δεύτερο τεστ τότε ο αιτών αποφασίζει μόνος του για το αν θα χρησιμοποιήσει τον προμηθευτή.

Γενικά, είναι απαραίτητα στοιχεία σε έναν προμηθευτή η σαφήνεια και η ειλικρίνεια σε ότι αφορά τις πολιτικές του. Βέβαια αν ο προμηθευτής δεν είναι ειλικρινής και ορίζει μια πολιτική την οποία δεν ενισχύει, θα χάσει όλους τους πελάτες που δεν θέλουν να εμείνουν σε αυτή τη πολιτική καθώς και την εμπιστοσύνη όσων συνειδητοποιήσουν ότι οι πολιτικές του δεν ενισχύονται. Παρομοίως αν ο προμηθευτής δεν ορίσει σαφώς κάποιες από τις πολιτικές του, μπορεί να αλληλεπιδράσει με πελάτες που δεν μπορούν χειριστούν αυτές τις πολιτικές και για αυτό να αποτύχει η αλληλεπίδραση.

4.1.5.5 Ενισχύοντας τη Πολιτική και τη Πιστοποίηση σε OWL-S VM

Αναφέραμε προηγουμένως, ότι ένας τρόπος για να εκπληρωθεί η ασφάλεια ατομικών στοιχείων, ή η πιστοποίηση είναι μέσω της κωδικοποίησης ή των παραμέτρων εισαγωγής και εξαγωγής. Θα προτείναμε να παραμείνουν διαφανείς οι διαδικασίες κρυπτογράφησης στον αιτών, επεκτείνοντας το μηχανισμό επίκλησης της υπηρεσίας στην περίπτωση μας το OWL-S VM, με λειτουργίες αποκρυπτογράφησης ή σήμανσης των δεδομένων που ανταλλάσσονται ανάμεσα στο πελάτη και τον προμηθευτή.

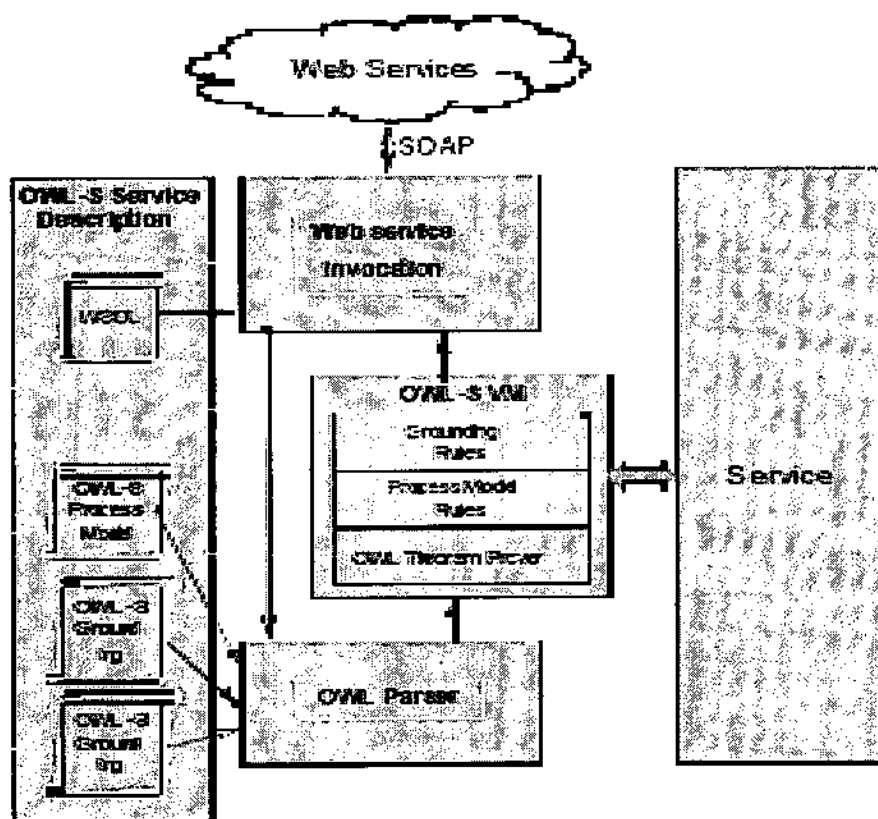
Το OWL-S Virtual Machine καθιστά ικανό το μοντέλο διαδικασίας και το μηχανισμό υποστήριξης του OWL-S, να χειρίζονται την κατάσταση αλληλεπίδρασης ανάμεσα στις υπηρεσίες δικτύου. Είναι ένας επεξεργαστής που βοηθάει στην αυτοματοποίηση της αλληλεπίδρασης μέσω της ανθρώπινης παρέμβασης. Η αρχιτεκτονική του OWL-S VM, που φαίνεται στο Σχήμα 11, αντιπροσωπεύεται από τα τρία στοιχεία στη κεντρική στήλη: την επίκληση για την υπηρεσία, το OWL-S VM

και την OWL-S Parser (ανάλυση). Το στοιχείο της επίκλησης ευθύνεται για την επαφή με άλλες υπηρεσίες και για την λήψη μηνυμάτων από άλλες υπηρεσίες. Η συναλλαγή με άλλες υπηρεσίες δικτύου μπορεί να βασίζεται στο SOAP ή απευθείας στο HTTP ή σε οποιαδήποτε άλλη μέθοδο επικοινωνίας. Εφόσον η υπηρεσία έχει λάβει το μήνυμα, το στοιχείο της επίκλησης εξάγει ότι του είναι χρήσιμο, με άλλα λόγια το περιεχόμενο των μηνυμάτων, και είτε το στέλνει στο OWL Parser είτε το περνάει απευθείας στο OWL-S VM.

Το OWL Parser είναι υπεύθυνο για την ανάγνωση τεμαχισμών στις οντολογίες και για τη μετατροπή τους σε κατηγορήματα που θα μπορούν να χρησιμοποιηθούν από το μηχανισμό συμπερασματολογίας της OWL-S. Με το OWL Parser μπορούμε επίσης να «κατεβάζουμε» μέσω του υπολογιστή, OWL οντολογίες που είναι διαθέσιμες στο δίκτυο καθώς και προδιαγραφές OWL-S άλλων υπηρεσιών για να αλληλεπιδρά με αυτές.

Το OWL-S VM χρησιμοποιεί τις οντολογίες που έχει συγκεντρώσει από το δίκτυο και τις προδιαγραφές OWL-S άλλων υπηρεσιών, για να ξεχωρίσει από τα μηνύματα που παρέλαβε τις πληροφορίες που θα στείλει. Προκειμένου να λάβει αυτές τις αποφάσεις χρησιμοποιεί ένα σύνολο από κανόνες που περιέχουν σημασιολογικά στοιχεία από την υποστήριξη και το μοντέλο διαδικασίας του OWL-S. Το OWL-S VM δημιουργεί και τα μηνύματα απόκρισης και για να το επιτύχει αυτό, χρησιμοποιεί το μηχανισμό υποστήριξης για τη μετατροπή των αφηρημένων πληροφοριών σε σταθερού περιεχομένου μηνύματα, τα οποία με τη σειρά τους περνάνε στη μονάδα επίκλησης για να μετατραπούν σε πραγματικά μηνύματα που στέλνονται στους παραλήπτες.

Οι άλλες δύο στήλες στο διάγραμμα είναι εξίσου σημαντικές. Η αριστερή στήλη δείχνει τις πληροφορίες που έχουν φορτωθεί από το δίκτυο και πως αυτές χρησιμοποιούνται από τις υπηρεσίες δικτύου OWL-S. Συγκεκριμένα η WSDL χρησιμοποιείται για την επίκληση υπηρεσιών δικτύου, ενώ οι οντολογίες και οι άλλες προδιαγραφές πρώτα αναλύονται και μετά χρησιμοποιούνται από το OWL-S VM ώστε να λάβει αποφάσεις σχετικά με το πώς θα συνεχίσει.



Σχήμα 11 - Η Αρχιτεκτονική της DAML-S Υπηρεσίας Δικτύου

Έχουμε επεκτείνει το OWL-S VM για να ενισχύσουμε τις πολιτικές ατομικότητας και πιστοποίησης. Έχουμε εφαρμόσει τις απαραίτητες μετατροπές ασφαλείας στις παραμέτρους I/O στο OWL-S VM. Με αυτό τον τρόπο, κατά τη διάρκεια της εκτέλεσης μιας ατομικής υπηρεσίας, το OWL-S VM χρησιμοποιεί τα υπομνήματα των σημασιολογικών παραμέτρων στο μοντέλο διαδικασίας για να ενισχύσει τους περιορισμούς ατομικότητας και πιστοποίησης που θα μπορούν να εφαρμοστούν μέσω τεχνικών κρυπτογράφησης (χρησιμοποιώντας την απόκρυψη και τη ψηφιακή υπογραφή).

Τα υπομνήματα ασφαλείας του SOAP χρησιμοποιούνται για την εφαρμογή της πραγματικής απόκρυψης ή υπογραφής των μηνυμάτων.

Με την εφαρμογή αυτών των μηχανισμών ασφαλείας, θα μπορούν οι υπηρεσίες δικτύου να διατηρούν μια ασφαλή επικοινωνία με τους συνεργάτες της.

4.1.5.6 Σύνοψη

Προσπαθήσαμε να μοντελοποιήσουμε τις διάφορες πτυχές της ασφάλειας των υπηρεσιών δικτύου. Προτείναμε τη πρόσθεση υπομνημάτων ασφάλειας ατομικών στοιχείων και πιστοποίησης για να εισάγουμε και να εξάγουμε παραμέτρους που θα βοηθήσουν στις υπηρεσίες του σημασιολογικού δικτύου. Το υπόμνημα περιλαμβάνει και το τύπο κρυπτογράφησης αν υπάρχει. Αναφέραμε το ρόλο της έγκρισης και της ασφάλειας των ατομικών στοιχείων για τις σημασιολογικές υπηρεσίες δικτύου και περιγράψαμε μια υποψήφια γλώσσα ορισμού πολιτικής.[44]

4.2 Η Χρήση των Σημασιολογικών Υπηρεσιών Δικτύου στο Περιβάλλον των Κινητών Εφαρμογών

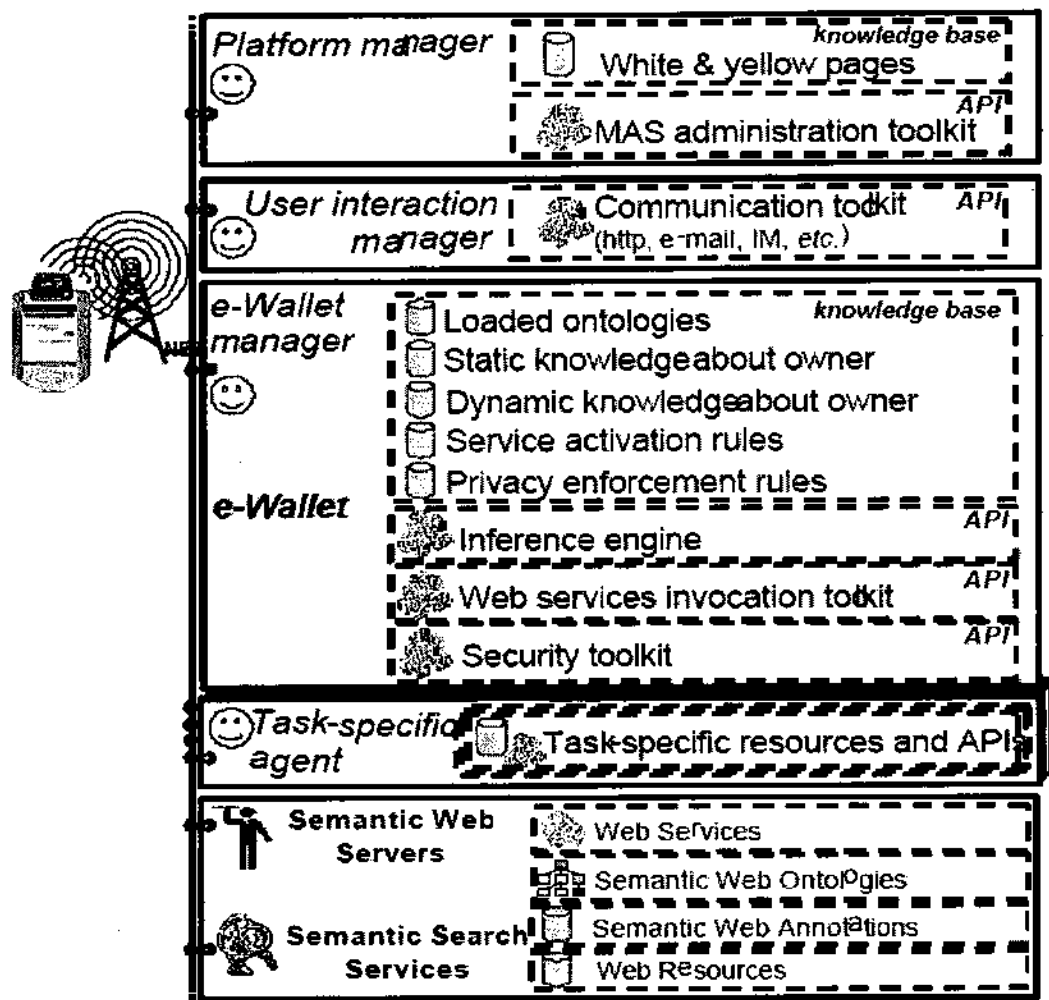
4.2.1 Εισαγωγή

Υπάρχουν πλέον εκατοντάδες φορητές συσκευές που επιτρέπουν τη πρόσβαση στο διαδίκτυο, ανοίγοντας το δρόμο σε πολλές νέες εφαρμογές και υπηρεσίες. Παρόλα αυτά στις υπάρχουσες εφαρμογές χρησιμοποιούνται ακόμα απλές υπηρεσίες. Για να προχωρήσουμε παραπέρα θα πρέπει να ξεπεράσουμε τους υπάρχοντες περιορισμούς αυξάνοντας το επίπεδο αυτοματισμού και αναπτύσσοντας υπηρεσίες που θα μπορούν να κατανοούν το περιβάλλον στο οποίο οι χρήστες λειτουργούν.

Παρακάτω θα δούμε με ποιους τρόπους μπορούμε να χρησιμοποιούμε τις πληροφορίες περιβάλλοντος μέσω των υπηρεσιών δικτύου για την εκτέλεση περίπλοκων εργασιών. Θα χρησιμοποιήσουμε σαν παράδειγμα το περιβάλλον του "Πανεπιστημίου μου" (*my Campus*), ένα πρωτότυπο περιβάλλον του σημασιολογικού δικτύου, στοχεύοντας στη προαγωγή της καθημερινής ζωής στο Πανεπιστήμιο Carnegie Mellon University (CMU).

4.2.2 Σύνοψη για το «Πανεπιστήμιο μου» (My Campus)

Στο «Πανεπιστήμιο μου» οι χρήστες μπορούν να χρησιμοποιήσουν διαφορετικούς πράκτορες για να τους βοηθήσουν σε διαφορετικές εργασίες. Για την σωστή λειτουργία, οι πράκτορες θα πρέπει να γνωρίζουν τα χαρακτηριστικά του περιβάλλοντος των χρηστών καθώς και τους πιθανούς χρήστες. Τα χαρακτηριστικά αυτά μπορεί να βρεθούν και αργότερα από διάφορες πηγές, που ουσιαστικά διαφέρουν από τον ένα χρήστη στον άλλο (π.χ δεν χρησιμοποιούν όλοι το Microsoft Outlook σαν ημερολόγιο) και με το χρόνο μπορεί να διαφέρουν και για τον ίδιο το χρήστη. Για να προσπεράσουμε το συγκεκριμένο πρόβλημα μοντελοποιούμε τις πληροφορίες περιβάλλοντος στο «Πανεπιστήμιο μου» σε σημασιολογικές υπηρεσίες δικτύου. Οι πράκτορες θα μπορούν αυτόματα να βρουν τις υπηρεσίες αυτές και να έχουν ταυτόχρονα και πρόσβαση. Η πρόσβαση στις πηγές περιβάλλοντος ενός χρήστη ελέγχεται με βάση τις προτιμήσεις που έχει ορίσει ο χρήστης στα θέματα ασφάλειας των ατομικών στοιχείων.



Σχήμα 12 - Η αρχιτεκτονική του "My Campus"

Το e-Wallet Manager εξυπηρετεί σαν χώρος αποθήκευσης της στατικής γνώσης που αφορά το χρήστη, όπως το Net.Passport, μόνο που εδώ η γνώση αναπαρίσταται με τη χρήση του OWL. Επιπλέον περιέχει γνώση που αφορά τους τρόπους πρόσβασης σε περισσότερες πληροφορίες σχετικά με το χρήστη επικαλώντας διάφορες πηγές που αναπαρίστανται ως υπηρεσίες δικτύου. Αυτή η γνώση αποθηκεύεται σε μια φόρμα κανόνων όπου αναφέρονται τα διαφορετικά χαρακτηριστικά περιβάλλοντος των υπηρεσιών. Με αυτό τον τρόπο το e-Wallet καθίσταται ικανό να αναγνωρίζει και να ενεργοποιεί εκείνες τις πηγές που σχετίζονται περισσότερο με τις απαντήσεις των ερωτημάτων σχετικά με το περιβάλλον του χρήστη. Οι κανόνες ασφάλειας των ατομικών στοιχείων που ορίζονται από το χρήστη αποθηκεύονται επίσης στο e-Wallet, εξασφαλίζοντας έτσι ότι οι πληροφορίες που αφορούν το χρήστη θα αποκαλύπτονται μόνο σε εξουσιοδοτημένα μέρη. Επιπλέον

προσαρμόζουν την ακρίβεια των πληροφοριών που παρέχονται, με βάση τους κανόνες προτίμησης του χρήστη.

Στο Σχήμα 12 παρέχεται μια γενική εικόνα του myCampus, εικονογραφώντας μια περίπτωση όπου η πρόσβαση γίνεται από PDA μέσω ενός ασύρματου δικτύου. Γενικά αυτή η αρχιτεκτονική αναφέρεται σε είδη περιβάλλοντος όπου οι χρήστες συνδέονται με την αντίστοιχη υποδομή μέσω ενός αριθμού καναλιών και συσκευών πρόσβασης. Οι πληροφορίες που τα αφορούν τη συγκεκριμένη συσκευή πρόσβασης και το κανάλι θεωρούνται κομμάτι του περιβάλλοντος του χρήστη και είναι διαθέσιμες μέσω του e-Wallet.

Είναι φανερό ότι οι πράκτορες δεν αντιμετωπίζουν περιορισμούς όταν θέλουν να έχουν πρόσβαση σε πληροφορίες που αφορούν τους χρήστες. Στο Πανεπιστήμιο CMU όπου έχουμε αναπτύξει το myCampus, έχουν πρόσβαση σε διάφορες υπηρεσίες όπως σε 23 υπηρεσίες δικτύου για εστιατόρια ή στην υπηρεσία που αναφέρει το δελτίο καιρού.

Στις παρακάτω ενότητες θα εστιάσουμε στη χρήση των υπηρεσιών δικτύου στο σύστημα μας.

4.2.3 Χρησιμοποιώντας τις Υπηρεσίες Δικτύου στις Πληροφορίες Περιβάλλοντος

Σε αυτή την ενότητα αναφέρουμε λεπτομερώς πως οι υπηρεσίες δικτύου χρησιμοποιούνται σαν πηγές για τις πληροφορίες περιβάλλοντος. Όπως έχουμε αναφέρει τα χαρακτηριστικά του περιβάλλοντος μπορούν να αποκτηθούν από διάφορες πηγές. Στο myCampus οι πηγές πληροφοριών σχετικά με το περιβάλλον είναι οι σημασιολογικές υπηρεσίες δικτύου. Αυτό σημαίνει ότι κάθε πηγή περιγράφεται από ένα προφίλ στο οποίο αναφέρονται οι λειτουργικές ιδιότητες σε σχέση με μία ή περισσότερες οντολογίες. Παραδείγματος χάριν το Microsoft Outlook είναι το στιγμιότυπο μιας πηγής πληροφοριών που παρέχει ταυτόχρονα πληροφορίες για τις ημερολογιακές δραστηριότητες και για την τοποθεσία. Οι περιγραφές υπηρεσιών περιλαμβάνουν επίσης λεπτομέρειες που αφορούν την επίκληση μιας υπηρεσίας. Χάρης σε αυτά τα προφίλ, μπορούμε αυτόματα να βρούμε και να έχουμε πρόσβαση σε πηγές που σχετίζονται με τις πληροφορίες περιβάλλοντος.

4.2.3.1 Κανόνες Επίκλησης Υπηρεσιών

Ένας αποτελεσματικός τρόπος για να αναγνωρίσουμε μια υπηρεσία που θα μπορεί να μας δώσει πληροφορίες σχετικά με κάποιο χαρακτηριστικό περιβάλλοντος π.χ τη τοποθεσία του χρήστη, είναι να χρησιμοποιήσουμε ένα σύνολο κανόνων. Οι κανόνες επίκλησης υπηρεσιών μας δείχνουν το σχεδιασμό των χαρακτηριστικών του περιβάλλοντος καθώς και τις πηγές πληροφοριών που είναι διαθέσιμες για τη πρόσβαση σε αυτά τα χαρακτηριστικά, αντιμετωπίζοντας κάθε προσωπική πηγή πληροφοριών σαν σημασιολογική υπηρεσία. Για παράδειγμα η τοποθεσία του χρήστη φαίνεται από την υπηρεσία Microsoft Outlook. Επιπλέον ισχυροποιούν τις οντολογίες των προσωπικών πηγών πληροφόρησης, καθιστώντας ικανό το e-Wallet να επιλέξει ανάμεσα από ένα αριθμό πιθανών πηγών πληροφόρησης, βασιζόμενο στη διαθεσιμότητα, στην ακρίβεια και σε άλλους σχετικούς παράγοντες. Για παράδειγμα στην απάντηση σε ένα ερώτημα σχετικά με τη τοποθεσία του χρήστη, οι κανόνες μπορούν να ορίσουν ότι όταν ο χρήστης οδηγεί, η καλύτερη διαθέσιμη μέθοδος είναι το GPS στο αυτοκίνητο. Αν είναι στη δουλειά και το ασύρματο PDA είναι ανοιχτό, τότε μπορεί να μάθει τη τοποθεσία του χρήστη χρησιμοποιώντας τη λειτουργία ανίχνευσης του ασύρματου δικτύου LAN της επιχείρησης. Αν όλα αποτύχουν, τότε ανατρέχοντας στο ημερολόγιο μπορεί να βρει πληροφορίες σχετικά με τη τοποθεσία.

4.2.3.2 Οι Σημασιολογικές Υπηρεσίες Χρησιμοποιώντας το OWL-S

Η πρόσθεση χαρακτηριστικών περιβάλλοντος στις υπηρεσίες μέσω κανόνων είναι ένα αποδοτικό σύστημα, ειδικότερα σε εφαρμογές που βασίζονται στο χρόνο. Αυτό όμως δεν είναι πάντα πρακτικό ειδικά αν λάβουμε υπόψη μας ότι οι χρήστες μπορεί να αποκτήσουν πράκτορες για την εκτέλεση νέων εργασιών ή νέες πηγές πληροφοριών περιβάλλοντος. Η λεπτομερής διατήρηση των κανόνων επίκλησης των υπηρεσιών που καλύπτουν κάθε πιθανή περίπτωση, είναι απλά μη πρακτικό. Αντιθέτως αποτελεί πιο ευέλικτη προσέγγιση το να βασιστούμε στις αυτόματες υπηρεσίες εύρεσης. Γενικά, οι πληροφορίες περιβάλλοντος για ένα δεδομένο χρήστη διασφαλίζονται στέλνοντας ένα ερώτημα στο e-Wallet του χρήστη. Όπως θα αναφερθεί λεπτομερώς και παρακάτω, σε αυτή τη περίπτωση το e-Wallet βασίζεται

στον συνδυασμό των κανόνων αναγνώρισης της τοπικής υπηρεσίας και σε μηχανισμούς εύρεσης τοπικών και διεθνών μηχανισμών προκειμένου να αναγνωρίσει μια ή περισσότερες πηγές για τις πληροφορίες περιβάλλοντος. Κατά τη διάρκεια της διαδικασίας εξασφαλίζεται ότι το αίτημα είναι συμβατό με τους κανόνες ασφάλειας του χρήστη.

Η εύρεση υπηρεσιών γίνεται ως εξής. Οι υπηρεσίες εγγράφονται σε ευρετήρια μαζί με τα προφίλ που περιγράφουν τις ικανότητες και τα χαρακτηριστικά τους. Δυστυχώς τα πρότυπα των υπηρεσιών που υπάρχουν για την εύρεση όπως το UDDI δεν επαρκούν. Το UDDI δεν περιγράφει μια υπηρεσία σε όρους λειτουργικής προσφοράς αλλά παρέχει μηχανισμούς για τη ταξινόμηση των υπηρεσιών σε όρους τυποποιημένων ταξινομιών όπως το North American Industry Classification System (NAICS)[45] και υποστηρίζει μόνο τη συντακτική ταυτοποίηση. Ένα ευρετήριο που διαφημίζει τα λειτουργικά χαρακτηριστικά μιας υπηρεσίας αποτελεί πολύ καλή εναλλακτική λύση. Αν αυτές οι περιγραφές αναπαρίστανται σε σημασιολογικές γλώσσες θα μπορούσε η αιτιολόγηση να χρησιμοποιηθεί στη σημασιολογική ταυτοποίηση των λειτουργιών της υπηρεσίας. Το πλαίσιο εργασίας της οντολογίας OWL για υπηρεσίες (OWL-S) μπορεί να χρησιμοποιηθεί για τη δημιουργία ευρετηρίων που περιέχουν τις περιγραφές των υπηρεσιών.

Οι πληροφορίες στο e-Wallet αποθηκεύονται σαν σημασιολογικά υπομνήματα, για αυτό μπορεί να χρησιμοποιήσει το OWL-S βασισμένο στη σημασιολογική εύρεση υπηρεσιών. Για περαιτέρω διασαφήνιση της ανάγκης για σημασιολογική εύρεση μπορούμε χρησιμοποιήσουμε το παράδειγμα μιας υπηρεσίας που διαθέτει λίστα με Ιταλικά εστιατόρια και χρησιμοποιεί σαν δεδομένο εισόδου τον ταχυδρομικό κώδικα. Ο χρήστης του e-Wallet θα μπορούσε να ψάχνει για εστιατόριο χρησιμοποιώντας την οντολογία φαγητό. Ας υποθέσουμε ότι το OWL-S χρησιμοποιείται για να αναπαραστήσει την υπηρεσία σαν ατομική υπηρεσία με δεδομένο εισαγωγής `loc:zipcode` (τοποθεσία: ταχυδρομικός κώδικας) και εξαχθέν δεδομένο `food:ItalianRestaurant` (φαγητό: Ιταλικό εστιατόριο). Ας υποθέσουμε επίσης ότι η οντολογία φαγητό κατηγοριοποιεί το Ιταλικό εστιατόριο σαν υποκατηγορία της τάξης του εστιατορίου. Χρησιμοποιώντας σημασιολογική συμπερασματολογία θα μπορούσε κάποιος να συμπεράνει ότι όλα τα στιγμιότυπα του Ιταλικού εστιατορίου είναι στιγμιότυπα του εστιατορίου και για αυτό η συγκεκριμένη υπηρεσία επιστράφηκε ως αποτέλεσμα. Αυτό το είδος ταυτοποίησης δεν είναι δυνατό να πραγματοποιηθεί σε UDDI γιατί κατά πρώτον δεν καθορίζονται τα δεδομένα εισαγωγής και εξαγωγής

και κατά δεύτερον το πλαίσιο εργασίας UDDI δεν παρέχει τη δυνατότητα σημασιολογικής υποστήριξης.

4.2.4 Η Σύνθεση των Υπηρεσιών

Η σύνθεση ορίζεται ως μια εργασία συγκέντρωσης ατομικών/βασικών υπηρεσιών για την εκτέλεση περίπλοκων εργασιών. Μπορούμε να αναλύσουμε τη σύνθεση μέσω ενός παραδείγματος. Όταν ένας χρήστης θέλει να ενημερωθεί σχετικά με το δελτίο καιρού, θα πρέπει αρχικά το σύστημα να εντοπίσει τη τοποθεσία του χρήστη προτού αναζητήσει μια υπηρεσία καιρού. Όταν εντοπιστεί η τοποθεσία λαμβάνεται και από το e-Wallet του χρήστη το οποίο ουσιαστικά καλεί την υπηρεσία δικτύου για να κρατήσει αυτή τη πληροφορία. Αυτή είναι μια πρωταρχική φόρμα σύνθεσης όπου η υπηρεσία καιρού απαιτεί την επίκληση της υπηρεσίας εντοπισμού. Αυτή σύνθεση πιθανότατα χρησιμοποιούσε "τύπους"(types) ταυτοποίησης των δεδομένων που εισάγονται και εξάγονται. Συνεχίζοντας τη παραπάνω υπόθεση θα μπορούσαμε να σκεφτούμε ότι η δεύτερη υπηρεσία θα μπορούσε να ζητηθεί μόνο αν η πρώτη απέδιδε μια συγκεκριμένη "τιμή" (value). Αυτές τις περιπτώσεις δεν μπορούμε να τις χειριστούμε με την ταυτοποίηση τύπων. Για να διατηρήσουμε αυτό το περιορισμό διάταξης χρησιμοποιούμε τις προϋποθέσεις που μεταφράζονται ως εξής: όταν μια προϋπόθεση σχετίζεται με μια υπηρεσία τότε μπορούμε να ζητήσουμε αυτή την υπηρεσία μόνο αν ικανοποιείται η προϋπόθεση. Μια προϋπόθεση ικανοποιείται βάσει της τιμής των δεδομένων που εξάγονται και με τα οποία ταυτόχρονα, σχετίζεται. Χρησιμοποιούμε μαζί την ταυτοποίηση τύπου και τιμής για την σύνθεση.

Οι υπηρεσίες περιγράφονται χρησιμοποιώντας δεδομένα που εισάγουμε, εξάγουμε και με τις προϋποθέσεις. Μοντελοποιούμε τις υπηρεσίες στο σύστημα μας με τη χρήση της οντολογίας για διαδικασίες της OWL-S. Πιο συγκεκριμένα όλες οι υπηρεσίες περιγράφονται χρησιμοποιώντας τις ατομικές διαδικασίες της OWL-S. Προτείνουμε επίσης τη πρόσθεση μιας νέας δομής realisedBy, η οποία συνδέει τις προϋποθέσεις με τα δεδομένα που εξάγονται. Μόνο κατά τη διάρκεια της επίκλησης μπορούμε να δούμε κατά πόσο τα δεδομένα που εξάγονται λαμβάνουν υπόψη τους τις προϋποθέσεις. Χρησιμοποιούμε αυτή τη δομή αρχικά στη σύνθεση και έπειτα στην επίκληση των υπηρεσιών. Αυτή η τεχνική αποσύνδεσης της σύνθεσης και της

επίκλησης μας βοηθάει να μάθουμε κατά πόσο το συγκεκριμένο σύνολο υπηρεσιών εξυπηρετεί το στόχο του χρήστη πριν καν ακόμα ζητήσουμε μια υπηρεσία. Αν έπρεπε να πραγματοποιήσουμε ταυτόχρονα τη διαδικασία εύρεσης και σύνθεσης τότε η εύρεση θα γινόταν απευθείας. Το πρόβλημα της απευθείας αλυσιδωτής εύρεσης προκύπτει εξαιτίας της διακλάδωσης που υπάρχει, δημιουργώντας έτσι ένα πολύ μεγάλο χώρο στον οποίο γίνεται η εύρεση. Λόγω αυτού υπάρχουν και άλλες επιπλοκές. Προκειμένου να επιτύχουμε το στόχο θα πρέπει να καλέσουμε διάφορες υπηρεσίες και αν ο στόχος δεν επιτευχθεί, τότε έχουμε χάσει απλά χρόνο καλώντας υπηρεσίες που δεν μας χρησιμεύουν. Για αυτό η προς τα πίσω αλυσιδωτή εύρεση είναι πιο αποτελεσματική.

4.2.4.1 Αυτόματοι Τελεστές Εξαγωγής Δεδομένων και Σχεδιασμού

Η συγκρότηση των ατομικών υπηρεσιών σε πιο σύνθετη μορφή μπορεί να αντιμετωπιστεί ως σχεδιασμός. Αυτό γίνεται μετατρέποντας τις ατομικές υπηρεσίες σε τελεστές σχεδιασμού και τρέχοντας έναν αλγόριθμο σχεδιασμού, μπορούμε να ενώσουμε αυτούς τους τελεστές. Το σχέδιο που δημιουργείται χρησιμοποιώντας αυτούς τους τελεστές συνιστά μια σύνθετη υπηρεσία. Όπως αναφέρθηκε προηγουμένως το περιβάλλον του myCampus έχει διάφορους εξειδικευμένους πράκτορες. Μερικοί από αυτούς εκτελούν απλές εργασίες όπως τη παροχή πληροφοριών σχετικά με τον καιρό ενώ άλλοι εκτελούν πιο περίπλοκες εργασίες όπως παραγγελίες για φαγητό, οι δεύτεροι χρησιμοποιούν το σχεδιασμό. Ο χρήστης ορίζει ένα στόχο χρησιμοποιώντας μια επιφάνεια σχεδιασμένη ειδικά για κάθε πράκτορα. Εφόσον έχει λάβει το στόχο, ο πράκτορας ελέγχει αν οι πληροφορίες ανήκουν στο e-Wallet του χρήστη ή αν μπορεί να τις παρέχει ο πράκτορας από τις υπηρεσίες που γνωρίζει. Αν και οι δύο έλεγχοι αποτύχουν τότε ξεκινάει η φάση εύρεσης. Αν βρεθεί η κατάλληλη υπηρεσία η διαδικασία μετατροπής των ατομικών υπηρεσιών σε τελεστές σχεδιασμού γίνεται ως εξής. Μια περιγραφή υπηρεσίας συνιστά μία ή περισσότερες ατομικές διαδικασίες. Οι περιγραφές της υπηρεσίας μετατρέπονται πρώτα σε κατηγορούμενο-υποκείμενο-αντικείμενο (PSO Predicate-Subject-Object). Για κάθε ατομική υπηρεσία φτιάχνουμε STRIPS-style τελεστές σχεδιασμού. Η πραγματική σύνθεση γίνεται από ένα απλό προς τα πίσω αλυσιδωτό αλγόριθμο σχεδιασμού.

Το πρόβλημα της σύνθεσης μπορεί να μετατραπεί σε πρόβλημα σχεδιασμού και δίνεται χρησιμοποιώντας το ζευγάρι τύπων $\langle Op, G \rangle$ όπου το Op είναι το σύνολο των τελεστών που αποκτούμε από τις περιγραφές της υπηρεσίας και το G είναι ο στόχος του χρήστη. Η σύνθεση πραγματοποιείται χρησιμοποιώντας το παρακάτω αλγόριθμο:

Compose (Goals G , Operators Op)

0. If G is empty
 - a. Return Success
1. Search Op for operators that satisfy all the goals G .
2. IF no operator is found
 - a. Return Fail
3. ELSE
 - a. Delete G
 - b. Add unavailable inputs of operator to G'
 - c. Add unavailable outputs (using realizedBy)
 - d. Compose(G' , Op)

Ο αλγόριθμος ψάχνει για εκείνες τις υπηρεσίες που θα ικανοποιήσουν το στόχο. Αν δεν βρεθούν, τότε αποτυγχάνει η σύνθεση και η εύρεση ξεκινάει από την αρχή για να βρει περισσότερους τελεστές. Αν από την άλλη όλοι οι στόχοι στο G ικανοποιηθούν, ψάχνει να βρει αν τα δεδομένα εισαγωγής σε αυτούς τους τελεστές είναι διαθέσιμα. Όσα δεν είναι μετατρέπονται σε στόχους στην επόμενη επανάληψη. Ένας παρόμοιος έλεγχος διαθεσιμότητας γίνεται στα δεδομένα που εξάγονται και σχετίζονται με τις προϋποθέσεις μέσω της δομής realizedBy. Όλα τα μη διαθέσιμα δεδομένα που εξάγονται, προστίθενται στους στόχους για την επόμενη επανάληψη. Η σύνθεση θα αποτύχει, αν το σύστημα δεν μπορέσει να βρει υπηρεσίες που θα ικανοποιήσουν ένα μέρος του στόχου.

Είναι πιθανό να βρεθούν πολλαπλοί χειριστές/υπηρεσίες που επιτυγχάνουν τον ίδιο στόχο/τμήμα του στόχου, τους οποίους υπολογίζουμε σαν ενδεχόμενα. Αν η πρώτη επιλογή αποτύχει να παράγει το επιθυμητό αποτέλεσμα τότε στρεφόμεστε στις εναλλακτικές υπηρεσίες. Η πρώτη επιλογή καθορίζεται με βάση τις προτιμήσεις του χρήστη.

Για περαιτέρω ανάλυση δείχνουμε τη χρήση της σύνθεσης χρησιμοποιώντας ένα σενάριο.

4.2.4.2 Σενάριο

Σύμφωνα με το σενάριο ο χρήστης επιθυμεί να βρει μια υπηρεσία παραδόσεων πίτσας και να κάνει μια παραγγελία. Για να εξηγήσουμε τη διαδικασία εύρεσης ας υποθέσουμε ότι το σημασιολογικό ευρετήριο χρησιμοποιεί μια οντολογία που βασίζεται στο NAICS. Προβαίνουμε σε ερώτηση στο κεντρικό ευρετήριο χρησιμοποιώντας την έκφραση `πίτσα_παράδοση_καταστήματα`. Το `Pizza_Delivery_Shops` (`πίτσα_παράδοση_καταστήματα`) είναι `subClassOf` `Limited_Service_Restaurants` στην οντολογία που βασίζεται στο NAICS. Το ευρετήριο επιστρέφει ένα μεγάλο αριθμό από υπηρεσίες που ταιριάζουν. Οι υπηρεσίες που είναι ταξινομημένες ως `Pizza_Delivery_Shops` έχουν υψηλότερη θέση σε σχέση με τις υπηρεσίες που είναι ταξινομημένες ως `Limited_Service_Restaurants`. Οι επιλογές μειώνονται περισσότερο αν λάβουμε υπόψη το περιβάλλον του χρήστη, στη συγκεκριμένη περίπτωση τη τοποθεσία του, που μπορεί να εξακριβωθεί από το `e-Wallet`. Ας υποθέσουμε ότι σαν αποτέλεσμα λαμβάνουμε μόνο ένα κατάστημα παράδοσης πίτσας. Το κύριο κομμάτι της περιγραφής της υπηρεσίας που επιστρέφεται από το ευρετήριο μας παραπέμπει στη Διαδικασία η οποία περιέχει τις ατομικές διαδικασίες. Συνθέτουμε τους τελεστές σχεδιασμού από αυτές τις περιγραφές και προσπαθούμε να κατασκευάσουμε το σχέδιο που θα καταστήσει το χρήστη ικανό να προβεί στη παραγγελία και να του παραδοθεί στη συγκεκριμένη διεύθυνση. Ο πίνακας 3 δείχνει τους τελεστές που χρησιμοποιούνται στην υπηρεσία παράδοσης πίτσας.

```
;Pizza Delivery Service
```

```
; s: service name
```

```
; i: input
```

```
; o: output
```

```
; p: precondition
```

```
(s LocationCheck
```

```
(i pin)
```

(o RangeCheck))
(s CreditCardInfo
(i CreditCardNumber)
(i CreditCardType)
(i CreditCardExpDate)
(o CreditCardStatus))
(s PizzaBuy
(p CCAuth)
(p DeliveryAddReq)
(i PizzaType)
(o PizzaBought))
(s DeliveryAdd
(p InRange)
(i Add1)
(i Add2)
(o AddReq))
(realizedBy InRange RangeCheck)
(realizedBy CCAuth CreditCardStatus)
(realizedBy DeliveryAddReq AddReq)

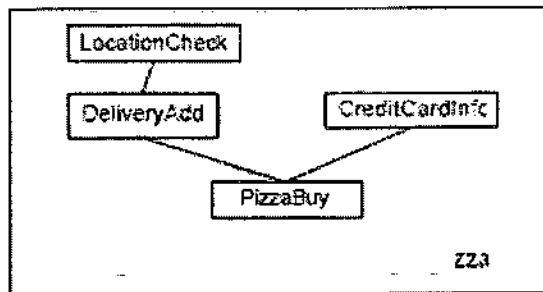
Πίνακας 3 - Τελεστές λειτουργίας για την παράδοση πίτσας

Η υπηρεσία ελέγχου τοποθεσίας (Location Check Service) επιβεβαιώνει ότι η παράδοση πραγματοποιήθηκε στη τοποθεσία του χρήστη. Η υπηρεσία πληροφοριών πιστωτικών καρτών (Credit Card Info Service) ελέγχει την πιστωτική κάρτα. Η υπηρεσία παράδοσης (Delivery Add Service) λαμβάνει την ακριβή διεύθυνση του χρήστη και τέλος η υπηρεσία αγοράς πίτσας (Pizza Buy Service) επιτρέπει στο χρήστη να αγοράσει τη πίτσα.

Η αίτηση του χρήστη μετατρέπεται σε στόχο που στη συγκεκριμένη περίπτωση είναι η αγορά πίτσας. Η υπηρεσία Pizza Buy, που επιτυγχάνει το στόχο και για αυτό περιλαμβάνεται στο σχέδιο, έχει δύο προϋποθέσεις και ένα δεδομένο εισαγωγής. Οι προϋποθέσεις σχετίζονται με τα δεδομένα που εξάγονται χρησιμοποιώντας το κατηγορούμενο realizedBy. Τα δεδομένα αυτά μαζί με το μοναδικό εισαχθέν δεδομένο αποτελούν τους στόχους της επόμενης επανάληψης.

Περιλαμβάνονται και άλλοι τελεστές στο σχέδιο με παρόμοιο τρόπο. Στο τέλος οι στόχοι ελέγχονται με βάση το e-Wallet του χρήστη. Οι χαμένες πληροφορίες λαμβάνονται προτρέποντας το χρήστη.

Όταν όλες οι πληροφορίες που χρειάζονται είναι διαθέσιμες, το σχέδιο μπορεί να υλοποιηθεί χάρις στο WSDL μέσω της υποστήριξης της OWL-S.



Σχήμα 13 - Σχέδιο Παραγγελίας

4.2.5 Σύμπέρασμα

Είδαμε συνοπτικά το περιβάλλον ενός σημασιολογικού δικτύου μέσω του myCampus στοχεύοντας στη προαγωγή της καθημερινής ζωής στο Πανεπιστήμιο. Μέσα στο myCampus οι χρήστες μπορούν με το χρόνο να αποκτήσουν διάφορους πράκτορες που θα τους βοηθήσουν στην εκτέλεση διαφορετικών εργασιών όπως στον προγραμματισμό συναντήσεων, στη διαμοίραση εγγράφων, στην οργάνωση εκδηλώσεων, στο φιλτράρισμα και τη δρομολόγηση των μηνυμάτων που εισέρχονται κ.α. Οι περισσότεροι από τους πράκτορες πρέπει να γνωρίζουν το περιβάλλον μέσα στο οποίο λειτουργεί ο χρήστης καθώς και τις πληροφορίες του περιβάλλοντος άλλων χρηστών. Στο myCampus οι πηγές πληροφόρησης του περιβάλλοντος, το ημερολόγιο, η λειτουργία εντοπισμού κ.α, αναπαρίστανται σαν σημασιολογικές υπηρεσίες δικτύου. Αναφέρονται ως προφίλ και σχετίζονται με διάφορες οντολογίες. Πληροφορίες σχετικά με την επίκληση μιας υπηρεσίας, περιλαμβάνουν και οι περιγραφές υπηρεσιών. Το τελικό αποτέλεσμα είναι ένα περιβάλλον όπου στις πηγές σχετικά με τις πληροφορίες για το χρήστη γίνεται αυτόματα η εύρεση και η πρόσβαση για την καλύτερη υποστήριξη των ερωτημάτων. Με αυτή τη προσέγγιση είναι δυνατή η εξυπηρέτηση χρηστών που χρησιμοποιούν διαφορετικά σύνολα

πηγών πληροφόρησης π.χ διαφορετικά συστήματα ημερολογίου και να προσαρμόζεται στις περιπτώσεις όπου οι χρήστες μπορεί να αλλάζουν πηγές πληροφόρησης.

Τα ερωτήματα σχετικά με το περιβάλλον ενός χρήστη υποβάλλονται στο e-Wallet του χρήστη το οποίο ενεργεί σαν φύλακας των προσωπικών πληροφοριών του ενισχύοντας τους κανόνες ιδιωτικού περιεχόμενου που έχει ορίσει.

Οι πράκτορες στο myCampus μπορεί να είναι από απλοί πράκτορες οι οποίοι βασίζονται σε μία ή περισσότερες πηγές πληροφοριών σχετικά με τους χρήστες τους ή πιο σύνθετοι πράκτορες που είναι ικανοί να δημιουργούν σχέδια ως απάντηση των αιτημάτων των χρηστών τους. Κάναμε μια ανάλυση ως προς τον τρόπο που οι σημασιολογικές υπηρεσίες δικτύου μπορούν να χρησιμοποιηθούν για την υποστήριξη 1) της δυναμικής εύρεσης και υποστήριξης των πηγών πληροφόρησης σχετικά με το χρήστη μέσω του e-Wallet 2) της αυτόματης παραγωγής σχεδίων από τους πράκτορες μέσω της εύρεσης υπηρεσιών που είναι σχεδιασμένες σαν τελεστές σχεδίασης και μπορούν δυναμικά να συντεθούν ώστε να ικανοποιήσουν έναν ή περισσότερους στόχους.[46]

4.3 Το Παράδοξο του Μεσίτη για το OWLS

4.3.1 Σύνοψη

Οι μεσίτες χρησιμοποιούνται ευρέως στα συστήματα διανεμημένων πληροφοριών, όπως στα συστήματα πολλαπλών πρακτόρων και στις διανεμημένες βάσεις δεδομένων. Ωστόσο δεν έχει γίνει ακόμα λεπτομερής ανάλυση της αρχιτεκτονικής των μεσιτών ενώ ταυτόχρονα δεν έχει προταθεί καμία γενική λύση σχετικά με το πώς θα πρέπει να επιτυγχάνονται οι εργασίες τους. Παρακάτω θα υπάρξει μια λεπτομερής ανάλυση σχετικά με αυτές τις εργασίες, καθώς και μια εφαρμογή βασισμένη στο OWLS. Θα δούμε ότι ενώ το OWLS μπορεί να παρέχει όλες τις πληροφορίες που χρειάζεται ένας μεσίτης, η απευθείας εφαρμογή ενός μεσίτη χρησιμοποιώντας το OWLS οδηγεί σε παράδοξο. Μπορούμε να λύσουμε το παράδοξο, επεκτείνοντας το μοντέλο διαδικασίας (Process Modeling) της γλώσσας

OWLS. Τέλος θα προταθούν λύσεις σε διάφορα ζητήματα που προκύπτουν στη διαχείριση μεσιτών από την αλληλεπίδραση με τις υπηρεσίες δικτύου.

4.3.2 Εισαγωγή

Οι μεσίτες διευκολύνουν την αλληλεπίδραση ανάμεσα σε δύο ή περισσότερα μέρη. Παραδείγματος χάριν αν δύο μέρη επιθυμούν να επικοινωνήσουν, αλλά δεν χρησιμοποιούν την ίδια γλώσσα επικοινωνίας, οι μεσίτες παρέχουν υπηρεσίες μετάφρασης ή ακόμα αν η μια πλευρά δεν «εμπιστεύεται» την άλλη, ο μεσίτης διαδραματίζει ρόλο ενδιάμεσου εμπιστοσύνης. Ακόμη οι μεσίτες μπορούν να προσφέρουν ανωνυμία στο ένα ή και στα δύο μέρη, αποτελώντας τον ενδιάμεσο της συναλλαγής.

Οι μεσίτες προέκυψαν από τους αυτόνομους πράκτορες, και αποτελούν μία σημαντική ανακάλυψη στο τομέα των συγχρονισμένων μηχανισμών. Χρησιμοποιούνται σε πολλές εφαρμογές με πράκτορες, όπως στην ενοποίηση πηγών που περιέχουν ετερογενείς πληροφορίες και βάσεων δεδομένων, στο ηλεκτρονικό εμπόριο πιο πρόσφατα στο συντονισμό υπηρεσιών δικτύου σε πλαίσιο εργασίας IRS-II. Θεωρητικές μελέτες δείχνουν ότι οι μεσίτες μπορούν να εκτελέσουν μια σειρά από συντονισμένες ενέργειες, όπως τη στάθμιση βάρους μεταξύ διαφορετικών πρακτόρων ή την ανωνυμία μεταξύ αυτών που αιτούνται και αυτών που παρέχουν.

Λόγω των ιδιοτήτων του και της ευρείας εφαρμογής του, ο μεσίτης θα μπορούσε να αποτελεί συστατικό στοιχείο της δομής των υπηρεσιών δικτύου. Ωστόσο η σημερινή αρχιτεκτονική υπηρεσιών δικτύου δεν περιλαμβάνει μεσίτες που χρησιμοποιούν πλήρως τις λειτουργίες της εύρεσης και της μεσολάβησης, σαν τμήμα της δομής των υπηρεσιών δικτύου.

Παρακάτω αναλύονται οι απαιτήσεις ενός μεσίτη που εκτελεί τις λειτουργίες της εύρεσης και της μεσολάβησης μεταξύ των πρακτόρων και των υπηρεσιών δικτύου. Γίνεται φανερό ότι ο μεσίτης εκτελεί περίπλοκες εργασίες αιτιολόγησης που περιλαμβάνουν α) την ερμηνεία των καταχωρήσεων που περιέχουν τις ικανότητες αυτών που παρέχουν υπηρεσίες β) την ερμηνεία των ερωτημάτων των αιτούντων που θα πρέπει να πραγματοποιηθούν από τον προμηθευτή υπηρεσιών γ) την εύρεση του καλύτερου προμηθευτή σύμφωνα με το ερώτημα του αιτούντα δ) την

επίκληση του προμηθευτή που έχει επιλεγεί εκ μέρους του αιτούντα αλληλεπιδρώντας με τον προμηθευτή προκειμένου να ολοκληρωθεί η διαδικασία ε) την επιστροφή των αποτελεσμάτων του ερωτήματος στον αιτούντα. Η επίτευξη αυτών των εργασιών απαιτεί οντολογίες που θα περιγράφουν τις ικανότητες των υπηρεσιών δικτύου, το μοτίβο αλληλεπίδρασης και την περιοχή στην οποία λειτουργούν, καθώς και τη λογική με βάση την οποία γίνεται η αιτιολόγηση σε αυτές τις οντολογίες. Επιπλέον θα γίνει περιγραφή της εφαρμογής ενός μεσίτη που χρησιμοποιεί OWL-S.

4.3.3 OWL-S

Η OWL-S είναι μια γλώσσα περιγραφής των υπηρεσιών του Σημασιολογικού Δικτύου, που εμπλουτίζει τις περιγραφές των υπηρεσιών δικτύου με σημασιολογική πληροφορία. Η OWL-S είναι χωρισμένη σε 3 τμήματα:

- 1) Στο Προφίλ (Profile), που περιγράφει τις ικανότητες των υπηρεσιών δικτύου, καθώς και τα πρόσθετα χαρακτηριστικά που βοηθούν στην περιγραφή της υπηρεσίας
- 2) Στο Μοντέλο Διαδικασίας (Process Model) που περιλαμβάνει τη περιγραφή των ενεργειών ενός προμηθευτή υπηρεσίας δικτύου, από την οποία ο αιτών μπορεί αντλήσει πληροφορία σχετικά με την υπηρεσία.
- 3) Στην Υποστήριξη (Grounding) που περιγράφει πως ένα μέρος της ανταλλαγής πληροφοριών αποτυπώνεται σε μηνύματα που ανταλλάσσει τελικά ο προμηθευτής και ο αιτών.

Η ικανότητα μιας υπηρεσίας δικτύου προσδιορίζεται από την περιγραφή των λειτουργιών της, δηλαδή τη περιγραφή του τι ακριβώς κάνει. Τις δυνατότητες των υπηρεσιών δικτύου μπορούμε να τις δούμε από δύο πλευρές: πρώτα σαν μια κατηγορία υπηρεσιών που βρίσκεται μέσα σε μια οντολογία υπηρεσιών π.χ η πώληση βιβλίων ανήκει στην οντολογία πώλησης προϊόντων ή σαν ένα μετασχηματισμό του συνόλου των δεδομένων που εισάγονται σε δεδομένα που εξάγονται π.χ η υπηρεσία μετατρέπει τα δεδομένα «τίτλος βιβλίου» και «συγγραφέας βιβλίου» στο «τιμολόγιο βιβλίου».

Το Προφίλ της OWL-S περιγράφει τις ικανότητες των υπηρεσιών δικτύου μέσω του μετασχηματισμού που παράγουν. Προκειμένου να γίνουν γνωστές οι δυνατότητες σε αυτούς που αιτούνται υπηρεσίες, ένας προμηθευτής υπηρεσιών διαφημίζει τις δυνατότητες με καταχωρήσεις που αφορούν την υποδομή, ή αλλιώς για την ακρίβεια με ενδιάμεσους πράκτορες, που καταγράφουν ποιοι πράκτορες είναι παρόντες στο σύστημα. Η UDDI αποτελεί παράδειγμα ενός ενδιάμεσου πράκτορα (middle agent) που μπορεί να κάνει περιορισμένη χρήση των πληροφοριών, που παρέχονται από το προφίλ της OWL-S. Το OWL-S/UDDI Matchmaker είναι ένα ακόμα παράδειγμα που συνδυάζει την UDDI και το OWL-S. Τέλος σαν ένα ακόμα παράδειγμα ενδιάμεσου πράκτορα, μπορούμε να αναφέρουμε το μεσίτη που εκτελεί τις λειτουργίες της εύρεσης και της μεσολάβησης.

Το δεύτερο κομμάτι του OWL-S είναι το μοντέλο διαδικασίας, που έχει δύο στόχους: ο πρώτος είναι να δείξει πώς ο προμηθευτής επιτυγχάνει τους στόχους του και ο δεύτερος να παρέχει το πρωτόκολλο αλληλεπίδρασης ανάμεσα στο αιτών και το προμηθευτή. Ο πρώτος στόχος επιτυγχάνεται επιτρέποντας στον προμηθευτή να δημοσιοποιήσει περιγραφή του τρόπου με τον οποίο λειτουργεί μέχρι το σημείο όπου ο προμηθευτής αισθάνεται «άνετα». Το OWL-S διακρίνεται σε δύο τύπους διαδικασιών: σε σύνθετες και ατομικές διαδικασίες. Οι ατομικές διαδικασίες αφορούν λειτουργίες που μπορεί ο προμηθευτής να εκτελέσει απευθείας. Οι σύνθετες διαδικασίες χρησιμοποιούνται για να περιγράψουν ένα σύνολο διαδικασιών είτε ατομικών είτε σύνθετων, οργανωμένες πάνω στη βάση ενός σχεδιαγράμματος ροής ελέγχου. Για παράδειγμα μία σειρά διαδικασιών που οι λειτουργίες της εκτελούνται η μία μετά την άλλη, ορίζεται σαν μια σύνθετη διαδικασία. Άλλες υποδομές ελέγχου που υποστηρίζονται από το OWL-S είναι η *cond* για υποθετικές παρατηρήσεις, η επιλογή για μη ντετερμινιστικές επιλογές ανάμεσα σε διαφορετικά σχεδιαγράμματα ελέγχου και η δημιουργία που αφορά μια νέα παράλληλη σειρά υπολογισμών. Τέλος η OWL-S περιλαμβάνει στοιχεία επανάληψης όπως το *ενώ (while)* και το *επανάλαβε μέχρι (repeat until)*.

Η εκτέλεση μιας διαδικασίας δημιουργεί μια κατάσταση μετάβασης όπου είτε οι πληροφορίες ανταλλάσσονται με κάποιο συνεργάτη είτε ένας πράκτορας προκαλεί αλλαγή στο περιβάλλον. Αυτή η κατάσταση χαρακτηρίζεται από τα στοιχεία φ, Π, όπου το Π αντιπροσωπεύει το σύνολο των παράλληλων υπολογισμών και το φ την

κατάσταση των υπολογισμών μέσα στην οποία εκτελείται η διαδικασία². Οι λειτουργίες μεταβάλλουν την κατάσταση αλλάζοντας είτε την κατάσταση των επαναλαμβανόμενων υπολογισμών, όταν για παράδειγμα μια ατομική διαδικασία αναγνώσει ένα μήνυμα από μια πύλη είτε αλλάζοντας το σύνολο των παράλληλων υπολογισμών μέσα από τη δημιουργία νέων ή το κλείσιμο παλιών. Η επίσημη σημασιολογία των ατομικών και σύνθετων διαδικασιών του OWL-S φαίνονται στον Πίνακα 4³. Τα σχέδια επαναλαμβανόμενων υπολογισμών υλοποιούνται σαν συνδυασμοί όρων και συνεχειών. Κάθε κανόνας στον Πίνακα 4 ορίζει πως η εκτέλεση μιας διαδικασίας μπορεί να αλλάξει ολόκληρη τη κατάσταση. Οι σειρές των διαδικασιών εκφράζονται με προσωρινούς περιορισμούς $\text{return } v \gg e$, όπου το e αναφέρεται στα αποτελέσματα v του προηγούμενου βήματος. Οι άλλοι κανόνες ορίζουν το αποτέλεσμα της εκτέλεσης άλλων τύπων σχεδίου ελέγχου. Το $\text{Cond}^{\text{True}}$ ορίζει τα αποτελέσματα της εκτέλεσης μιας δήλωσης που περιέχει συνθήκη αν η συνθήκη είναι αληθής, παρόμοιος κανόνας θα χρησιμοποιούνταν για τη περίπτωση που η συνθήκη θα ήταν ψευδής. Το $\text{Choice}^{\text{Left}}$ ορίζει τα αποτελέσματα της εκτέλεσης μιας μη ντετερμινιστικής επιλογής που βρίσκεται στη πρώτη λειτουργία της λίστας, ένας παρόμοιος κανόνας θα χρησιμοποιούνταν και για άλλες επιλογές. Τέλος το Atomic περιγράφει τα αποτελέσματα της εκτέλεσης μιας ατομικής διαδικασίας που έχει επίδραση στη κατάσταση των επαναλαμβανόμενων υπολογισμών αλλά δεν μεταβάλλει το σύνολο των παράλληλων διαδικασιών Π.

² Τα εκτελέσιμα σημασιολογικά στοιχεία που χρησιμοποιούνται είχαν αρχικά προταθεί για την DAML-S 0.6. Ενώ διάφορες πτυχές της γλώσσας έχουν αλλάξει κατά τη διαδικασία εξέλιξης της σε OWL-S 1.0., τα εκτελέσιμα σημασιολογικά στοιχεία των βασικών δομών του Μοντέλου Διαδικασίας παραμένουν έγκυρα.

³ Γίνεται μια σύντομη επεξήγηση των εκτελέσιμων σημασιολογικών στοιχείων του OWL-S. Η πλήρης παρουσίαση βρίσκεται στην [1].

<i>Seq</i>	$\neg \Pi, E[\text{return } v \gg e], \varphi \rightarrow \Pi, (E[(e \ v)], \varphi)$
<i>Spawn</i>	$\neg \Pi, (E[\text{spawn } e], \varphi) \rightarrow \Pi, (E[\text{return}()], \varphi), (e, \emptyset)$
<i>Cond^{True}</i>	$\neg \Pi, (E[\text{cond } C \ e1 \ e2], \varphi) \rightarrow \Pi, (E[e1], \varphi)$
	$\neg \Pi, (E[e1], \varphi) \rightarrow \Pi', (E[e1'], \varphi')$
<i>Choice^{Left}</i>	$\neg \Pi, (E[\text{choice } e1 \ e2], \varphi) \rightarrow \Pi', (E[e1'], \varphi')$
<i>Atomic</i>	$\neg \Pi, (E[\text{atomic } e], \varphi) \rightarrow \Pi, (E[\text{return}()], \varphi')$

Πίνακας 4

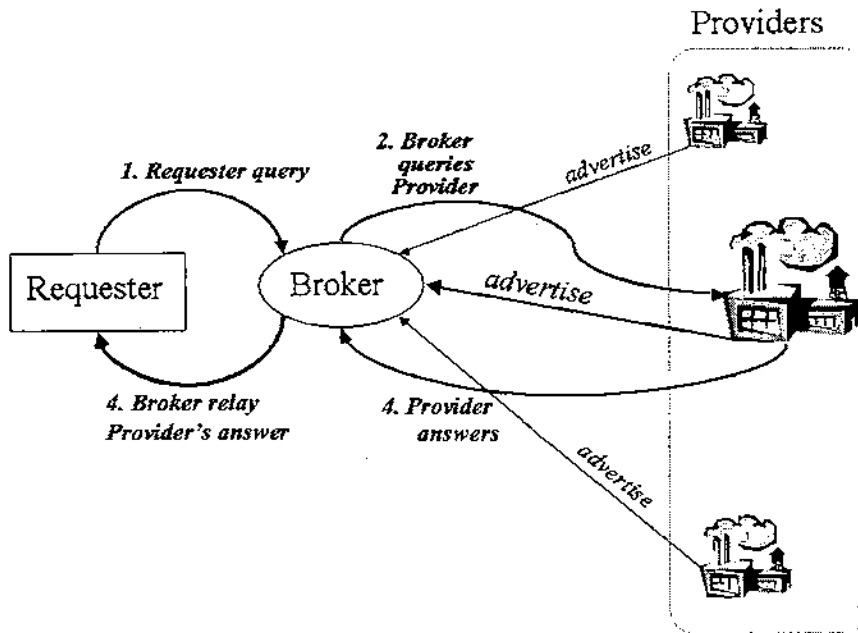
Το τελευταίο τμήμα της OWL-S είναι η Υποστήριξη που περιγράφει πως οι ατομικές διαδικασίες, που παρέχουν αποσπώμενες περιγραφές από τις ανταλλαγές με αυτούς που αιτούνται κάποια υπηρεσία, μετατρέπονται σε πραγματικά μηνύματα ή απομακρυσμένες κλήσεις στο δίκτυο. Πιο συγκεκριμένα η υποστήριξη της OWL-S περιλαμβάνει την αντιστοίχιση από τις ατομικές διαδικασίες στις προδιαγραφές των εισερχόμενων και εξερχόμενων μηνυμάτων της WSDL.

Η φιλοσοφία της αλληλεπίδρασης των υπηρεσιών δικτύου ανάμεσα σε κάποιον που αιτείται και σε κάποιον που παρέχει υπηρεσία είναι ότι αυτός που αιτείται θα πρέπει να γνωρίζει τη πληροφορία που ο προμηθευτής της υπηρεσίας απαιτεί στα διαφορετικά στάδια της αλληλεπίδρασης. Για παράδειγμα στα βιομηχανικά πρότυπα η αλληλεπίδραση αιτούντα και προμηθευτή πραγματοποιείται γνωρίζοντας το Web Services Description (WSD) του προμηθευτή που αποδίδεται σε WSDL. Στις υπηρεσίες του Σημασιολογικού Ιστού η αλληλεπίδραση αιτούντα και προμηθευτή προϋποθέτει από τη πλευρά του αιτούντα γνώση του Μοντέλου Διαδικασίας (συν το WSD) του προμηθευτή.

4.3.4 Γενική Περιγραφή ενός Μεσίτη

Κάθε συναλλαγή στην οποία εμπλέκεται ένας μεσίτης περιλαμβάνει 3 κομμάτια. Το πρώτο κομμάτι αφορά τον αιτών, αυτός ξεκινάει τη συναλλαγή ζητώντας πληροφορίες ή κάποια υπηρεσία στον μεσίτη. Το δεύτερο κομμάτι είναι ο

προμηθευτής, ο οποίος επιλέγεται ανάμεσα σε έναν αριθμό διαθέσιμων προμηθευτών ως ο πιο κατάλληλος για την επίλυση του προβλήματος του αιτούντα. Το τελευταίο κομμάτι είναι ο ίδιος ο μεσίτης.



Σχήμα 14 - Το Πρωτόκολλο του Μεσίτη

Το πρωτόκολλο στο Σχήμα 14 μπορεί να χωριστεί σε δύο μέρη: στο πρωτόκολλο δημοσιοποίησης και στο πρωτόκολλο μεσολάβησης. Στο πρωτόκολλο δημοσιοποίησης ο μεσίτης συλλέγει πρώτα τις δημοσιοποιήσεις των υπηρεσιών δικτύου που είναι διαθέσιμες να παρέχουν τις υπηρεσίες τους. Οι δημοσιοποιήσεις αυτές, οι οποίες απεικονίζονται στο Σχήμα 14 με τις λεπτές ευθείες γραμμές, χρησιμοποιούνται από το μεσίτη για να επιλέξει το καλύτερο προμηθευτή κατά τη διάρκεια της συναλλαγής με τον αιτούντα. Το πρωτόκολλο μεσολάβησης, που απεικονίζεται στο Σχήμα 14 με μαύρες καμπύλες γραμμές, απαιτεί α) από τον αιτούντα να ρωτήσει τον μεσίτη και να περιμένει απάντηση καθώς ο μεσίτης χρησιμοποιεί τις ικανότητες εύρεσης που έχει για να εντοπίσει κάποιον προμηθευτή που μπορεί να απαντήσει στο ερώτημα του αιτούντα. Εφόσον ο προμηθευτής έχει βρεθεί β) ο μεσίτης μετασχηματίζει την ερώτηση για τον προμηθευτή και προβαίνει τελικά στην ερώτηση. Εφόσον έχει λάβει το ερώτημα γ) ο προμηθευτής στέλνει την απάντηση στο μεσίτη και δ) στέλνει εκείνος με τη σειρά του την απάντηση στον αιτούντα.

Το πρωτόκολλο που περιγράφεται παραπάνω φανερώνει ότι ο μεσίτης πρέπει να εκτελέσει ένα σημαντικό αριθμό από περίπλοκες αιτιολογικές εργασίες τόσο για το

κομμάτι της εύρεσης όσο και για το κομμάτι της μεσολάβησης. Η διαδικασία της εύρεσης απαιτεί δύο διαφορετικές αιτιολογικές εργασίες. Η πρώτη είναι να αποσπάσει από το ερώτημα του αιτούντα τις ιδιότητες που είναι απαραίτητες να έχει ένας προμηθευτής ώστε να απαντήσει. Η δεύτερη διαδικασία είναι η σύγκριση και η ταυτοποίηση των ιδιοτήτων που είναι απαραίτητες για να απαντηθεί το ερώτημα, με τις ιδιότητες του προμηθευτή, ώστε να βρεθεί ο καλύτερος για το συγκεκριμένο ερώτημα.

Ο μεσίτης κατά τη διαδικασία μεσολάβησης θα πρέπει να μετατρέψει το ερώτημα του αιτούντα ώστε να μπορέσει να το στείλει στον προμηθευτή. Η διαδικασία μεσολάβησης έχει δύο συνιστώσες. Η πρώτη είναι η αποτελεσματική χρήση των πληροφοριών που παρέχονται από τον αιτούντα στον μεσίτη και η δεύτερη είναι η μετατροπή των μηνυμάτων του αιτούντα σε μηνύματα κατάλληλα διαμορφωμένα για τον προμηθευτή και αντίθετα.

Εφόσον ο αιτών δεν γνωρίζει ποιος είναι ο κατάλληλος προμηθευτής, το αρχικό ερώτημα που στέλνει στο μεσίτη και το ερώτημα που χρειάζεται ο προμηθευτής ώστε να μπορέσει να προσφέρει την αντίστοιχη υπηρεσία, μπορεί να μην ταιριάζουν.

Παραδείγματος χάριν ο αιτών ζητά να κλείσει την οικονομικότερη πτήση για Αθήνα – Πράγα. Εκτός από τον τόπο αναχώρησης και άφιξης, ο επιλεγμένος προμηθευτής μπορεί να χρειάζεται ημερομηνία και ώρα αναχώρησης. Στο συγκεκριμένο παράδειγμα ο αιτών δεν διευκρινίζει την ώρα αναχώρησης και ο προμηθευτής δεν μπορεί να βρει την οικονομικότερη επιλογή. Είναι αρμοδιότητα του μεσίτη να διευθετήσει την διαφορά που προκύπτει ανάμεσα στη πληροφορία που ζητείται και σε αυτή που παρέχεται αναγνωρίζοντας ότι εφόσον η πληροφορία της ώρας αναχώρησης δεν παρέχεται θα πρέπει να ζητηθεί καθώς και να επιλέξει την οικονομικότερη πτήση ανάμεσα σε αυτές που έχει βρει ο προμηθευτής.

Επιπλέον ο μεσίτης μπορεί να χρειαστεί να κάνει τις μετατροπές ανάμεσα στις οντολογίες και τους όρους που χρησιμοποιούνται από τα δύο μέρη. Για παράδειγμα ο αιτών μπορεί να έχει ζητήσει πληροφορίες από την IBM, όπου ο προμηθευτής περιμένει ότι τα εισαχθέντα δεδομένα θα είναι σε όρους της International Business Machine Corporation. Ένα πιο περίπλοκο παράδειγμα αποτελεί η περίπτωση όπου ο αιτών μπορεί να έχει ζητήσει το δελτίο καιρού στην Πράγα και αντί για αυτό ο προμηθευτής να μπορεί να διαθέσει το δελτίο καιρού στα μεγάλα αεροδρόμια. Η αρμοδιότητα του μεσίτη σε αυτή τη περίπτωση είναι να συμπεράνει πιο είναι το πιο

κατάλληλο αεροδρόμιο και να το χρησιμοποιήσει στο ερώτημα για τον προμηθευτή. Έτσι αντί να ζητήσει το δελτίο καιρού για τη Πράγα μπορεί να ζητήσει το δελτίο καιρού για το αεροδρόμιο της Πράγας.

Τελικά ο μεσίτης έχει την σημαντική αρμοδιότητα της μετάφρασης ανάμεσα στις διαφορετικές φόρμες συντακτικών που χρησιμοποιούνται για τα ερωτήματα και τις απαντήσεις. Τα παραπάνω παραδείγματα έχουν ως υπόθεση διαφορετική σημασιολογία ανάμεσα σε διαφορετικά μηνύματα τα οποία ο μεσίτης πρέπει να μεταφράσει και έπειτα να στείλει. Τα μηνύματα αυτά πρέπει να συνταχθούν κατάλληλα και παρόλη τη σημασιολογική ομοιότητα, τα μηνύματα θα μπορούσαν να γίνουν αντιληπτά με διαφορετικούς τρόπους. Σκοπός του μεσίτη είναι να λύσει τις συντακτικές διαφορές και να διαμορφώσει τα μηνύματα με τέτοιο τρόπο ώστε να είναι κατανοητά από όλα τα μέρη.

Εν κατακλείδι, ο μεσίτης εκτελεί ένα αριθμό από περίπλοκες αιτιολογικές εργασίες, ξεκινώντας από την εύρεση και καταλήγοντας στην μετάφραση και σύνταξη των μηνυμάτων. Για την επίτευξη αυτών των εργασιών ο μεσίτης χρειάζεται την υποστήριξη ενός επίσημου πλαισίου εργασίας που επιτρέπει την περίπλοκη αιτιολόγηση των πρακτόρων, τι μπορούν δηλαδή να κάνουν και πως μπορεί να αλληλεπιδράσει με αυτούς.

4.3.5 Ο Μεσίτης Υποστηρίζεται από την OWL-S

Η γλώσσα οντολογίας OWL-S παρέχει την υποδομή για να στηρίξει τον μεσίτη, τόσο στη διαδικασία της εύρεσης όσο και στη διαδικασία της μεσολάβησης ανάμεσα στις υπηρεσίες δικτύου. Το προφίλ της OWL-S υποστηρίζει τη διαδικασία της ανακάλυψης παρουσιάζοντας τις δυνατότητες των υπηρεσιών δικτύου και των πρακτόρων. Το Μοντέλο Διαδικασίας και η Περιοχή Γνώσης της OWL-S βοηθούν στην αλληλεπίδραση ανάμεσα στο μεσίτη, τον αιτών και τον προμηθευτή της υπηρεσίας.

Η Υποστήριξη δείχνει τη διαδρομή από τη σημασιολογική μορφή των ανταλλασσόμενων μηνυμάτων, όπως αυτή ορίζεται στο μοντέλο διαδικασίας, στη συντακτική μορφή όπως αυτή ορίζεται στις προδιαγραφές εισαχθέντων και εξαχθέντων δεδομένων της WSDL. Επιπλέον δείχνει τη διαδρομή ανάμεσα στη συνοψισμένη σημασιολογική παρουσίαση των μηνυμάτων και στη συντακτική μορφή

που αυτά τα μηνύματα υιοθετούν όταν μετατρέπονται σε πραγματικές ανταλλαγές πληροφοριών. Ο μεσίτης χρησιμοποιεί αυτή τη διαδρομή για να μεταφράσει τα μηνύματα που λαμβάνει καθώς και για να συνθέσει τα μηνύματα που στέλνει είτε στον αιτών είτε στον προμηθευτή.

Έχει προταθεί ένας σημαντικός αριθμός από αλγόριθμους που ταιριάζουν στην OWL-S και βασίζονται στις υπηρεσίες δικτύου, που αξιοποιούν τις οντολογίες της OWL και τις λογικές που σχετίζονται με αυτή για να μπορούν να συμπεράνουν ποιες δημοσιοποιήσεις ικανοποιούν μια αίτηση. Αυτοί οι αλγόριθμοι μπορούν να χρησιμοποιηθούν για την επίλυση του προβλήματος της συμφωνίας ανάμεσα στις δυνατότητες που απαιτούνται και στις δυνατότητες των διαθέσιμων προμηθευτών.

Ο διαχωρισμός του ερωτήματος από τις ικανότητες που απαιτούνται είναι πιο περίπλοκος. Καταρχήν δεν υπάρχει σαφής υποστήριξη στην OWL-S για τα ερωτήματα, επομένως είναι βολικότερο να χρησιμοποιήσουμε την OWL Query Language, η οποία βασίζεται στην ίδια λογική με την OWL-S. Ο μετασχηματισμός ωστόσο εξακολουθεί να είναι πρόβλημα, παρακάτω θα προτείνουμε έναν αλγόριθμο για το μετασχηματισμό των ερωτημάτων σε δυνατότητες.

Αφού έχει επιλεγεί ο προμηθευτής, ο μεσίτης έχει πρόσβαση στο Μοντέλο Διαδικασίας του από το οποίο μπορεί να αντλήσει το πρωτόκολλο αλληλεπίδρασης του προμηθευτή, με το να εξάγει τι πληροφορίες θα χρειαστεί ο προμηθευτής, σε τι σειρά και ποιες θα επιστρέψει. Όσον αφορά τα υπόλοιπα στοιχεία της αλληλεπίδρασης ο μεσίτης λειτουργεί ως απευθείας αιτών. Παρόλα αυτά η σχέση δεν είναι ευθύς. Εφόσον ο μεσίτης λειτουργεί εκ μέρους του αιτών, θα πρέπει με κάποιο τρόπο να μετασχηματίζει το αρχικό ερώτημα του αιτών, καθώς και όλα τα μηνύματα που προκύπτουν από αυτό, σε ένα ερώτημα ή μια σειρά ερωτημάτων προς τον προμηθευτή. Αυτός ο μετασχηματισμός είναι απαραίτητος εφόσον ο αιτών δεν έχει απευθείας πρόσβαση στο Μοντέλο Διαδικασίας του προμηθευτή, αλλά αλληλεπιδρά με αυτόν μόνο μέσω του μεσίτη.

Συνεχίζοντας μπορούμε να πούμε ότι εφόσον ο αιτών έχει υποβάλλει το ερώτημά του χωρίς να έχει πρόσβαση στο Μοντέλο Διαδικασίας, ο μεσίτης θα πρέπει να συμπεράνει ποιες επιπρόσθετες πληροφορίες χρειάζεται από τον αιτών. Αφού αυτό έχει πραγματοποιηθεί χρησιμοποιεί έπειτα αυτή τη γνώση για τη δημιουργία ενός νέου Μοντέλου Διαδικασίας. Το νέο αυτό Μοντέλο παρουσιάζεται από το Μεσίτη στον αιτών όχι σαν Μοντέλο Διαδικασίας του προμηθευτή αλλά του μεσίτη. Αυτό είναι λογικό εφόσον ο αιτών αλληλεπιδρά μόνο με τον μεσίτη. Το νέο αυτό μοντέλο δείχνει

στον αιτών ποιες πληροφορίες είναι απαραίτητες και σε τι σειρά. Αργότερα θα δούμε πως ο μεσίτης συμπεραίνει ποιες επιπρόσθετες πληροφορίες είναι απαραίτητες και πως δημιουργεί το νέο Μοντέλο Διαδικασίας.

Εφόσον για τον αιτών ο μεσίτης είναι αντιπρόσωπος του προμηθευτή, το Μοντέλο Διαδικασίας του μεσίτη θα πρέπει να περιέχει τα κρισιμότερα στοιχεία του Μοντέλου Διαδικασίας του προμηθευτή. Στο μεταξύ εφόσον ο μεσίτης δεν γνωρίζει τον προμηθευτή μέχρι να τον βρει και να τον επιλέξει, βασισμένος στο ερώτημα του αιτών, αντιμετωπίζει μία πρόκληση: θα πρέπει να δημοσιοποιήσει το μοντέλο διαδικασίας που εξαρτάται από το μοντέλο διαδικασίας του προμηθευτή, δεν είναι όμως γνωστό ποιος είναι αυτός, μέχρι ο αιτών να αποκαλύψει το ερώτημά του. Από την άλλη πλευρά ο αιτών δεν μπορεί να αλληλεπιδράσει με τον μεσίτη, να διατυπώσει δηλαδή κάποιο ερώτημα, μέχρι ο μεσίτης να δημοσιοποιήσει το μοντέλο διαδικασίας. Το αποτέλεσμα είναι ότι προκύπτει παράδοξο, το οποίο προκύπτει λόγω κάποιας ακαμψίας που παρουσιάζει η προδιαγραφή της OWL-S και αφορά την επίκληση μιας υπηρεσίας, η οποία απαιτεί το μοντέλο διαδικασίας πριν από την αλληλεπίδραση και δεν επιτρέπει-σε κανένα μέσο να μεταβάλλει το μοντέλο διαδικασίας κατά τη διάρκεια της αλληλεπίδρασης.

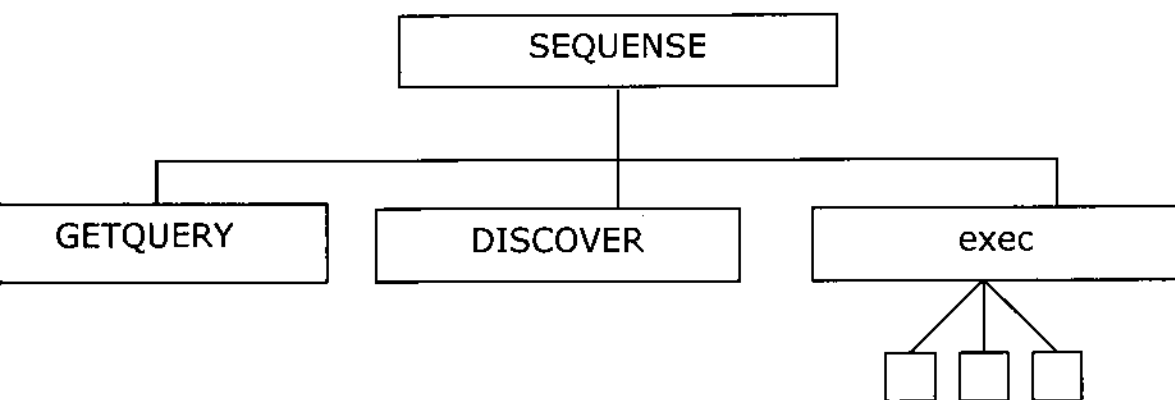
4.3.6 Επεκτείνοντας το OWL-S

Η λύση που προτάθηκε για το παράδοξο του μεσίτη, απαιτεί την επέκταση της προδιαγραφής του μοντέλου διαδικασίας της OWL-S, ώστε να επιτρέψει τη δυναμική μετατροπή του μοντέλου διαδικασίας ενός πράκτορα, κατά τη διάρκεια της αλληλεπίδρασης. Αυτό θα έχει σαν αποτέλεσμα να μπορεί ο μεσίτης να προσφέρει ένα αρχικό, ουδέτερο προμηθευτή, μοντέλο διαδικασίας στον αιτών και έπειτα να το μετατρέψει σύμφωνα με τις απαιτήσεις του μοντέλου διαδικασίας του προμηθευτή. Έτσι καταλήγουμε στο νέο μοντέλο διαδικασίας που χρησιμοποιεί ο αιτών όταν αλληλεπιδρά με τον μεσίτη.

Για τη πραγματοποίηση αυτής της λύσης πρέπει να επεκταθεί η γλώσσα του μοντέλου διαδικασίας της OWL-S, προσθέτοντας μία εντολή που καλείται `exec`. Η εντολή `exec` θεωρεί σαν εισαχθέν δεδομένο το μοντέλο διαδικασίας, το οποίο και εκτελεί. Για αυτό ο μεσίτης μπορεί να συντάξει ένα νέο μοντέλο διαδικασίας το οποίο επιστρέφει σαν εξαχθέν δεδομένο μιας από τις λειτουργίες του και έπειτα

χρησιμοποιεί το `exec` για να μετατρέψει το νέο μοντέλο διαδικασίας σε εκτελέσιμο κώδικα που προσδιορίζει το νέο πρωτόκολλο αλληλεπίδρασης του μεσίτη.

Το μοντέλο διαδικασίας του μεσίτη, το οποίο είναι ουδέτερου προμηθευτή φαίνεται στο Σχήμα 15.



Σχήμα 15 - Το Μοντέλο Διαδικασίας του Μεσίτη

Σύμφωνα με το σχήμα ο μεσίτης εκτελεί τρεις λειτουργίες: η πρώτη είναι το `get query` όπου ο μεσίτης λαμβάνει την ερώτηση από τον αιτών. Η δεύτερη είναι η εύρεση όπου ο μεσίτης χρησιμοποιεί τις ικανότητες εύρεσης προκειμένου να βρεθεί ο καλύτερος προμηθευτής. Το αποτέλεσμα αυτής της διαδικασίας είναι το νέο μοντέλο διαδικασίας το οποίο εξαρτάται από τον προμηθευτή που έχει βρεθεί. Τέλος ο μεσίτης εκτελεί τη λειτουργία του `exec`, η οποία περνάει τον έλεγχο στο νέο μοντέλο διαδικασίας. Αυτή η αλλαγή ελέγχου απεικονίζεται στο σχέδιο με τρία μικρά τετράγωνα που δείχνουν τις διαδικασίες που θα εκτελεστούν ως συνέπεια του `exec`.

Η χρήση του `exec` λύνει το πρόβλημα του παράδοξου αφαιρώντας τη δυσκαμψία του μοντέλου διαδικασίας της OWL-S. Η λειτουργία επιτρέπει το διαχωρισμό της υπηρεσίας εύρεσης από την υπηρεσία επίκλησης και αλληλεπίδρασης. Πρώτα ολοκληρώνεται η εύρεση και έπειτα ξεκινάει η αλληλεπίδραση, η οποία εξαρτάται από τον προμηθευτή που έχει ανακαλυφθεί, μέσω του `exec`.

Ένα σημαντικό ερώτημα που παραμένει αναπάντητο είναι το αν μπορεί να βρεθεί ένας τρόπος για να χρησιμοποιηθεί το OWL και η OWL-S χωρίς να είναι απαραίτητη η επέκταση της γλώσσας. Δυστυχώς μια τέτοια επέκταση δεν μπορεί να υπάρξει διότι το OWL καθώς και η OWL-S δεν παρέχουν τον τρόπο για το

μετασχηματισμό ενός όρου σε ένα κατηγορημα της λογική. Αυτό είναι το ουσιαστικό βήμα που πραγματοποιείται από το `exec`.

4.3.6.1 Η Επίσημη Σημασιολογία του `exec`

Τα σημασιολογικά στοιχεία της λειτουργίας του `exec` είναι η εκτέλεση των διαδικασιών που περιέχει. Με άλλα λόγια η αλλαγή της κατάστασης που δημιουργείται από το `exec (P)` είναι ίση με τη μεταβολή της συνθήκης που δημιουργείται από την απευθείας εκτέλεση του `P`. Αυτή την ιδέα συλλαμβάνει το αξίωμα που αναφέρεται στον Πίνακα 5, το οποίο είναι η φυσική επέκταση της αξιωματικής εκτέλεσης των σημασιολογικών στοιχείων της OWL-S που αναφέρονται στον Πίνακα 4.

<code>exec (P)</code>	$\Pi, (E[P], \varphi) \rightarrow \Pi', (E[P'], \varphi')$
	$\Pi, E[\text{exec}(P)], \varphi \rightarrow \Pi', (E[P'], \varphi')$

Πίνακας 5

Η εκτέλεση μιας εντολής `exec` φαίνεται στον Πίνακα 5. Ο κανόνας αυτός ορίζει ότι η εκτέλεση του `exec (P)` με συνθήκη (Π, φ) θα πρέπει να παράγει τα ίδια αποτελέσματα με αυτά που παράγει η εκτέλεση του `P` στην ίδια συνθήκη με τη συνθήκη (Π, φ) . Αυτός ο ορισμός μας επιτρέπει να μετατρέψουμε την προδιαγραφή μιας διαδικασίας `P` σε εκτέλεση της διαδικασίας, που είναι ακριβώς αυτό που ζητάμε με τον ορισμό του `exec`.

4.3.7 Η Εφαρμογή ενός Μεσίτη

Έχουμε υλοποιήσει το πρωτότυπο ενός μεσίτη που χρησιμοποιεί την OWL-S με την επέκταση του `exec` που περιγράφεται παραπάνω, για τη διαμεσολάβηση ανάμεσα σε πράκτορες και υπηρεσίες δικτύου. Βασίζουμε την εφαρμογή του μεσίτη στο OWL-S Virtual Machine, ο οποίος είναι ένας γενικός επεξεργαστής OWL-S που επιτρέπει στις υπηρεσίες δικτύου και στους πράκτορες να αλληλεπιδρούν με βάση

την προδιαγραφή της OWL-S για την υπηρεσία δικτύου και τις οντολογίες της OWL. Στην εφαρμογή του μεσίτη επεκτείνουμε την OWL-S VM, για να συμπεριλάβουμε τα σημασιολογικά στοιχεία του exec. Περαιτέρω αναπτύσσουμε την αιτιολογία που επιτρέπει στο μεσίτη να εκτελέσει την διαδικασία της εύρεσης και να μεσολαβήσει στην αλληλεπίδραση ανάμεσα στο προμηθευτή και τον αιτών.

4.3.7.1 Εύρεση Βασισμένη στο Μεσίτη

Ο μεσίτης αναμένει από τον αιτών ένα ερώτημα σε μορφή OWL-QL[47], όπου το κατηγορημα αντιστοιχεί σε μια ιδιότητα στην οντολογία, οι όροι στο ερώτημα είναι είτε μεταβλητές είτε στιγμιότυπα που είναι σύμφωνα με τις σημασιολογικές απαιτήσεις του κατηγορήματος.

Η διαδικασία της εύρεσης δέχεται σαν εισαχθέν δεδομένο το ερώτημα αυτού που αιτείται και παράγει σαν εξαχθέν δεδομένο την δημοσιοποίηση ενός προμηθευτή που μπορεί να απαντήσει στο ερώτημα. Η διαδικασία της εύρεσης αποτελείται από τρία βήματα. Αρχικά ο μεσίτης προσπαθεί να αποσπάσει από το ερώτημα τις ιδιότητες που απαιτούνται προκειμένου να απαντηθεί, για αυτό το λόγο δημιουργεί το αίτημα για μια υπηρεσία. Έπειτα ο μεσίτης βρίσκει τους κατάλληλους προμηθευτές ταιριάζοντας τις ικανότητες που απαιτούνται με τις ικανότητες των δημοσιοποιήσεων των προμηθευτών. Τέλος χρησιμοποιεί τα παραπάνω λαμβάνοντας υπόψη και άλλες παραμέτρους, ώστε να επιλέξει το καταλληλότερο προμηθευτή. Το ταίριασμα των ικανοτήτων πραγματοποιείται μέσω του OWL-S matching engine.

1. set V = set of variables in the query
2. set T= set of instantiated terms in the query
3. set I= abstraction of each term in T to its immediate class
4. use predicate definition in the ontology to abstract variables in V to their class
5. set O= abstraction of each variable in V to its class
6. generate a service request with input I and outputs O

Πίνακας 6 - Ο Αλγόριθμος Διαχωρισμού

Ο αυτόματος διαχωρισμός της αίτησης για κάποια υπηρεσία από το αρχικό ερώτημα του αιτών είναι ένα ανεξερεύνητο πρόβλημα. Στη διαδικασία διαχωρισμού πρέπει να λαμβάνονται υπόψη οι περιορισμοί της διαδικασίας εύρεσης της OWL-S, η δημιουργία ενός προφίλ υπηρεσίας OWL-S που περιλαμβάνει τα απαραίτητα εισαχθέντα και εξαχθέντα δεδομένα, τα οποία αντικατοπτρίζουν το σημασιολογικό περιεχόμενο του ερωτήματος καθώς και τις απαιτήσεις της αίτησης για υπηρεσία.

Ο αλγόριθμος περιλαμβάνει έξι βήματα, όπως περιγράφεται στο Πίνακα 6. Στα βήματα 1 και 2 εξάγονται οι όροι από το ερώτημα, διακρίνοντας ανάμεσα σε όρους μεταβλητών και στιγμιότυπων. Στο τρίτο βήμα το σύνολο από τα εισαχθέντα δεδομένα που προέκυψαν από την αίτηση για υπηρεσία, προέρχεται από το διαχωρισμό των στιγμιότυπων στα άμεσα τμήματα τους. Το τέταρτο βήμα περιλαμβάνει τη διαχείριση των μεταβλητών. Στην OWL-QL οι μεταβλητές είναι της τάξης *Variable*. Χρησιμοποιούμε τον ορισμό του κατηγορήματος για να περιορίσουμε τις τιμές των μεταβλητών στην πιο περιορισμένη τάξη τιμών που θα μπορούσαν να ανατεθούν. Στο πέμπτο βήμα-χρησιμοποιούμε το περιορισμό του τέταρτου βήματος για να δημιουργήσουμε ένα σύνολο από εξαχθέντα δεδομένα (O). Τέλος στο έκτο βήμα η αίτηση για υπηρεσία γεννάται από τον προσδιορισμό των εισαχθέντων και εξαχθέντων δεδομένων⁴.

4.3.8 Μεσολάβηση Βασισμένη στο Μεσίτη

Όταν ο μεσίτης επιλέξει έναν προμηθευτή πρέπει να μεσολαβήσει ανάμεσα στον αιτών και το προμηθευτή. Η διαδικασία της μεσολάβησης εξαρτάται από το μοντέλο διαδικασίας του προμηθευτή το οποίο ορίζει ποιες πληροφορίες απαιτούνται και πότε. Θεωρητικά ο μεσίτης μπορεί απλά να παρουσιάσει στον αιτών το μοντέλο διαδικασίας του προμηθευτή και να μεσολαβήσει, περιορισμένα, στην προώθηση του μηνύματος. Κάτι τέτοιο όμως θα ήταν ανεπαρκές διότι έτσι αγνοεί τη πληροφορία που του έχει περάσει ο αιτών. Παραδείγματος χάριν αν ο αιτών ζητήσει από το μεσίτη να του κλείσει ένα ταξίδι για τη Πράγα, ο δεύτερος μπορεί α του βρει μια υπηρεσία δικτύου για ταξίδια η οποία όμως χρειάζεται να γνωρίζει το τόπο αναχώρησης και

⁴ Τα δεδομένα που εισάγονται και εξάγονται αποτελούν τη πιο σημαντική πληροφορία για την ταυτοποίηση. Αν το ερώτημα περιέχει πρόσθετη πληροφόρηση, θα μπορούσε να αποσπαστεί.

άφιξης. Ο μεσίτης θα πρέπει να αναγνωρίσει ότι ο αιτών έχει προσδιορίσει το τόπο προορισμού, οπότε θα πρέπει να ερωτηθεί μόνο για το τόπο αναχώρησης.

1. KB= knowledge from query
2. I= input of process
3. for $i \in I$
4. select k from KB with the same semantic type of I
5. if k exists
6. remove i from I

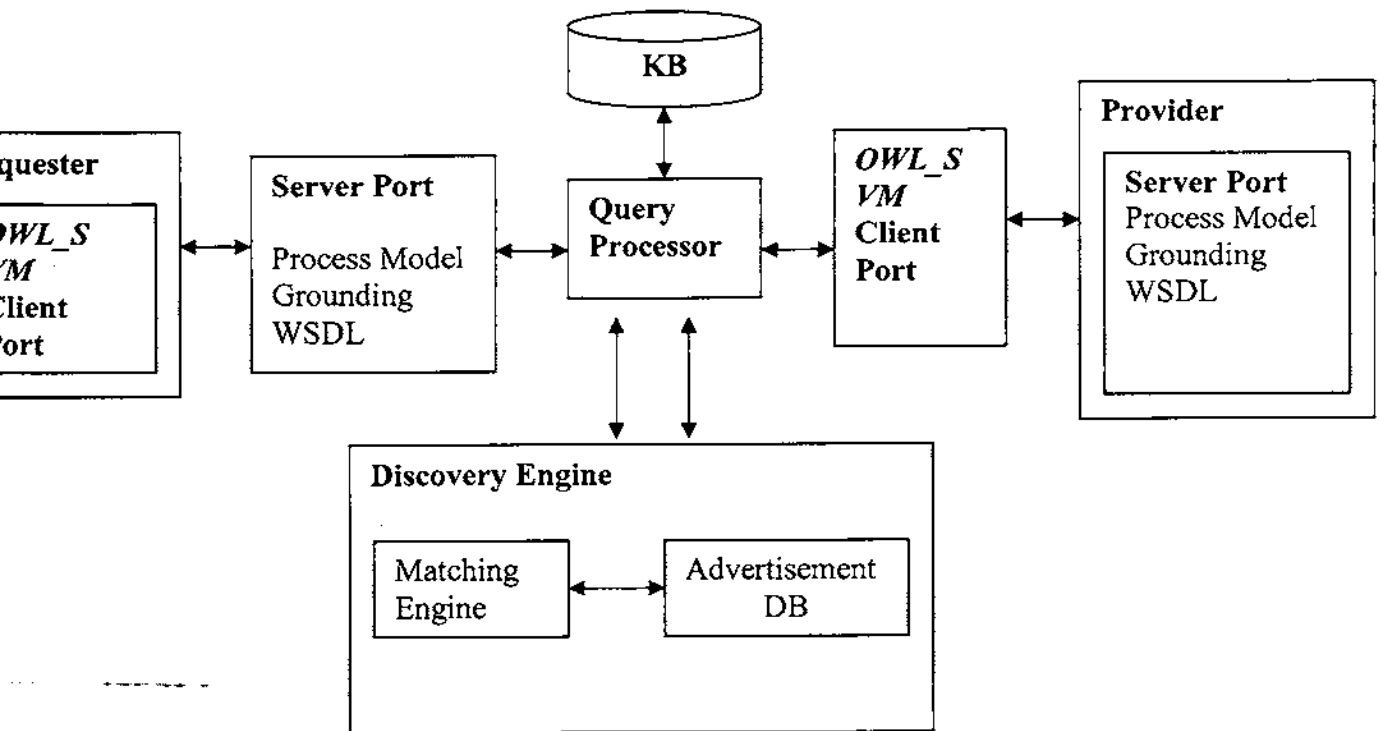
Πίνακας 7

Ο αλγόριθμος περιορισμού των πλεοναζόντων πληροφοριών φαίνεται στο Πίνακα 5. πρώτα ο μεσίτης καταγράφει τη πληροφορία που διατίθεται από το ερώτημα σε KB (βήμα 1) καθώς και τα εισαχθέντα δεδομένα της διαδικασίας (βήμα 2). Έπειτα για κάθε εισαχθέν δεδομένο i , ο μεσίτης ελέγχει στο KB για πληροφορίες που μπορούν να χρησιμοποιηθούν στη θέση του i . Αν βρεθεί κάποια το i αφαιρείται από τα εισαχθέντα δεδομένα της διαδικασίας.

4.3.9 Αλληλεπίδραση Βασισμένη στο Μεσίτη

Η αρχιτεκτονική ενός μεσίτη απεικονίζεται στο Σχήμα 16. Για να υπάρξει αλληλεπίδραση ανάμεσα στο προμηθευτή και τον αιτών ο μεσίτης χρησιμοποιεί δύο θύρες: μια θύρα εξυπηρετητή δικτύου την οποία χρησιμοποιεί για να αλληλεπιδρά με τον αιτών και μια θύρα πελάτη για να πραγματοποιείται η αλληλεπίδραση με τον προμηθευτή. Τις λειτουργίες της θύρας ενός εξυπηρετητή δικτύου μπορούμε να τις περιγράψουμε χρησιμοποιώντας την OWL-S. Συγκεκριμένα, ο μεσίτης εκθέτει στον αιτών το μοντέλο διαδικασίας, τη περιοχή γνώσης και την προδιαγραφή WSDL. Ο πελάτης (αιτών) χρησιμοποιεί τις περιγραφές αυτές στην αλληλεπίδραση του OWL-S Virtual Machine με τον μεσίτη. Εφόσον το μοντέλο διαδικασίας του ουδέτερου προμηθευτή εκτίθεται από το μεσίτη, χρησιμοποιείται η επέκταση του exec. Παράλληλα η χρήση του OWL-S Virtual Machine περιλαμβάνει την εφαρμογή των αξιωμάτων του exec. Η θύρα πελάτη εφαρμόζεται επίσης σαν OWL-S Virtual

Machine που χρησιμοποιεί το μοντέλο διαδικασίας, την υποστήριξη και την προδιαγραφή WSDL του προμηθευτή, προκειμένου να υπάρξει αλληλεπίδραση.



Σχήμα 16- Η Αρχιτεκτονική του Μεσίτη

Η αιτιολόγηση του μεσίτη γίνεται στη φάση της επεξεργασίας του ερωτήματος, όπως φαίνεται και στο Σχήμα 16. Εκεί γίνεται η μετάφραση των μηνυμάτων ανάμεσα στα δύο μέρη καθώς και η εφαρμογή των αλγορίθμων από τους πίνακες 4 και 5. Συγκεκριμένα ο επεξεργαστής ερωτημάτων αποθηκεύει τις πληροφορίες που έχει λάβει από το ερώτημα σε μια βάση γνώσης. Η βάση γνώσης χρησιμοποιείται όποτε κρίνεται απαραίτητο κατά τη διάρκεια της εκτέλεσης. Επιπλέον ο επεξεργαστής ερωτημάτων αλληλεπιδρά με το μηχανισμό εύρεσης, όποτε λαμβάνει κάποια δημοσιοποίηση ικανοτήτων και όποτε χρειάζεται να βρεθεί ένας προμηθευτής που θα μπορεί να απαντήσει στο ερώτημα του πελάτη.

7. Συμπέρασμα

Ανεξάρτητα από την ευρεία χρήση που γίνεται στους μεσίτες σε διαφορετικούς τομείς των διανεμημένων συστημάτων και ανεξάρτητα από τον αριθμό των χρήσεων που μπορούν οι μεσίτες να έχουν στα θέματα της εύρεσης και μεσολάβησης

υπηρεσιών δικτύου, δεν έχει γίνει καμία λεπτομερής ανάλυση πάνω στο θέμα των εργασιών που θα έπρεπε οι μεσίτες να εκτελούν. Μια τέτοια ανάλυση προσπαθήσαμε να κάνουμε παρουσιάζοντας τις προκλήσεις που υπάρχουν καθώς και τις λύσεις που θα μπορούσαν να προταθούν.

Η πρώτη πρόκληση είναι το παράδοξο του μεσίτη, το οποίο συμβαίνει όταν ο μεσίτης δεν μπορεί να εκδώσει το μοντέλο διαδικασίας που βασίζεται σε έναν μέχρι στιγμής άγνωστο προμηθευτή, μέχρι να λάβει το ερώτημα από τον αιτών, που με τη σειρά του δεν μπορεί να στείλει ένα ερώτημα αν δεν γνωρίζει το μοντέλο διαδικασίας του μεσίτη. Το παράδοξο αυτό προέρχεται από την προδιαγραφή αλληλεπίδρασης της OWL-S και βασίζεται στη προδιαγραφή ενός μοντέλου διαδικασίας που καθοδηγεί την αλληλεπίδραση του αιτών και του προμηθευτή. Για αυτό προτείναμε την επέκταση της γλώσσας του μοντέλου διαδικασίας με τη λειτουργία *exec*, που επιτρέπει τη δυναμική διαμόρφωση του μοντέλου διαδικασίας του μεσίτη κατά τη διάρκεια της εκτέλεσης του για να συμπεριλάβει τα μοντέλα διαδικασίας των νέων, δυναμικά ευρεθέντων, μελών. Παρέχουμε τη σημασιολογία για ένα χειριστή *exec*, η οποία βασίζεται στην επίσημη-σημασιολογία της OWL-S και δείχνουμε πως μπορεί να χρησιμοποιηθεί σαν βάση της OWL-S για να περιγράψει την αλληλεπίδραση περισσότερων από δύο μελών.

Το δεύτερο σύνολο προκλήσεων προέρχεται από τη διαχείριση της διαδικασίας της μεσολάβησης ανάμεσα στον προμηθευτή και τον αιτών. Προτείναμε την ανάπτυξη μιας μεθόδου διαχώρισης από την υπηρεσία ερωτήματος στην υπηρεσία επίκλησης, με την δημιουργία ενός αλγορίθμου. Επιπλέον προτείναμε έναν αλγόριθμο για τον μεσίτη ώστε να γίνεται αποτελεσματική χρήση της γνώσης που παρέχεται από τον αιτών κατά τη διάρκεια της αλληλεπίδρασης με τον προμηθευτή.

Τα ζητήματα που ανέρχονται σε ότι αφορά τη μεσολάβηση ανάμεσα στον προμηθευτή και τον αιτών δεν αφορούν μόνο τους μεσίτες αλλά παρουσιάζονται και στην ίδια τη σύνθεση των υπηρεσιών δικτύου. Στο πλαίσιο σύνθεσης των υπηρεσιών δικτύου ένας σχεδιαστής μπορεί να θέσει ένα στόχο τον οποίο να αναθέσει αλλού. Η υπηρεσία δικτύου σε αυτή τη περίπτωση πρέπει να αποσπάσει από το στόχο τις ικανότητες που απαιτούνται από έναν προμηθευτή για την επίτευξή του και έπειτα να χρησιμοποιήσει την υπάρχουσα γνώση και το στόχο για να αλληλεπιδράσει με τον προμηθευτή. Στη συγκεκριμένη έρευνα προσπαθούμε να ενοποιήσουμε την εργασία μας στο πλαίσιο του μεσίτη σε μια αυτόματη σύνθεση.[48]

4.4 Visual Modeling of OWL-S Services

4.4.1 Σύνοψη

Ο Σημασιολογικός Ιστός σταδιακά συγκεντρώνει το ενδιαφέρον και γίνεται πραγματικότητα. Περισσότεροι άνθρωποι το γνωρίζουν και προσπαθούν να χρησιμοποιήσουν τις τεχνολογίες του Σημασιολογικού Ιστού σε εφαρμογές. Αυτό αφορά τη χρήση εργαλείων που μπορούν να χειριστούν τη γρήγορη δημιουργία οντολογιών και την επικύρωση με ένα εύκολο και διαφανή τρόπο. Στην περιοχή του SemanticWeb Web Services (SWWS) μία OWL-S προδιαγραφή καθορίζει ένα σύνολο οντολογιών μέσα από το οποίο μπορεί να δημιουργηθεί η σημασιολογική περιγραφή μιας υπηρεσίας. Μερικές φορές αυτό δεν είναι απλό και μπορεί να έχει ως αποτέλεσμα μια λανθασμένη προδιαγραφή της περιγραφής ή μπορεί να οδηγήσει ένα άπειρο χρήστη να προσφύγει σ' ένα άλλο τύπο περιγραφής γλώσσας. Παρακάτω θα περιγράψουμε το εργαλείο OWL-S Editor, το οποίο παρέχει δύο μεθοδολογίες μέσα από τις οποίες οι περιγραφές υπηρεσιών Ιστού μπορούν να αναπτυχθούν χωρίς να εκθέτουν τον δημιουργό, στο βασικό OWL-S συντακτικό. Αυτές οι μεθοδολογίες βασίζονται στην αντιστοίχιση από τη WSDL στο OWL-S και στην μοντελοποίηση μιας σύνθετης υπηρεσίας χρησιμοποιώντας τα πρότυπα UML Activity Diagrams.

4.4.2 Αιτιολόγηση

Το παράδειγμα των Υπηρεσιών Ιστού αναδεικνύεται σε ένα σημαντικό πεδίο λόγω της διανεμημένης φύσης αυτής της τεχνολογίας. Πρόσφατα έχουμε δει τη γέννηση ενός αριθμού γλωσσών που παρέχουν στον ερευνητή, τους τρόπους και τα μέσα, να περιγράφουν και να διαφημίζουν τις υπηρεσίες στον Ιστό. Ορισμένες από τις γλώσσες έχουν γίνει δεκτές ως δεδομένα, ενώ άλλες ακόμα προσπαθούν να γίνουν αποδεκτές στο ευρύ κοινό.

Η WSDL είναι μια τέτοια γλώσσα και έχει γίνει δεκτή ως δεδομένη από το W3C. Αποτελεί και τη βάση για τη σύνθεση γλωσσών για υπηρεσίες Ιστού, όπως η BPEL4WS (Business Process Execution Language for Web Services 2002), WSCI

(Web Services Choreography Interface 2002) και OWL-S (Web Ontology Language for Services 2003). Αυτά παρέχουν στον δημιουργό των υπηρεσιών Ιστού τις δομές για τη δημιουργία μιας σύνθεσης υπηρεσιών, με μια πιθανή επαναχρησιμοποίηση ή μια επέκταση των υπάρχουσων. Η δημιουργία των περιγραφών για τις υπηρεσίες Ιστού με κάποια από τις γλώσσες αυτές απαιτεί τη βαθιά κατανόηση αυτών των γλωσσών, καθιστώντας τις όχι και τόσο δημοφιλείς σε έναν δημιουργό με λιγότερη εμπειρία. Αυτοί οι δημιουργοί δυσκολεύονται να βρουν εργαλεία που θα μπορούν να τους βοηθήσουν ή ακόμα να αυτοματοποιήσουν την διαδικασία δημιουργίας περιγραφών υπηρεσίας με ένα απλό και διαφανή τρόπο. Ένα σύνολο εργαλείων όπως η BPWS4J (IBM 2002) και το .NET (Microsoft), παρέχουν τέτοια λειτουργικότητα τόσο στον έμπειρο όσο και στον άπειρο χρήστη. Στην περιοχή του SWWS, τα εργαλεία είναι ένα σημαντικό θέμα από τη στιγμή όπου αρκετές συνδεδεμένες τεχνολογίες προσπαθούν να κάνουν τα πράγματα ακόμα πιο δύσκολα.

Το OWL-S, είναι προς το παρόν, η μόνη γλώσσα που συνδέει τη περιοχή του σημασιολογικού Ιστού με τις υπηρεσίες Ιστού. Μια τυπική περιγραφή υπηρεσίας σε αυτή τη γλώσσα περιέχει 4 οντολογίες, την υπηρεσία, το προφίλ υπηρεσίας, το μοντέλο υπηρεσίας και την υποστήριξη υπηρεσίας. Η ανώτερη οντολογία της υπηρεσίας συνδέεται με άλλες οντολογίες που κάνουν τη σημασιολογική περιγραφή. Η οντολογία του προφίλ μας δίνει μια περίληψη της υπηρεσίας, εκθέτοντας τα δεδομένα, τα αποτελέσματα, τις προϋποθέσεις και τις επιδράσεις όπως καθορίζονται στο μοντέλο διαδικασίας. Άλλοι τύποι πληροφοριών όπως η αναγνώσιμη περιγραφή, από ανθρώπους, περιεχομένου και οι πληροφορίες επαφών επιχειρήσεων εκφράζονται επίσης σ' αυτήν την τεχνολογία. Το μοντέλο υπηρεσίας καθορίζει τις διαδικασίες που συγκροτούν την υπηρεσία με τον καθορισμό των αποδεκτών παραμέτρων. Επιπροσθέτως, οι διαδικασίες μπορεί να είναι σύνθετης φύσης μέσα στην οποία ο σχεδιασμός των άλλων διαδικασιών περιγράφεται χρησιμοποιώντας δομές ελέγχου που καθορίζονται στην οντολογία του μοντέλου Υπηρεσίας. Τέλος, η οντολογία υποστήριξης της υπηρεσίας καθορίζει την αντιστοίχιση της ατομικής διαδικασίας και των παραμέτρων της με τις λειτουργίες και τα μέρη των μηνυμάτων που καθορίζονται στο WSDL της υπηρεσίας.

Για να δημιουργήσει μια περιγραφή OWL-S, ο χρήστης πρέπει να είναι σίγουρος για τη γλώσσα σήμανσης. Ένα από τα κύρια προβλήματα σε αυτή τη περίπτωση είναι ότι ο καθορισμός του OWL-S είναι ακόμα υπό ανάπτυξη και πολλές πτυχές της γλώσσας αυτής ακόμα αναπτύσσονται. Αυτές οι πτυχές, περιέχουν την

έκφραση συνθηκών, επιρροών και τους μηχανισμούς δέσμευσης της ροής δεδομένων. Η DRS και η SWRL μοιάζουν να είναι γλώσσες σήμανσης μέσα από τις οποίες οι συνθήκες θα εκφράζονται. Τέτοιες οντολογίες μπορούν να εκφράζουν λογικές εκφράσεις στο OWL και επιτρέπουν το καθορισμό των προϋποθέσεων και των επιρροών στην οντολογία του μοντέλου υπηρεσίας του OWL-S. Αυτές οι εξελίξεις στη γλώσσα δεν είναι εύκολο να γίνουν και να καθοριστούν, αφού απαιτούν ο χρήστης να έχει γνώσεις σε άλλα πεδία που σχετίζονται με τη λογική όπως και στο OWL-S.

Το εργαλείο OWL-S Editor, που παρουσιάζουμε εδώ παρέχει στο χρήστη μια απλοποιημένη διεπαφή, ούτως ώστε οι παραπάνω ιδέες να μπορούν να αναπτυχθούν γραφικά ώστε να μην χρειάζεται ο χρήστης να συντάσσει τις βασικές δομές του OWL-S. Ο Massimo Paolucci περιγράφει πως μια αρχική περιγραφή DAML-S μπορεί να αποκτηθεί από μια υπάρχουσα WSDL. Από τη στιγμή που το OWL-S χτίζεται πάνω στο DAML-S, οι ιδέες που παρουσιάζονται στο έγγραφο αυτό ισχύουν και για αυτό. Η μέθοδος μετατροπής θεωρείται από τον Paolucci ως μια αντιστοίχιση ένα προς ένα ανάμεσα στην ατομική διαδικασία του DAML-S (ή μια ατομική διαδικασία του OWL-S) και μια λειτουργία WSDL. Η λειτουργία αυτή θα εξελιχθεί σε μια ολοκληρωμένη προδιαγραφή υποστήριξης υπηρεσίας και σε ένα μερικώς ορισμό των οντολογιών του Προφίλ και του Μοντέλου της υπηρεσίας. Ένας πλήρης καθορισμός όλων των οντολογιών που περιέχονται στο OWL-S δεν μπορούν να αποκτηθούν από το WSDL επειδή η προηγούμενη είναι πλουσιότερη και εκφράζει μια πιο λεπτομερή περιγραφή της υπηρεσίας. Εκτελέσαμε αυτή τη μεθοδολογία στο εργαλείο OWL-S Editor με τη βοήθεια ενός οδηγού (OwlsWiz), ο οποίος αποσπά τις λεπτομέρειες της βασικής μετατροπής και επιτρέπει στο χρήστη να καθορίζει τις πληροφορίες που λείπουν, μέσω μιας γραφικής διεπαφής του χρήστη.

Συνθέτοντας μια υπηρεσία OWL-S με τη βοήθεια δομών ελέγχου που καθορίζονται στην οντολογία του μοντέλου υπηρεσίας, είναι άλλη μια δύσκολη ενέργεια. Ωστόσο, εργαλεία όπως το Protégé, μέσα από το OWL και τον Web Service Composer, καθιστούν δυνατή την εικονική σύνταξη των οντολογιών και των υπηρεσιών, δεν υπάρχουν εργαλεία που να καθιστούν ικανούς τους χρήστες να δημιουργήσουν εικονικά μια σύνθετη υπηρεσία χρησιμοποιώντας μια πρότυπη διαγραμματική τεχνική χωρίς να εκθέτουν στους χρήστες τις τεχνικές πληροφορίες που σχετίζονται με τη βασική σήμανση. Η UML Activity Diagrams είναι μια πρότυπη διαγραμματική τεχνική η οποία είναι κατάλληλη για να χειριστεί τις δομές και τις ιδέες

που καθορίζονται στην οντολογία του Μοντέλου Υπηρεσίας του OWL-S. Ως μέρος του OwlsWiz, το εργαλείου μας, εκτελεί μια εικονική σύνθεση η οποία κάνει χρήση των UML Activity Diagrams, για να ενεργοποιήσει την εικονική σύνθεση της υπηρεσίας και την αυτόματη δημιουργία της επεξεργασίας σήμανσης. Έτσι αποσπώνται οι βασικές λεπτομέρειες της δομής της οντολογίας του Μοντέλου Υπηρεσίας καθιστώντας δυνατή τη γρήγορη και απλή δημιουργία μιας σύνθετης υπηρεσίας. Ο χρήστης θα πρέπει ακόμα να γνωρίζει τα βασικά σημαντικά χαρακτηριστικά πίσω από το OWL-S, δηλαδή να έχει γνώση σε ότι αφορά τις προϋποθέσεις και τα αποτελέσματα, ωστόσο δεν είναι ανάγκη να γνωρίζει σε βάθος πως εκφράζονται αυτές οι έννοιες στη σήμανση.

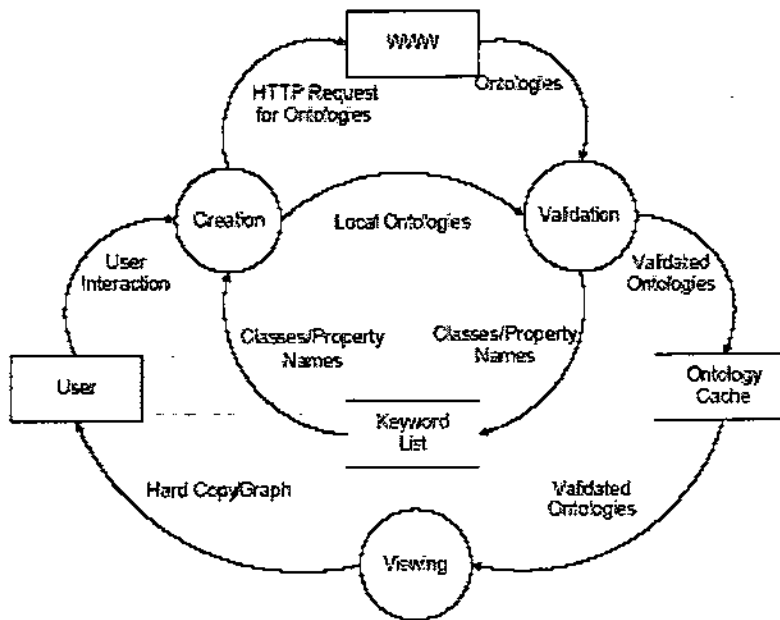
Το υπόλοιπο έγγραφο δομείται ως εξής. Πρώτα δίνουμε μια περίληψη του εργαλείου μας και μετά δίνουμε μια σε βάθος άποψη του OwlWiz και του Visual Composer κάνοντας αναφορές στην υπηρεσία του συστήματος Πληροφοριών Υγείας. Θα αναφερθούμε σε θέματα που σχετίζονται με την υιοθέτηση μιας διαγραμματικής παρουσίασης του Composer καθώς επίσης και σε άλλες λύσεις που σχετίζονται με το καθορισμό των συνθηκών και των επιρροών. Μετα θα αποτιμήσουμε το εργαλείο και θα κάνουμε αναφορά σ' άλλες σχετικές εργασίες.

4.4.3 Περίληψη

Το μεγαλύτερο μέρος της προσπάθειας έγινε για την ανάπτυξη του οδηγού και του εικονικού συνθέτη (visual composer), επιπλέον προτείνουμε μια γενική αρχιτεκτονική, όπως απεικονίζεται στο Σχήμα 17, ενός συστήματος μέσα στο οποίο θα μπορούν οι περιγραφές του OWL-S να δημιουργούν και να εκτιμηθούν. Η ιδέα είναι να θεωρήσουμε μια OWL-S περιγραφή ως ένα έργο εφαρμογής. Ο χρήστης δεν θα χρειαστεί να «φορτώσει» τις OWL-S οντολογίες μια προς μια, αλλά θα μπορεί να ανοίγει ένα μοναδικό φάκελο και θα μπορεί να ορίζει τα μονοπάτια και τα ονόματα των φακέλων των οντολογιών που αποτελούν τη περιγραφή. Επιπλέον θα δίνεται στον έμπειρο χρήστη η δυνατότητα να συντάσσει απευθείας τη σύνδεση, αν το επιθυμεί.

Ο χρήστης μπορεί να δημιουργήσει μια περιγραφή είτε από μια υπάρχουσα WSDL περιγραφή είτε από ένα καινούργιο πρότυπο, αν το επιθυμεί. Οι επιθυμητές non-local οντολογίες παίρνονται από το δίκτυο και επικυρώνονται πριν γίνει η

περαιτέρω διεργασία. Κατά τη διάρκεια του σταδίου επικύρωσης, οι τάξεις και οι μηχανισμοί αναγνώρισης ιδιοτήτων αποσπώνται και διατηρούνται σε μια λίστα εννοιών. Ο mark up editor χρησιμοποιεί αυτή τη λίστα για να εκτελέσει τη διαδικασία ολοκλήρωσης της σήμανσης κατά τη διάρκεια της σύνταξης. Οι οντολογίες που προκύπτουν μπορούν να επικυρωθούν και να προβληθούν είτε ως γράφημα είτε ως πίνακας. Αυτή η μέθοδος έχει υιοθετηθεί από την υπηρεσία RDF Validation (World Wide Web Consortium 2003).



Σχήμα 17 - Η Γενική Αρχιτεκτονική του OWL-S Editor

4.4.4 OWLSWIZ

Το κύριο χαρακτηριστικό του OWL-S Editor είναι ο οδηγός, που βοηθάει στη δημιουργία ενός πλήρους συνόλου OWL-S οντολογιών από μια WSDL περιγραφή. Ο οδηγός αυτός παρουσιάζει στο χρήστη τη διαδικασία δημιουργίας οντολογιών βήμα προς βήμα. Ο χρήστης αρχικά πρέπει να εισάγει τριών ειδών βασικές πληροφορίες:

- Τη διαδρομή για το αρχείο δεδομένων της WSDL
- Το ευρετήριο που θα αποθηκευτεί η περιγραφή του OWL-S
- Τη βασική URL από την οποία θα αναπτυχθεί η περιγραφή

Το βασικό URL ευρετήριο είναι πολύ σημαντικό, διότι θα χρησιμοποιηθεί στη δημιουργία των URI αναφορών για τις διάφορες οντολογίες που συνθέτουν την

OWL-S περιγραφή. Αν το URL είναι λάθος, τότε δεν θα είναι επιτυχής η επικύρωση ωστόσο η περιγραφή θα έχει ούτως η άλλως δημιουργηθεί. Το δεύτερο βήμα περιλαμβάνει τη δημιουργία μιας ανώτερης οντολογίας υπηρεσίας. Εδώ ο χρήστης μπορεί να εισάγει ένα γενικότερο σχόλιο για την υπηρεσία και να προσθέσει επιπλέον συγκεκριμένες αναφορές στα URI που εισάγονται, αν η υπηρεσία πρόκειται να χρησιμοποιήσει άλλες οντολογίες. Όταν ο χρήστης βρίσκεται στο βήμα δημιουργίας του Προφίλ πρέπει να εισάγει τέσσερα βασικά είδη πληροφοριών:

- Ένα αναγνώσιμο σε ανθρώπους όνομα υπηρεσίας (Service Name)
- Μια αναγνώσιμη σε ανθρώπους περιγραφή της υπηρεσίας
- Πληροφορίες επικοινωνίας με την οντότητα που παρέχει την υπηρεσία
- Μία URL που αναφέρεται στη κατάταξη της υπηρεσίας

Οι πληροφορίες επικοινωνίας εκφράζονται ως μια μετατρέπόμενη οντολογία VCard. Αυτή η οντολογία έχει απλουστευτεί για να μπορούν να εκφράζονται οι απλοί τύποι πληροφοριών όπως το όνομα, ο τίτλος, το e-mail και ο τηλεφωνικός αριθμός. Για την κατάταξη της υπηρεσίας θεωρούμε ότι υπάρχει κάποια άλλη υπηρεσία η οποία παρέχει πληροφορίες κατάταξης για τις υπηρεσίες.

Όταν δημιουργούμε το Μοντέλο Διαδικασίας, οι ατομικές διαδικασίες και τα δεδομένα που εισάγονται και εξάγονται αποσπώνται από την WSDL. Αυτή η διαδικασία όμως απαιτεί περαιτέρω αλληλεπίδραση με το χρήστη όπως φαίνεται και στο Σχήμα 18. Σε αυτό το στάδιο ο χρήστης μπορεί να χρησιμοποιήσει το Visual Composer για να δημιουργήσει μια σύνθετη διαδικασία, συνδέοντας τις αποσπώμενες ατομικές διαδικασίες.

Κατά τη διάρκεια δημιουργίας της οντολογίας Υποστήριξης, δεν απαιτείται αλληλεπίδραση με το χρήστη λόγω του ότι η οντολογία δημιουργείται ολοκληρωτικά από την WSDL. Το τελευταίο βήμα στον οδηγό ενημερώνει απλά το χρήστη ότι η OWL-S περιγραφή έχει δημιουργηθεί επιτυχώς και δείχνει επίσης τις πληροφορίες που αφορούν τα ονόματα αρχείων των οντολογιών.



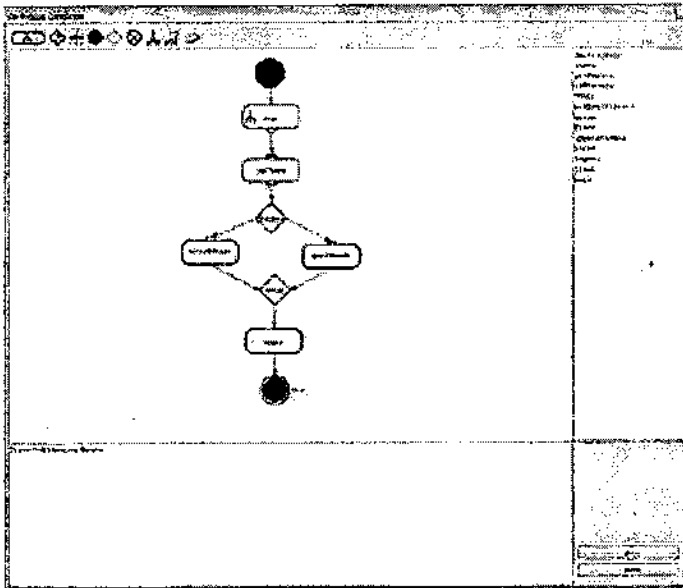
Σχήμα 18 Φωτογραφία Οθόνης του OwlsWiz κατά το βήμα Δημιουργίας του Μοντέλου Διαδικασίας

4.4.5 Visual Composer

Ο Visual Composer προσεγγίζεται από τον οδηγό κατά τη διάρκεια της δημιουργίας του Μοντέλου Υπηρεσίας. Η ιδέα είναι να χρησιμοποιεί τις αποσπώμενες ατομικές διαδικασίες και τα δεδομένα/αποτελέσματα για να τα συνθέσει γραφικά και να δημιουργήσει αυτόματα τη σήμανση. Η γραφική παρουσίαση που υιοθετείται από τον συνθέτη περιέχει τη χρήση των UML Activity Diagrams και οι δομές που έχουν εκτελεστεί ως τώρα είναι: Sequence, If-Then-Else και Split. Αυτές οι δομές έχουν αντιστοιχηθεί με ειδικούς παράγοντες σε αυτή τη διαγραμματική παρουσίαση. Για να γίνει η διαφοροποίηση ανάμεσα στους διαφορετικούς τύπους διαδικασιών στο OWL-S, έγιναν κάποιες γραφικές αλλαγές στο στοιχείο Action. Για να δείξουμε πως τα διαγράμματα αντιστοιχούν στις δομές ελέγχου του OWL-S και στις διαδικασίες θα χρησιμοποιούμε μια εικονική υπηρεσία Πληροφοριών Υγείας της οποίας η σύνθεση φαίνεται στο Σχήμα 19. Ένας ασθενής μπορεί να κάνει log-in στο σύστημα και να έχει πρόσβαση σε δύο διαφορετικές υπηρεσίες. Η μια υπηρεσία θα επιστρέφει το ιατρικό ιστορικό του ασθενή και η άλλη θα επιστρέφει τη λίστα των ιατρικών ραντεβού.

Διαδικασίες όπως το login, get Choice, medical history, appointments και logout αναπαρίστανται ως Action διαγράμματα. Όταν υπάρχει ροή δεδομένων

ανάμεσα στις επεξεργασίες, χρησιμοποιείται η έννοια pins. Η επεξεργασία get Choice έχει μαζί τη ροή των δεδομένων και των αποτελεσμάτων. Οι επεξεργασίες του OWL-S μπορεί να έχουν προϋποθέσεις και επιρροές. Αυτές τις παρουσιάζουμε με το σύμβολο take. Το σύμβολο φυσιολογικά χρησιμοποιείται για να δείξει ότι υπάρχουν κάποιες βασικές υποδραστηριότητες που ενισχύουν τη συγκεκριμένη ενέργεια. Η δομή If-Then-Else, παρουσιάζεται χρησιμοποιώντας το σύμβολο Decision. Μια τέτοια δομή θα πρέπει να είναι συνοδευμένη από το στοιχείο Merge, με σκοπό να προσχωρήσει στη προδιαγραφή UML 2.0 η οποία δηλώνει ότι κανένα action στοιχείο δεν μπορεί να έχει περισσότερες από μια εισόδους δεδομένων. Αν και το παράδειγμα του Σχήματος 19 δεν χρησιμοποιεί δομή ελέγχου διαχωρισμού, το τελικό αντιστοιχείται με το σύμβολο fork στην ίδια παρουσίαση.



user tool

Σχήμα 19 Φωτογραφία Οθόνης για το Εργαλείο του Visual Composer

4.4.5.1 Εκφράζοντας Συνθήκες και Επιρροές

Δεν υπάρχει καμία προδιαγραφή στο OWL-S που να καθορίζει ξεκάθαρα πως θα πρέπει να εκφράζονται οι συνθήκες και οι επιρροές. Αυτό είναι κάτι που αλλάζει

στο OWL-S 1.1, όπου η δομή Expression καθορίζει την ένωση ανάμεσα στην Atomic List (SWRL) και στην Formula (DRS).

Ο Visual Composer υιοθετεί ένα υποσύνολο αυτών των γλωσσών με σκοπό να παρουσιάσει στο χρήστη απλά σενάρια, στα οποία οι συνθήκες, οι προϋποθέσεις και οι επιρροές μπορούν να εκφραστούν. Δεδομένου του μικρού χρονικού διαστήματος για να αναπτυχθούν αυτοί οι παράγοντες, δεν θα μπορούσαμε να εκμεταλλευτούμε τη πλήρη λειτουργικότητα των αναφερόμενων γλωσσών, αντιθέτως θα υιοθετήσουμε ικανοποιητικές λύσεις για τις πιο κοινές περιπτώσεις στις οποίες οι έννοιες που εκφράζονται χρησιμοποιούνται.

Η συνθήκη της δομής If-Then-Else συγκρίνει τη παράμετρο αποτελέσματος μιας διαδικασίας με μια τιμή. Προς το παρόν ο οδηγός δεν εκτελεί τον μετασχηματισμό από τους σύνθετους τύπους δεδομένων XSD σε OWL έννοιες και αυτό περιορίζει τη σύγκριση των παραμέτρων με τις τιμές των ατομικών τύπων δεδομένων όπως εκφράζονται στο XML Schema Datatype Specification. Από εδώ και πέρα ο χρήστης θα πρέπει να επιβεβαιώνει ότι η τιμή η οποία θα συγκριθεί με την παράμετρο, είναι του ίδιου τύπου. Προαιρετικά η συνθήκη του If-Then-Else μπορεί να συγκρίνει τα αποτελέσματα των παραμέτρων των δύο διαδικασιών. Και στις δύο περιπτώσεις, οι συνθήκες βασίζονται στη σύγκριση μιας μαθηματικής ισότητας της λειτουργίας και ο χρήστης θα πρέπει να επιβεβαιώνει ότι οι παράμετροι είναι συγκρίσιμοι με την επιλεχθείσα λειτουργία. Αυτός ο τύπος εκφράζεται με μια δομή DRS Atomic Formula όπου το υποκείμενο είναι μια αναφορά στο αποτέλεσμα της παραμέτρου της επεξεργασίας, το κατηγορημα είναι μια αναφορά στην ισότητα της λειτουργίας και το αντικείμενο καθορίζεται είτε ως μια τιμή είτε ως μια αναφορά σε άλλες παραμέτρους αποτελεσμάτων. Η σήμανση που καθορίζεται στην οντολογία Μοντέλου Επεξεργασίας είναι ισότιμη με τη δομή που δημιουργείται για τη συνθήκη της δομής is History If-Then-Else στο Σχήμα 19. Αυτή η φόρμουλα δηλώνει μια μεταβλητή του ίδιου τύπου όπως και για την παράμετρο των αποτελεσμάτων από την επεξεργασία get Choice και τη συγκρίνει με μια τιμή 0 όπως καθορίζεται από το χρήστη. Αν η συνθήκη είναι σωστή, τότε η υπηρεσία επιτρέπει στον ασθενή να έχει πρόσβαση στην υπηρεσία Medical History.

Για προϋποθέσεις και επιρροές, ο δημιουργός υιοθετεί μια διαφορετική λύση από τις παραπάνω. Υποθέτουμε, ότι υπάρχει μια οντολογία που καθορίζει τα κατηγορήματα και τις τάξεις που χρειάζονται οι προϋποθέσεις και οι επιρροές, που καθορίζονται από τη συγκεκριμένη υπηρεσία. Λαμβάνοντας υπόψη την επεξεργασία

login, στην υπηρεσία HIS, μια τυπική προϋπόθεση θα περιείχε ένα έγκυρο όνομα χρήστη.

Βασιζόμενοι στο ποιες είναι οι προϋποθέσεις ή οι επιρροές που περιγράφονται, ένα κατηγορημα μπορεί να έχει δύο ορισμούς από ότι ένα. Παραδείγματος χάριν η επεξεργασία login μπορεί να έχει ως αποτέλεσμα το loginUser, που χρησιμοποιεί το όνομα χρήστη και την πιστοποίηση για ν' αφήσει το χρήστη να μπει στο σύστημα. Η ιδέα είναι να χρησιμοποιήσουμε μία IndividualPropertyAtom δομή όπως καθορίζεται στην SWRL Αυτή η δομή αναφέρεται στο Property Predicate και στους ορισμούς του κατηγορήματος, ονομαστικά ως argument1 και argument2. Και στις δύο περιπτώσεις, οι μεταβλητές πρέπει να δηλώνονται.

4.4.5.2 Εκφράζοντας τη Ροή Δεδομένων

Η τρέχουσα έκδοση του OWL-S παρέχει ένα απλό τρόπο διαχείρισης της ροής δεδομένων. Η ιδέα είναι να χρησιμοποιούμε μια ιδιότητα same Values η οποία καθορίζει τη δομή Value of, η οποία αντιστοιχεί τις παραμέτρους της επεξεργασίας. Το πρόβλημα με αυτή τη τεχνική είναι ότι αν το στιγμιότυπο της ίδιας διαδικασίας χρησιμοποιείται σε διαφορετικά μέρη μιας χορογραφίας, η αντιστοίχιση των παραμέτρων από τις διαφορετικές κλήσεις στην επεξεργασία, μπορούν να διακριθούν. Η πρόταση απεικόνισης για την έκδοση 1.1 τοποθετεί αυτό το θέμα μέσα από την χρήση της δομής ελέγχου Perform, η οποία παρουσιάστηκε στο τέλος του σταδίου ανάπτυξης του Visual Composer, σ' ένα σημείο όπου ήδη αναζητούσαν μια διαφορετική λύση. Η ιδέα είναι βασικά ίδια με την προαναφερόμενη δομή, αλλά η ιδέα της «ετικέτας» χρησιμοποιείται.

Η χρήση της τεχνικής απαιτεί μια οντολογία η οποία καθορίζει τις δομές που χρησιμοποιούνται. Τα ονόματα «ετικέτας» καθορίζονται στην δομή TagBind, η οποία είναι μια υποκατηγορία του ControlConstruct. Στη σειρά των δομών ελέγχου ένα σύνολο από Call δομές ορίζει μια αναφορά στην επεξεργασία που θα εκτελεστεί καθώς επίσης στα συνδεδεμένα tag names που θα χρησιμοποιηθούν για να δεσμεύσουν τις παραμέτρους της επεξεργασίας. Μαζί με το σύνολο των Call δομών, άλλο ένα σύνολο από Dataflow δομές εκφράζεται. Ο αριθμός τέτοιων δομών θα εξαρτηθεί από το πόσες δεσμεύσεις ροής δεδομένων υπάρχουν ανάμεσα στις

καθορισμένες επεξεργασίες. Για κάθε απλή δέσμευση, μια αντίστοιχη Dataflow δομή θα εκφράζεται. Αυτή η δομή καθορίζει τις ιδιότητες source και destination. Οι δύο αυτές ιδιότητες κατατάσσονται στην τάξη Parameter Inst. Οι ιδιότητες που καθορίζονται σε αυτή τη τάξη περιέχουν το όνομα της «ετικέτας» και μια αναφορά στην σχετική διαδικασία καθώς και στο κατά πόσο η «ετικέτα» καθορίζει τη ροή δεδομένων και των αποτελεσμάτων. Η σήμανση του εικονικού μοντέλου Επεξεργασίας HIS δείχνει πως μια τεχνική μπορεί να χρησιμοποιηθεί για να καθορίσει αυτούς τους τύπους ροής δεδομένων.

4.4.6 Αποτίμηση

Τα περιθώρια του χρόνου ήταν πολύ στενά και υπήρχαν πολλά θέματα για τα οποία δεν είχαμε τον έλεγχο. Εκτός από αυτό, το API που μετασχηματίζει τη WSDL στο OWL-S δεν ήταν διαθέσιμο για να «φορτωθεί» και θα έπρεπε να υιοθετήσουμε μια δική μας λύση. Κάναμε την υπόθεση ότι τα δεδομένα στην περιγραφή WSDL δεν καθορίζουν κανένα τύπο. Αυτό περιορίζει τον εκδότη στο να χειριστεί τους ατομικούς τύπους δεδομένων όπως καθορίζονται στο XSD Schema Datatype Specification. Ο οδηγός έχει ελεγχθεί σε ακραίες συνθήκες όπου τα δεδομένα της WSDL περιέχουν αυτούς τους τύπους των στοιχείων. Το αποτέλεσμα ήταν ότι οι πληροφορίες που σχετίζονταν με τις ατομικές διαδικασίες και τις παραμέτρους είχαν σωστά αποσπαστεί, χωρίς όμως να κατανοούν τους σύνθετους τύπους δεδομένων όπως στις οντολογίες OWL. Αυτό αποδεικνύει ότι ο βασικός μηχανισμός απόδοσης αυτής της αντιστοίχισης είναι στη πραγματικότητα δυνατός.

Οι οντολογίες που έχουν γίνει έχουν επικυρωθεί χρησιμοποιώντας το RDF Validation Service και το OWL Ontology Validator. Όταν επικυρώνονται οι βασικές οντολογίες OWL-S, λάθη όπως "Not a valid OWL DL Subgraph", έχουν καταμετρηθεί. Αυτό το λάθος γίνεται επειδή το OWL-S εμπίπτει στο OWL-Full. Σε ότι αφορά τις οντολογίες του μοντέλου επεξεργασίας, εργαζόμαστε ακόμα πάνω σε θέματα που αφορούν τον τρόπο με τον οποίο οι συνθήκες, οι επιρροές και η ροή δεδομένων εκφράζονται. Τις περισσότερες φορές βασίζαμε την εργασία μας σε προσχέδιες προτάσεις που έχουν περιγραφεί σε εργασίες (όπως DRS και SWRL προτάσεις) και σε λίστες αλληλογραφίας (Semantic Web Interest Group). Επιπλέον θα πρέπει να λάβουμε υπόψη μας τις νέες εκδόσεις της OWL-S 1.1, ειδικά τις

εκφράσεις και τους μηχανισμούς δέσμευσης της ροής δεδομένων. Εργαζόμαστε ακόμα σε αυτό το κομμάτι για να δημιουργήσουμε οντολογίες οι οποίες είναι συμβατές με το OWL-S 1.1.

Η χρήση των API είναι επίσης περιορισμένη. Κατά τη διάρκεια των αρχικών σταδίων της ανάπτυξης του εργαλείου, το JenaToolkit (HP Semantic Web Program 2004) δεν υποστήριζε το OWL Full. Για αυτό προτιμούμε να εκτελούμε δικά μας απλά API για να επιτύχουμε τους στόχους μας. Παρόμοια, το OWL-S API (Evren Sirin 2004) έχει πρόσφατα αποδεσμευτεί σε τέτοιο βαθμό, όπου πολλές από τις τάξεις που χρειαζόμασταν ήταν έτοιμες και δεν ήταν εφικτό σε μας να αλλάξουμε σε αυτό το API. Παρόλα αυτά ολοκληρώσαμε το JenaToolkit (version 2.1 με την υποστήριξη του OWL Full) το οποίο επικυρώνει τις οντολογίες που έχουν δημιουργηθεί από τον οδηγό.

Ο editor γενικά, είναι περιορισμένος σε κάποια εκδοτικά χαρακτηριστικά. Επίσης απουσιάζει η έξυπνη διαχείριση λαθών και το σύστημα επικύρωσης. Το εργαλείο περιορίζεται στο να δείχνει λάθη σε ξεχωριστό παράθυρο από το να δείχνει σε πραγματικό χρόνο τα λάθη ή τα προβλήματα που μπορεί να έχουν οι οντολογίες.

4.4.7 Σχετική Εργασία

Προς το παρόν ο αριθμός των διαθέσιμων εργαλείων για τη δημιουργία OWL-S περιγραφών είναι περιορισμένος, ωστόσο περιμένουμε ότι αυτός ο αριθμός θα αυξηθεί τους επόμενους μήνες, καθώς η προδιαγραφή του OWL-S σταθεροποιείται και περισσότερη έρευνα θα έχει ολοκληρωθεί. Έχουμε βρει ήδη δύο εργαλεία που σχετίζονται με τον editor: το Web Service Composer και το Protégé OWL Plugin (Holger Krubnauch 2004).

Το Web Service Composer αποτελείται από δύο υπομονάδες, ένα οπτικό συνθέτη και μια μηχανή συμπερασματολογίας. Ο προηγούμενος χρησιμοποιείται ως μια διεπαφή ανάμεσα στο χρήστη και το μηχανισμό και παρέχει τα μέσα που δυναμικά δημιουργούν μια ροή ανάμεσα σε ένα αριθμό υπηρεσιών. Ο μηχανισμός συμπερασματολογίας βασίζεται στο Prolog και διαχειρίζεται οντολογίες OWL, μετασχηματίζοντας τις πληροφορίες σε RDF τριάδες και τ' αποθηκεύει στη βάση γνώσης του συνθέτη. Ο μηχανισμός χρησιμοποιεί ένα σύνολο ενσωματωμένων αξιωμάτων για να παρέχουν στην υπηρεσία τη δυνατότητα ταυτοποίησης. Παρέχει

δύο τύπους ταυτοποίησης, ονομαστικά, μια ακριβή και μια υποθετική. Ο συνθέτης χρησιμοποιεί ένα μηχανισμό φιλτραρίσματος για να περιορίσει τον αριθμό υπηρεσιών που επιστρέφονται ύστερα από τη διαδικασία της ταυτοποίησης. Η οπτική διεπαφή βοηθά το χρήστη στη δημιουργία μιας ροής ανάμεσα στις υπηρεσίες που συνθέτονται και δημιουργεί αυτόματα, το μοντέλο επεξεργασίας και το προφίλ αντιστοίχισης. Το εργαλείο βοηθάει επίσης στην εκτέλεση της υπηρεσίας που συνδέθηκε, με το να επικαλείται τις ατομικές υπηρεσίες και μεταφέροντας δεδομένα ανάμεσα στις υπηρεσίες σύμφωνα την καθορισμένη ροή από τον χρήστη.

Η γενική οντολογία Protégé καθιστά δυνατή την ξεχωριστή δημιουργία των οντολογιών που δημιουργούν την OWL-S περιγραφή της υπηρεσίας. Το εργαλείο αυτό παρουσιάζει γραφικά τις ιδιότητες και τις τάξεις που μπορούν να δημιουργηθούν. Ο χρήστης πρέπει να είναι γνώστης των δομών που καθορίζονται στις οντολογίες OWL-S για να μπορεί να δημιουργήσει μια ολοκληρωμένη περιγραφή, και αυτό είναι ένα μειονέκτημα. Όταν οι εισαγωγές δεδομένων προστίθενται στην οντολογία, οι δεδομένοι παράγοντες εισάγονται χρησιμοποιώντας την εσωτερική αιτιολόγηση. Αυτοί οι παράγοντες παρουσιάζονται σε δομή δένδρου και αυτό διευκολύνει την κατανόηση της ιεραρχίας των τάξεων που αποσπώνται από τις εισαγόμενες οντολογίες. Επίσης για τη σύνθεση υπηρεσιών δεν υπάρχει καμία πρότυπη γραφική παρουσίαση που χρησιμοποιείται.

4.4.8 Μελλοντική Εργασία

Σκοπεύουμε να επεκτείνουμε αυτό το εργαλείο για να μετασχηματίσουμε τους σύνθετους τύπους δεδομένων XSD στις έννοιες του OWL. Αυτό απαιτεί μελλοντική εργασία σε άλλα μέρη του συνθέτη, όπως στα κομμάτια όπου οι παράμετροι επεξεργασίας θα συγκρίνονται με ειδικές τιμές.

Ο μηχανισμός αιτιολόγησης είναι ένα άλλο σημαντικό χαρακτηριστικό που σχεδιάζουμε να προσθέσουμε στο εργαλείο. Αυτό το χαρακτηριστικό θα επέτρεπε την κατηγοριοποίηση και την υπαγωγή των οντολογιών που χρησιμοποιούνται από την υπηρεσία. Μπορεί να χρησιμοποιηθεί για να υποθέσουμε αυτόματα τις υπολογισμένες προϋποθέσεις και τις επιρροές της επεξεργασίας. Αυτή είναι μια περιοχή στο OWL-S, η οποία ερευνάται, ελπίζουμε ότι η μελλοντική έκδοση θα καλύπτει αυτά τα θέματα.

Ο Visual Composer μπορεί να αναπτυχθεί περαιτέρω για να διαχειρίζεται τις άλλες δομές ελέγχου του OWL-S. Όλες οι δομές που είναι καθορισμένες στο OWL-S μπορούν να παρουσιαστούν χρησιμοποιώντας τα UML Activity Diagrams. Ορισμένες δομές είναι αναμφισβήτητα εκφρασμένες στον Visual Composer (όπως οι Sequence και οι μηχανισμοί ροής δεδομένων), ενώ οι υπόλοιποι απαιτούν τη χρήση σαφών δομών (όπως τα Split, If-Then-Else, Split+Join κ.τ.λ). Το composer μπορεί να δημιουργεί μια μόνο σύνθετη επεξεργασία και θα μπορούσε λοιπόν να επεκταθεί για να κάνει δημιουργεί πολλαπλές συνθέσεις επεξεργασιών.

Από τη στιγμή που τα OWL-S API είναι διαθέσιμα, ένας μικρός επανασχεδιασμός θα μπορούσε να πραγματοποιηθεί με σκοπό να χρησιμοποιηθεί η επίσημη βιβλιοθήκη προτύπων. Παρόλα αυτά το OWL-S Editor παρουσιάζει λύσεις με σεβασμό στις συνθήκες και τη ροή δεδομένων, που ακόμα δεν έχουν εκτελεστεί σε αυτό το API. Η ιδέα είναι να επεκταθούν οι απαιτούμενες τάξεις της βιβλιοθήκης προκειμένου να διατηρηθούν τα χαρακτηριστικά που παρουσιάζονται από αυτό το εργαλείο.

Η διεπαφή του χρήστη μπορεί γενικά να βελτιωθεί με διάφορους τρόπους. Ένα χαρακτηριστικό που θα προστεθεί είναι ένα σύστημα βοήθειας στο περιεχόμενο, όπου ο χρήστης μπορεί ν' αποκομίσει την απαιτούμενη βοήθεια στο συγκεκριμένο βήμα από τον οδηγό. Αυτό θα διευκόλυνε τη χρήση του εργαλείου και την ίδια στιγμή θα έδινε τη δυνατότητα στο χρήστη να γίνει βαθύτερος γνώστης του OWL-S. Άλλο ένα χαρακτηριστικό είναι το εργαλείο ολοκλήρωσης σήμανσης όπου το editor προτείνει τη λίστα των τάξεων που μπορούν να χρησιμοποιηθούν σε μια αναφερόμενη οντολογία. Επιπροσθέτως ο μηχανισμός αιτιολόγησης μπορεί να φιλτράρει αυτές τις τάξεις με σκοπό να παρέχει αυτές τις τάξεις και τα ονόματα ιδιοτήτων που είναι έγκυρα στο συγκεκριμένο περιεχόμενο.

4.4.9 Επίλογος

Το OWL-S Editor χρησιμοποιείται από χρήστες που χρειάζονται ένα γρήγορο τρόπο για να δημιουργήσουν τις περιγραφές των Σημασιολογικών Υπηρεσιών Δικτύου, ενώ παράλληλα χρησιμοποιούν προδιαγραφές όπως η WSDL και τα UML Activity Diagrams. Ο κύριος στόχος πίσω από αυτό το εργαλείο είναι η ανάκτηση των

βασικών λεπτομερειών της OWL-S δομής, που θα παρέχουν ένα εύκολο, προσβάσιμο τρόπο στη δημιουργία σύνθετων περιγραφών. [49]

5 Βιβλιογραφία

1. **XML.com** <http://www.xml.com/pub/a/ws/2001/04/04/webservices/index.html>, Venu Vasudevan.
2. Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by Its Inventor. Tim Berners-Lee, with Mark Fischetti. Harper San Francisco, 1999, www.ScientificAmerican.com.
3. D. Fensel, C. Bussler: The Web Service Modeling Framework WSMF, Electronic Commerce Research and Applications, 1(2), 2002.
4. The DAML services coalition: DAML-S: Semantic Markup for Web Services (version 0.9), available at <http://www.daml.org/services/daml-s/0.9/daml-s.pdf>, 2003.
5. T. Andrews, F. Curbera, H. Dholakia, Y. Goland, J. Klein, F. Leymann, K. Liu, D. Roller, D. Smith, S. Thatte, I. Trickovic, S. Weerawarana. Business Process Execution Language for Web Services, version 1.1, available at <ftp://www6.software.ibm.com/software/developer/library/ws-bpel11.pdf>, 2003.
6. Business Process Modeling Language (BPML). Accessed June 2003 from www.bpml.org.
7. W3C. Web Service Choreography Interface (WSCI) 1.0. Accessed June 2003 from www.w3.org/TR/wsci
8. Web Service Composer Project, Maryland Information and Network Dynamics Lab, University of Maryland, USA; <http://www.mindswap.org/~evren/composer/>
9. M. Paloucci, T. Kawamura, T. R. Payne, K. Sycara: Semantic Matching of Web Services Capabilities. In Int. Semantic Web Conference, Sardinia, Italy, pages 333-347, June 2002.
10. B. Benatallah, M. Hacid, C. Rey, F. Toumani: Semantic reasoning for Web Services Discovery, WWW 2003 workshop on E-services and the Semantic Web (ESSW'03), Budapest, Hungary, 2003.
11. W.M.P. van der Aalst, M. Dumas, A.H.M. ter Hofstede, P. Wohed. Pattern based analysis of BPML (and WSCI), QUT Technical report, FIT-TR-2002-05, Queensland University of Technology, Brisbane, 2002.

12. P. Wohed, W.M.P. van der Aalst, M. Dumas, A.H.M. ter Hofstede. Pattern based analysis of BPEL4WS. QUT Technical report, FIT-TR-2002-04, Queensland University of Technology, Brisbane, 2002.
13. Rubén Lara, Holger Lausen, Sinuhé Arroyo, Jos de Bruijn, Dieter Fensel, Universität Innsbruck, <http://deri.semanticweb.org/>
14. Erik Christensen, Francisco Curbera, Greg Meredith, Sanjiva Weerawarana. Web Services Description Language (WSDL) 1.1, 2001. At <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>
15. The Universal Description, Discovery and Integration (UDDI) protocol. Version 3, 2003. At <http://www.uddi.org/>
16. David Martin, Mark Burstein, Ora Lassila, Massimo Paolucci, Terry Payne, Sheila McIlraith. Describing Web Services using OWL-S and WSDL. October 2003. At <http://www.daml.org/services/owl-s/1.0/owl-s-wsdl.html>
17. The Rule Markup Initiative. At <http://www.dfki.uni-l.de/ruleml/>.
18. D. McDermott and D. Dou. Representing Disjunction and Quantifiers in RDF. Proceedings of the First International Semantic Web Conference (ISWC2002), 2002.
19. Ian Horrocks, Peter F. Patel-Schneider, Harold Boley, Said Tabet. OWL Rules Language, Draft version. Technical report, 29 October 2003
20. M. Paolucci, N. Srinivasan, K. Sycara, and T. Nishimura. Toward a Semantic Choreography of Web services: from WSDL to DAML-S. In Proceedings of ICWS03, 2003.
21. D McDermott. The Planning Domain Definition Language Manual. Yale Computer Science Report 1165 (CVC Report 980003), 1998.
22. D. McDermott. Estimated-Regression Planning for Interaction with Web Services. In Proceedings of the Sixth International Conference on AI Planning and Scheduling, pp. 204—211, 2002.
23. R. Fikes and N. J. Nilsson. STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving. Artificial Intelligence 2, pp. 189-208, 1971.
24. S. McIlraith and R. Fadel. Planning with Complex Actions. In Proceedings of the Ninth International Workshop on Non-Monotonic Reasoning (NMR2002), pages 356-364, April, 2002.

25. S. McIlraith and T. Son. Adapting Golog for Composition of Semantic Web Services. In Proceedings of the Eighth International Conference on Knowledge Representation and Reasoning (KR2002), pp. 482-493, 2002.
26. Ryusuke Masuoka, Yannis Labrou, Bijan Parsia, and Evren Sirin. Ontology-Enabled Pervasive Computing Applications. In IEEE Intelligent Systems, 18(10):68-72, 2003.
27. David Martin¹, Massimo Paolucci², Sheila McIlraith³, Mark Burstein, Drew McDermott, Deborah McGuinness, Bijan Parsia, Terry Payne, Marta Sabou, Monika Solanki, Naveen Srinivasan, Katia Sycara
28. D. L. McGuinness and F. van Harmelen. OWL Web Ontology Language Overview. <http://www.w3.org/TR/owl-features/>, February 2004. World Wide Web Consortium (W3C) Recommendation
29. J. Hendler and D. L. McGuinness. DARPA Agent Markup Language. IEEE Intelligent Systems, 15(6):72-73, 2001.
30. C. Schlenoff, M. Gruninger, F. Tissot, J. Valois, J. Lubell, and J. Lee. The Process Specification Language (PSL): Overview and Version 1.0 Specification. NISTIR 6459, National Institute of Standards and Technology, Gaithersburg, MD, 2000.
31. S. McIlraith, T. C. Son, and H. Zeng. Semantic Web Service. IEEE Intelligent Systems, 16(2):46-53, 2001.
32. I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosz, and M. Dean. Swrl: A semantic web rule language combining owl and ruleml, 2003. Available at <http://www.daml.org/2003/11/swrl/>.
33. KIF. Knowledge Interchange Format: Draft proposed American National Standard (dpans). Technical Report 2/98-004, ANS, 1998. Also at <http://logic.stanford.edu/kif/dpans.html>.
34. David Martin, SRI International (editor), Mark Burstein, BBN Technologies, Jerry Hobbs, USC Information Sciences Institute, Ora Lassila, Nokia, Drew McDermott, Yale University, Sheila McIlraith, University of Toronto, Srin Narayanan, International Institute of Computer Science, Massimo Paolucci, Carnegie Mellon University, Bijan Parsia, The MIND Laboratory of the University of Maryland at College Park, Terry Payne, University of Southampton, Evren Sirin, The MIND Laboratory of the University of Maryland at College Park, Naveen Srinivasan, Carnegie Mellon University, Katia Sycara, Carnegie Mellon University

35. David Martin, Mark Burstein, Ora Lassila, Massimo Paolucci, Terry Payne, Sheila McIlraith
36. <http://www.daml.org/services/owl-s/1.1/>
37. The ebXML website. <http://www.ebxml.org/>
38. The Open Grid Services Infrastructure (OGSI) 1.0, July 2003. <https://forge.gridforum.org/projects/ogsi-wg>
39. Anupriya Ankolekar, Carnegie Mellon University, David Martin, SRI International, Deborah McGuinness, Stanford University, Sheila McIlraith, University of Toronto, Massimo Paolucci, Carnegie-Mellon University, Bijan Parsia, The MIND Laboratory of the University of Maryland at College Park
40. L. Kagal, T. Finin, and A. Joshi, A Policy Based, Approach to Security on the Semantic Web, Second Int. Semantic Web Conference, Sanibel Island FL, October 2003.
41. J.M. Bradshaw, A. Uszok, R. Jeffers, N. Suri, P.Hayes, M. Burstein, A. Acquisiti, B. Benyo, M.R. Breedy, M. Carvalho, D. Diller, M Johnson, S. Kulkarni, J. Lott, M.Sierhuis, R. Van Hoof. Representation and Reasoning for DAML-Based Policy and Domain Services in KAoS and Nomads, AAMAS, 2003.
42. F. Gandon and N. Sadeh. Semantic Web Technologies to Reconcile Privacy and Context Awareness. Web Semantics Journal, Vol 1, No. 3, 2004.
43. G. Denker, L. Kagal, T. Finin, M. Paolucci, K.Sycara. Security for DAML Web Services: Annotation and Matchmaking, Second Int. Semantic Web Conference (ISWC2003), Sanibel Island FL, October 2003.
44. Lalana Kagal, Massimo Paolucci, Naveen Srinivasan, Grit Denker, Tim Finin, Katia Sycara
45. North American Industry Classification System, <http://www.census.gov/epcd/www/naics.html>
46. Mithun Sheshagiri, Norman M. Sadeh, and Fabien Gandon Mobile Commerce Laboratory School of Computer Science Carnegie Mellon University, Pittsburgh, PA 15213-3891
47. Fikes, R., Hayes, P., and Horrocks, I. "OWL-QL – A Language for Deductive Query Answering on the Semantic Web." Technical Report Knowledge Systems Laboratory, Stanford University, Stanford, CA, KSL-03-14, 2003.
48. Massimo Paolucci, Julien Soudry, Naveen Srinivasan and Katia Sycara, The Robotics Institute, Carnegie Mellon University

49. Mr. James Scicluna jsci023@um.edu.mt, Mr. Charlie Abela
charlie.abela@um.edu.mt, Dr. Matthew Montebello
matthew.montebello@um.edu.mt, Department of Computer Science and Artificial
Intelligence, University of Malta, Msida MSD 06, Malta (Europe)