

You may contact the author at: tt2@ecs.soton.ac.uk

Σημασιολογικές Υπηρεσίες Ιστού

Θανάσης Τυροπάνης

Σε αυτή την εργασία μετά από μία παρουσίαση των Υπηρεσιών Ιστού (Web Services) και των τεχνολογιών με τις οποίες υποστηρίζεται η ανάπτυξή τους παρουσιάζουμε τις Σημασιολογικές Υπηρεσίες Ιστού. Οι Σημασιολογικές Υπηρεσίες Ιστού είναι το αποτέλεσμα της εφαρμογής τεχνολογιών του Σημασιολογικού Ιστού (Semantic Web) σε Υπηρεσίες Ιστού. Αναλύουμε τα πλεονεκτήματα που παρουσιάζουν οι Σημασιολογικές Υπηρεσίες Ιστού όπως και τα θέματα που είναι ακόμα ανοικτά προς επίλυση. Παρουσιάζουμε επίσης τις μελλοντικές κατευθύνσεις όσον αφορά την εξέλιξη των Σημασιολογικών Υπηρεσιών Ιστού.

Λέξεις-κλειδιά: Υπηρεσίες Ιστού, Σημασιολογικές Υπηρεσίες Ιστού, SOAP, WSDL, UDDI, WS Composition, WS Choreography, OWL-S, WSMO, WSDL-S

1 Υπηρεσίες Ιστού και Αρχιτεκτονικές Υπηρεσιών

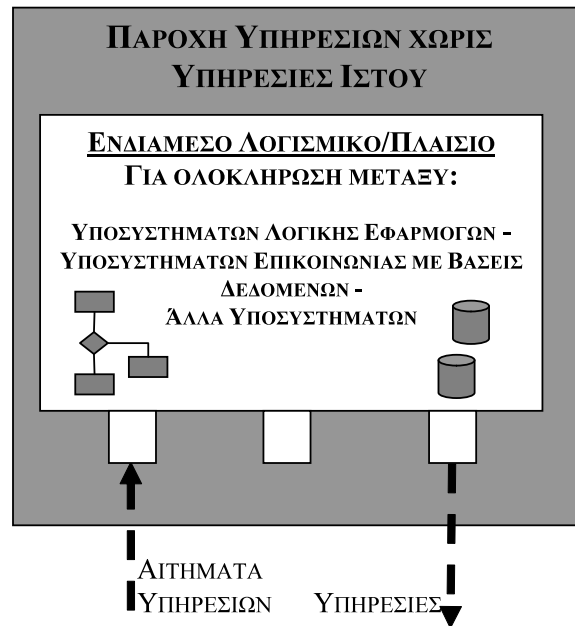
Τα τελευταία χρόνια παρουσιάζεται έντονο ενδιαφέρον και δραστηριότητα στην ανάπτυξη *Αρχιτεκτονικών Υπηρεσιών* (SOA - Service Oriented Architecture) για την παροχή υπηρεσιών κυρίως μέσω του Διαδικτύου. Οι Αρχιτεκτονικές Υπηρεσιών υπόσχονται υπηρεσίες οι οποίες θα προσαρμόζονται πιο εύκολα στις ανάγκες του κάθε χρήστη-πελάτη και έτσι θα είναι εξατομικευμένες. Ταυτόχρονα όμως θα προσφέρουν ευελιξία και στους φορείς παροχής υπηρεσιών αφού ο κύκλος της ανάπτυξης και προσφοράς νέων υπηρεσιών αναμένεται να είναι σημαντικά πιο σύντομος και οικονομικός με τη χρήση τους.

1.1 Αρχιτεκτονική Υπηρεσιών (Service Oriented Architecture)

Η κύρια ιδέα στην οποία βασίζονται οι Αρχιτεκτονικές Υπηρεσιών εντοπίζεται κυρίως στον καταμερισμό των χαρακτηριστικών που προσφέρει μία υπηρεσία σε μικρότερες υπηρεσίες οι οποίες θα μπορούν να είναι αυτόνομες αλλά και να συνεργάζονται μεταξύ τους. Τέτοιες υπηρεσίες είναι οι *Υπηρεσίες Ιστού* (Web Services). Οι Υπηρεσίες Ιστού είναι σε θέση να παρέχουν απλές υπηρεσίες αυτόνομα. Επίσης όμως, είναι σε θέση να συνεργάζονται με άλλες Υπηρεσίες Ιστού προκειμένου να δημιουργήσουν πιο σύνθετες υπηρεσίες. Με τη χρήση μίας Αρχιτεκτονικής Υπηρεσιών, ένας *Φορέας Παροχής Υπηρεσιών* (ΦΠΥ) θα έχει τη δυνατότητα να παρέχει αντί για μία μονολιθική υπηρεσία βασισμένη παραδοσιακές τεχνολογίες, όπως στο Σχήμα 1, ένα σύνολο απλούστερων Υπηρεσιών Ιστού.

Η λογική της χρήσης απλούστερων υπηρεσιών ιστού είναι ότι θα μπορεί ο ΦΠΥ να τις συνδυάζει ευέλικτα και δυναμικά και έτσι να διαμορφώνει νέες *Σύνθετες Υπηρεσίες Ιστού* (Composite Web Services) όπως απεικονίζονται στο Σχήμα 2.

Με τη χρήση Υπηρεσιών Ιστού ο ΦΠΥ αποκτά τη δυνατότητα προσφοράς διαφορετικών εκδοχών Σύνθετων Υπηρεσιών Ιστού και είναι δυνατό να καλυφθούν καλύτερα οι ανάγκες ενός μεγαλύτερου αριθμού χρηστών-πελατών από ότι με την παροχή μίας μονολιθικής υπηρεσίας.



Σχήμα 1 Παροχή Υπηρεσιών από ΦΠΥ χωρίς Αρχιτεκτονική Υπηρεσιών

Παράδειγμα Αρχιτεκτονικής Υπηρεσιών

Για παράδειγμα, ας υποθέσουμε ότι ένας πάροχος υπηρεσιών σχετικών με χρηματιστηριακές συναλλαγές προσφέρει υπηρεσίες στους πελάτες του μέσω του παγκοσμίου ιστού. Οι επισκέπτες των ιστοσελίδων του παρόχου μπορούν να ανατρέχουν στην υπηρεσία και:

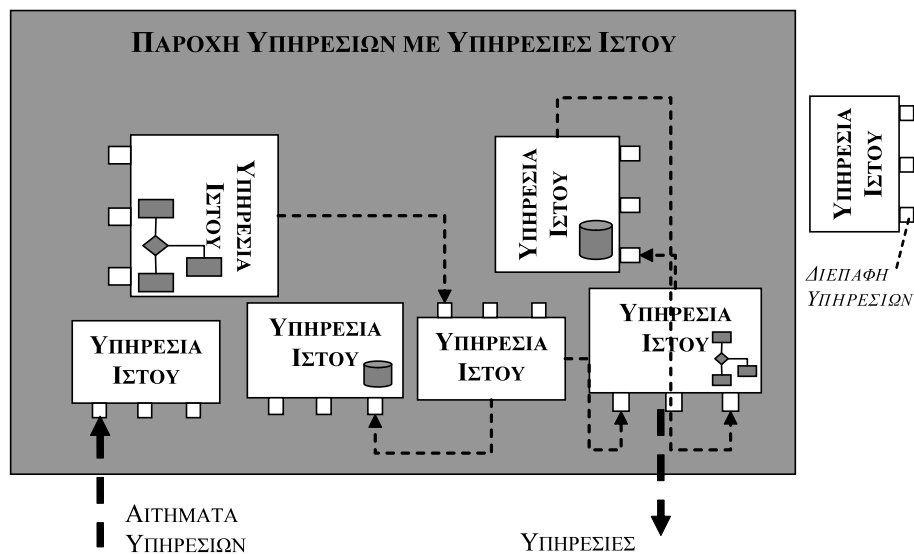
- (α) να εγγραφούν ως απλοί χρήστες ή πελάτες,
- (β) να ζητούν πληροφορίες για τις τιμές των μετοχών,
- (γ) να αγοράζουν μετοχές που είναι διαθέσιμες στο χρηματιστήριο με το οποίο συνεργάζεται ο πάροχος,
- (δ) να παρακολουθούν την εξέλιξη του χαρτοφυλακίου μετοχών τους, ενώ
- (ε) ορισμένοι χρήστες μπορούν προαιρετικά να κάνουν χρήση ενός έξυπνου διαμεσολαβητή ο οποίος με βάση κάποια κριτήρια και προτιμήσεις του χρήστη αλλά και την κατάσταση της αγοράς μπορεί να συνιστά αγορά ή πώληση μετοχών.

Μετά από κάποιους μήνες ο πάροχος βλέπει ότι ο έξυπνος διαμεσολαβητής έχει μεγάλη επιτυχία και ότι με το να τον προσφέρει ως υπηρεσία σε χρήστες που δεν

είναι ακόμα πελάτες του μπορεί να αυξήσει την πελατειακή του βάση. Για αυτό το λόγο αποφασίζει να επιτρέψει τη χρήση του έξυπνου διαμεσολαβητή και από άλλους χρήστες που θα εγγραφούν στο σύστημά του χωρίς να είναι πελάτες του. Έτσι, όταν θα προτείνεται η αγορά ή πώληση μετοχών οι χρήστες που δεν είναι πελάτες δε θα μπορεί να την εκτελέσουν τη συναλλαγή άμεσα με τον συγκεκριμένο πάροχο. Αυτό θα μπορούσε όμως να είναι ένα κίνητρο να εγγραφούν και ως πελάτες το παρόχου χρηματιστηριακών συναλλαγών.

Εάν ο πάροχος αυτού του παραδείγματος βασίζεται σε αρχιτεκτονικές που δεν είναι αρχιτεκτονικές υπηρεσιών (όπως οι αρχιτεκτονικές με χρήση υπηρεσιών ιστού) θα πρέπει να ενημερώσει την λογική εφαρμογής (application logic) της υπηρεσίας του και μετά από δοκιμές να την προσφέρει στους χρήστες. Αυτά θα πρέπει να γίνουν χωρίς να προκαλέσουν ενόχληση στους υπάρχοντες χρήστες της υπηρεσίας το οποίο θέτει επιπλέον δυσκολία στο εγχείρημα.

Ο πάροχος του παραδείγματος θα μπορούσε όμως να χρησιμοποιεί μια αρχιτεκτονική υπηρεσιών όπου οι λειτουργίες (α), (β), (γ) και (δ) θα μπορούσαν να προσφέρονται από αυτόνομες αλλά συνδυάσιμες υπηρεσίες ιστού. Με αυτό τον τρόπο, μια υπηρεσία που προσφέρει τη λειτουργία (ε) θα μπορούσε να εισαχθεί στην αρχιτεκτονική και να συνδυαστεί με τις υπόλοιπες λειτουργίες προκειμένου να προσφερθεί στους πελάτες. Όταν ο πάροχος αποφασίσει να προσφέρει την υπηρεσία (ε) από μόνη της για εγγεγραμμένους χρήστες μπορεί να συνδυάσει τις λειτουργίες των υπηρεσιών (α) και (ε) και να προσφέρει τη νέα υπηρεσία ευέλικτα, δυναμικά και χωρίς να προκαλέσει προβλήματα στους άλλους πελάτες/χρήστες του.



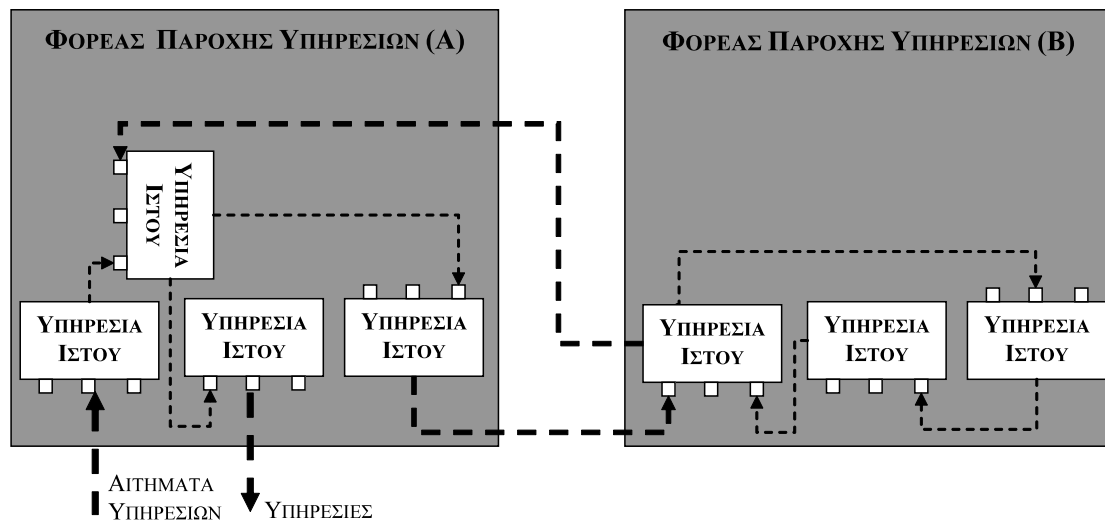
Σχήμα 2 Παροχή Υπηρεσιών από ΦΠΥ χωρίς Αρχιτεκτονική Υπηρεσιών

Η παροχή Σύνθετων Υπηρεσιών Ιστού μπορεί να βασιστεί στη σύνθεση Υπηρεσιών Ιστού που παρέχονται από τον ίδιο ή από διαφορετικούς παρόχους όπως παρουσιάζεται στο Σχήμα 3.

Φορείς Παροχής Υπηρεσιών και Αρχιτεκτονικές Υπηρεσιών

Η δυνατότητα συνδυασμού υπηρεσιών ιστού από διαφορετικούς παρόχους ευνοεί την παροχή πιο πλούσιων υπηρεσιών όπως επίσης και η εξειδίκευση παρόχων στην

ανάπτυξη συγκεκριμένου τύπου υπηρεσιών. Στο παράδειγμα με τον πάροχο χρηματιστηριακών υπηρεσιών που αναφέρεται στις προηγούμενες παραγράφους, θα ήταν δυνατή η συνεργασία του συγκεκριμένου παρόχου με άλλα χρηματιστήρια μέσω χρήσης των αντίστοιχων υπηρεσιών ιστού που αυτά μπορεί να διαθέτουν. Έτσι, η συνεργασία με επιπλέον χρηματιστήρια θα ήταν πιο ευέλικτη και θα βοηθούσε τον πάροχο του παραδείγματος από την μια μεριά να προσφέρει πιο πλούσιες υπηρεσίες στους πελάτες τους αφού θα μπορούν να συναλλάσσονται με περισσότερα χρηματιστήρια και από την άλλη μεριά πιθανόν να εστιάσει στην ανάπτυξη χρηματιστηριακών υπηρεσιών βασισμένων σε έξυπνους διαμεσολαβητές και να παραχωρήσει κάποιες από τις άλλες υπηρεσίες που προσφέρει σε άλλους παρόχους-συνεργάτες.



Σχήμα 3 Παροχή Υπηρεσιών συνδυάζοντας Υπηρεσίες από διαφορετικούς παρόχους

Οι Αρχιτεκτονικές Υπηρεσιών ασχολούνται με μια σειρά από άλλα θέματα τα οποία έχουν να κάνουν με Ασφάλεια (Security), Συναλλαγές (Transactions), Ανακάλυψη Υπηρεσιών (Service Discovery), Σύνθεση Υπηρεσιών (Composition), Ενορχήστρωση Υπηρεσιών (Orchestration), Χορογραφία Υπηρεσιών (Choreography), Διαπραγμάτευση για παροχή υπηρεσιών (Negotiation), κλπ.

1.2 Υπηρεσίες Ιστού (Web Services)

Η χρήση Αρχιτεκτονικών Υπηρεσιών στους ΦΠΥ για παροχή διαφορετικών εκδοχών σύνθετων υπηρεσιών βασισμένων στη σύνθεση Υπηρεσιών Ιστού (Web Service) παρουσιάζει σημαντικά πλεονεκτήματα για ΦΠΥ και πελάτες.

Αυτά τα πλεονεκτήματα εντοπίζονται στο ότι οι Υπηρεσίες Ιστού ως δομικά στοιχεία των Αρχιτεκτονικών Υπηρεσιών αποτελούν υπηρεσίες Αυτόνομες και Συνδυάσιμες όπως περιγράφεται στις προηγούμενες παραγράφους. Επιπλέον, οι Υπηρεσίες Ιστού είναι συστήματα Χαλαρής Σύζευξης (loosely coupled) που επικοινωνούν μεταξύ τους μέσω Ανταλλαγής Μηνυμάτων.

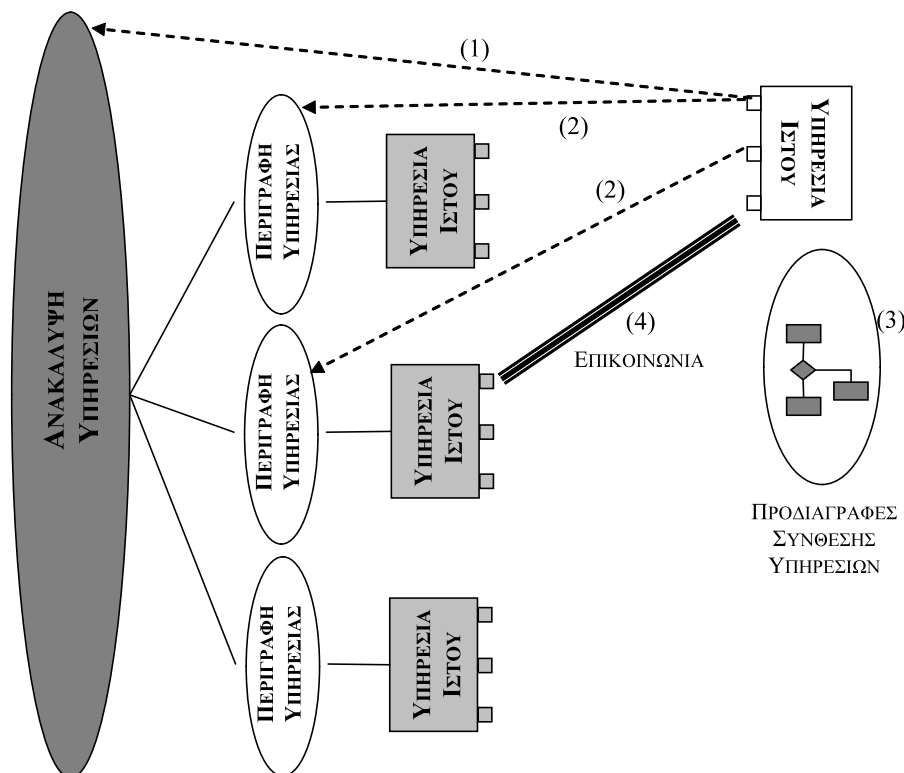
Ως συστήματα Χαλαρής Σύζευξης αναφέρονται αυτά για τα οποία δεν απαιτείται γνώση λεπτομερειών σχετικά με την πλατφόρμα στην οποία τρέχουν ή του τρόπου/τεχνολογιών υλοποίησή τους προκειμένου να γίνει επικοινωνία μαζί τους. Έτσι είναι δυνατή η χρήση τέτοιων συστημάτων σε πιο ευρεία κλίμακα η οποία μπορεί να γίνεται δυναμικά. Είναι με αυτό τον τρόπο δυνατή μια ευέλικτη και δυναμική συνεργασία τέτοιων συστημάτων.

Στον αντίποδα των συστημάτων χαλαρής σύζευξης βρίσκονται τα συστήματα *Ισχυρής Σύζευξης* (tightly coupled) όπου απαιτείται μεγαλύτερη γνώση λεπτομερειών σχετικά με την υλοποίηση ενός συστήματος προκειμένου να χρησιμοποιηθεί. Για παράδειγμα, τεχνολογίες ενδιάμεσου λογισμικού (middleware) όπως η CORBA ή η COM προϋποθέτουν, προκειμένου να γίνει επικοινωνία με κάποιο σύστημα, την ολοκλήρωση με κάποιο αντικείμενο που προσφέρει το σύστημα, ακόμα και αν απαιτείται η χρήση μόνο μίας μεθόδου αυτού του αντικειμένου. Εάν αλλάξει ο ορισμός της κλάσης αυτού του αντικειμένου η εφαρμογή που χρησιμοποιεί το υλοποιημένο αντικείμενο πρέπει να αλλάξει επίσης. Για αυτό το λόγο αυτά τα συστήματα ονομάζονται ισχυρής σύζευξης, λόγω της ισχυρής εξάρτησης που παρουσιάζουν το ένα με το άλλο.

Επίσης, η επικοινωνία μεταξύ διαφορετικών υλοποιήσεων συστημάτων ενδιάμεσου λογισμικού (όπως π.χ. μεταξύ CORBA και COM) παρουσιάζει έντονες δυσκολίες. Σύμφωνα με πολλούς οι αρχιτεκτονικές υπηρεσιών υπάρχουν για πάνω από 20 χρόνια ([Newcomer, Lomow, 2005](#)) εκείνο όμως που κάνει τη διαφορά σήμερα είναι οι τεχνολογίες που χρησιμοποιούνται και οι οποίες προσφέρουν διαλειτουργικότητα μεταξύ παρόχων και χαλαρή σύζευξη.

Ένα άλλο πλεονέκτημα των Υπηρεσιών Ιστού εντοπίζεται στη χρήση ανταλλαγής μηνυμάτων για επικοινωνία μεταξύ τους. Αυτό τονίζει την χαλαρή σύζευξη φύση τους αφού επιτρέπει, ακόμα και εάν γίνουν αλλαγές στο σχήμα των ανταλλασσόμενων μηνυμάτων σε ένα σύστημα να είναι και πάλι δυνατή η επικοινωνία του με ένα άλλο σύστημα υλοποιώντας σχετικά εύκολα ένα τρόπο μετατροπής σχήμα μηνυμάτων ([Weerawarana et al, 2005](#)). Επίσης η χρήση ανταλλαγής μηνυμάτων επιτρέπει μια πληθώρα μοντέλων αλληλεπίδρασης μεταξύ συστημάτων (π.χ. μονόδρομα, αμφίδρομα, ειδοποίηση, κλπ.).

Γίνεται λοιπόν αντιληπτό ότι ένα μεγάλο μέρος των ευκολιών που παρέχουν οι Αρχιτεκτονικές Υπηρεσιών όπως αναλύθηκε πιο πάνω οφείλεται ακριβώς στη Χαλαρή Σύζευξη φύση των Υπηρεσιών στις οποίες βασίζονται και στην ανταλλαγή μηνυμάτων μεταξύ τους.



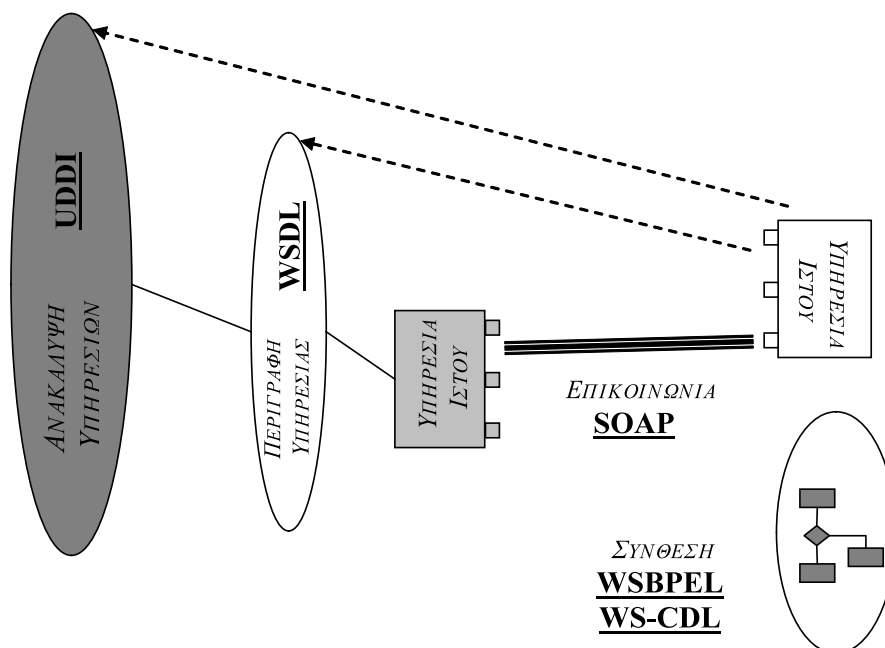
Σχήμα 4 Λειτουργίες Αρχιτεκτονικών Υπηρεσιών με Υπηρεσίες Ιστού

Δεδομένων των πλεονεκτημάτων που προσφέρουν οι Υπηρεσίες Ιστού και της εμβέλειας του εγχειρήματος της μετάβασης σε Αρχιτεκτονικές Υπηρεσιών για την ανάπτυξη και αξιοποίησή τους, σημαντικός είναι ο ρόλος των προτύπων που καθορίζουν τον τρόπο με τον οποίο πρέπει να γίνεται η περιγραφή, η ανακάλυψη, η σύνθεση και η χρησιμοποίηση Υπηρεσιών Ιστού για την παροχή Σύνθετων Υπηρεσιών Ιστού από έναν ή περισσότερους παρόχους από κοινού.

Η βιομηχανία σε συνεργασία με την ερευνητική κοινότητα εργάζονται σε βέλτιστες πρακτικές για την ανάπτυξη και υποστήριξη αρχιτεκτονικών υπηρεσιών ιστού ([Krafzig et al, 2005](#)) αλλά και σε πρότυπα τα οποία υποστηρίζουν τις ακόλουθες λειτουργίες όπως απεικονίζονται και στο Σχήμα 4:

- Επικοινωνία μεταξύ Υπηρεσιών Ιστού
- Περιγραφή Υπηρεσιών Ιστού
- Ανακάλυψη Υπηρεσιών Ιστού
- Σύνθεση Υπηρεσιών Ιστού

Η συμπληρωματικότητα των παραπάνω λειτουργιών όπως απεικονίζεται και στο Σχήμα 5 για την παροχή Σύνθετων Υπηρεσιών Ιστού χρησιμοποιώντας Αρχιτεκτονικές Υπηρεσιών προϋποθέτει ότι η διαδικασία δημιουργίας προτύπων απαιτεί τη συνεργασία μεταξύ ενός μεγάλου αριθμού ενδιαφερόμενων φορέων. Έτσι, ιδιαίτερα σημαντικός είναι ο ρόλος του W3C (World Wide Web Consortium) και του οργανισμού OASIS Open (Organization for the Advancement of Structured Information Standards).



Σχήμα 5 Βασικά Πρότυπα για Παροχή Υπηρεσιών Ιστού

Τα πρότυπα που επικρατούν για κάθε λειτουργία όπως παρουσιάζονται στο Σχήμα 5 είναι τα εξής:

- Πρωτόκολλο SOAP για την επικοινωνία μεταξύ Υπηρεσιών Ιστού που έχει αναπτυχθεί από το W3C.
- Γλώσσα WSDL (Web Service Description Language) για την περιγραφή Υπηρεσιών Ιστού που αναπτύσσεται από το W3C.
- Πρωτόκολλο UDDI για την ανακάλυψη Υπηρεσιών Ιστού που αναπτύσσεται από τον OASIS Open.
- Γλώσσες WSBEEL και WS-CDL για την σύνθεση υπηρεσιών ιστού.

Στα πλαίσια του οργανισμού W3C υπάρχει ομάδα εργασίας σε θέματα αρχιτεκτονικής υπηρεσιών ιστού (<http://www.w3.org/TR/ws-arch/>) όπου περιγράφεται το όλο πλαίσιο των υπηρεσιών ιστού από την οπτική γωνία της αρχιτεκτονικής.

1.3 Το πρωτόκολλο επικοινωνίας SOAP

Το πρωτόκολλο SOAP (το οποίο ξεκίνησε ως ακρώνυμο: Simple Object Access Protocol) παρέχει ένα πλαίσιο επικοινωνίας μεταξύ κόμβων (SOAP nodes) στον παγκόσμιο ιστό μέσω ανταλλαγής μηνυμάτων σε XML. Τέτοιοι κόμβοι μπορούν να είναι οι Υπηρεσίες Ιστού οι οποίες μπορούν να επικοινωνούν μέσω του πρωτοκόλλου SOAP. Ένα μήνυμα που αποστέλλεται από έναν Αρχικό Αποστολέα κόμβο (initial sender) σε έναν Τελικό Παραλήπτη κόμβο (ultimate receiver) μπορεί να παραληφθεί, επεξεργαστεί και προωθηθεί από έναν αριθμό Ενδιάμεσων κόμβων (intermediaries).

Το SOAP (<http://www.w3.org/TR/soap/>) είναι αποτέλεσμα τυποποίησης από τον οργανισμό W3C (World Wide Web Consortium <http://www.w3.org>). Η πιο πρόσφατη έκδοση είναι αυτή που εξέδωσε ο W3C το 2003, SOAP 1.2.

Το SOAP αναφέρεται κυρίως ως ένα αμνήμων (stateless) πρωτόκολλο που επιτρέπει την ανταλλαγή μηνυμάτων μονόδρομα (one-way) αλλά η υποστήριξη επιπλέον μοντέλων ανταλλαγής μηνυμάτων όπως αμφίδρομη επικοινωνία (request/response) μπορεί να υποστηριχθεί (α) σε επίπεδο εφαρμογής που κάνει χρήση του SOAP ή (β) σε επίπεδο πρωτοκόλλου που υλοποιεί τη μεταφορά των μηνυμάτων SOAP (transport).

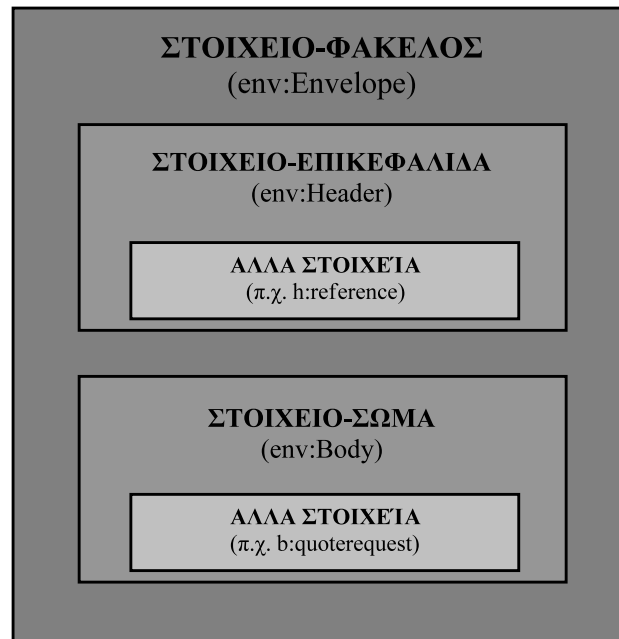
Τα μηνύματα που ανταλλάσσονται στα πλαίσια του SOAP προβλέπεται να έχουν συγκεκριμένη δομή (message structure) σύμφωνα με το πρότυπο όπως επίσης προβλέπεται ένα διεργασιακό μοντέλο (processing model) δηλ. ένας συγκεκριμένος τρόπος επεξεργασίας μηνυμάτων από τους κόμβους που εμπλέκονται στην ανταλλαγή ή τη μετάδοσή τους. Επίσης προβλέπεται η μεταφορά αυτών των μηνυμάτων χρησιμοποιώντας πρωτόκολλα όπως το HTTP ή το SMTP αν και προς το παρόν έχουν τυποποιηθεί οι προδιαγραφές για μεταφορά μέσω HTTP μόνο.

Δομή Μηνυμάτων (message structure)

Ουσιαστικά ένα μήνυμα SOAP είναι ένα XML κείμενο. Σύμφωνα με το πρότυπο τα μηνύματα SOAP έχουν ως στοιχείο-ρίζα (root element) το στοιχείο-φάκελος του μηνύματος που ονομάζεται *env:Envelope*, όπου το 'env' αντιστοιχεί στον ονοματοχώρο (χώρος ονομάτων, namespace) <http://www.w3.org/2003/05/soap-envelope>.

Ο φάκελος του μηνύματος με τη σειρά του περιέχει δύο στοιχεία: ένα προαιρετικό στοιχείο-επικεφαλίδα που ονομάζεται *env:Header* και ένα στοιχείο-σώμα που ονομάζεται *env:Body*. Από αυτό το επίπεδο και κάτω, τόσο το στοιχείο-επικεφαλίδα

όσο και το στοιχείο-σώμα μπορούν να περιέχουν άλλα στοιχεία που μπορεί να ορίσει ο χρήστης.



Σχήμα 6 Η δομή ενός μηνύματος SOAP

Τυπικά, το στοιχείο-επικεφαλίδα περιέχει πληροφορίες σχετικά με το πως μπορεί να επεξεργαστεί κάποιο μήνυμα ο κόμβος που το λαμβάνει και δεν αποτελούν μέρος της μεταφερόμενης πληροφορίας. Η κυρίως πληροφορία ή περιεχόμενο που μεταφέρει ένα SOAP μήνυμα βρίσκεται στο στοιχείο-σώμα του μηνύματος όπως φαίνεται και στο Σχήμα 6.

Ας δούμε τώρα και ένα παράδειγμα ενός SOAP μηνύματος:

```

<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <env:Header>
    <h:reference xmlns:h="http://stockinfo.example.com/reference"
      env:role="http://www.w3.org/2003/05/soap-envelope/role/next"
      env:mustUnderstand="true">
      <h:number>123456</h:number>
    </h:reference>
  </env:Header>
  <env:Body>
    <b:quoterequest xmlns:b="http://stockinfo.example.com/stockupdate">
      <b:stockexchange> WXYZ </b:stockexchange>
      <b:stock> ABCD </b:stock>
    </b:quoterequest>
  </env:Body>
</env:Envelope>
  
```

Αυτό το μήνυμα αποστέλλεται σε μία υπηρεσία ιστού την *stockinfo.example.com* η οποία για ένα συγκεκριμένο κωδικό χρηματιστηρίου (*b:stockexchange*) και ένα κωδικό μετοχής (*b:stock*) επιστρέφει την τρέχουσα αξία της μετοχής. Οι παράμετροι που απαιτούνται για αυτή τη λειτουργία δίδονται στο στοιχείο-σώμα του μηνύματος (*env:Body*). Το στοιχείο-επικεφαλίδα (*env:Header*) επίσης μεταφέρει κάποια πληροφορία η οποία όμως δε σχετίζεται άμεσα με τις παραμέτρους που απαιτούνται για την τελευταία τιμή της μετοχής. Η πληροφορία στο στοιχείο-επικεφαλίδα είναι ένα συγκεκριμένο νούμερο αναφοράς (reference number) το οποίο πιστοποιεί τον πελάτη ή χρήστη που καλεί την υπηρεσία ιστού και επιτρέπει στην υπηρεσία με βάση αυτό το νούμερο να δώσει την τιμή της μετοχής μέσα στο χρονικό διάστημα που έχει ορίσει ο πελάτης (π.χ. μέσα στα επόμενα πέντε λεπτά ή δέκα λεπτά, κλπ.).

Παρατηρούμε επίσης ότι στο παράδειγμα αυτό και συγκεκριμένα στο στοιχείο *h:reference* που βρίσκεται στο στοιχείο-επικεφαλίδα υπάρχουν δύο ιδιότητες (attributes): η ιδιότητα *env:role* και η ιδιότητα *env:mustUnderstand*. Αυτές εξηγούνται στις επόμενες παραγράφους.

Διεργασιακό Μοντέλο (processing model)

Ο W3C έχει τυποποιήσει συγκεκριμένους SOAP ρόλους που μπορεί να παίξει κάποιος κόμβος όταν λάβει ένα μήνυμα SOAP. Ένας τέτοιος «κόμβος» μπορεί να είναι μία υπηρεσία ιστού.

Ένας κόμβος SOAP μπορεί να είναι ο *Αρχικός Αποστολέας* (initial sender), ο *Τελικός Παραλήπτης* (ultimate receiver) ή ένας *Ενδιάμεσος* (intermediary) προκειμένου να γίνει η αποστολή και πιθανή επεξεργασία ενός μηνύματος SOAP που εστάλη από τον αρχικό αποστολέα μέχρι να φθάσει στον τελικό παραλήπτη.

Τα στοιχεία που βρίσκονται στο στοιχείο-επικεφαλίδα μπορούν να χρησιμοποιήσουν μία από τις τρεις τυποποιημένες τιμές που μπορεί να έχει η τυποποιημένη ιδιότητα *env:role* προκειμένου να υποδείξουν στους Ενδιάμεσους και στον Τελικό Παραλήπτη αν και πως επιτρέπεται να επεξεργαστούν το συγκεκριμένο στοιχείο. Έτσι, μπορεί να ζητηθεί η επεξεργασία του στοιχείου αυτού μόνο από τον τελικό παραλήπτη, από τους ενδιάμεσους και τον τελικό παραλήπτη ή από κανένα. Στο προηγούμενο παράδειγμα μηνύματος SOAP βλέπουμε ότι το στοιχείο *h:reference* που βρίσκεται στο στοιχείο-επικεφαλίδα του μηνύματος κάνει χρήση της ιδιότητας *env:role* και της δίνει τιμή "<http://www.w3.org/2003/05/soap-envelope/role/next>" το οποίο σημαίνει ότι οι ενδιάμεσοι και ο τελικός παραλήπτης μπορούν να επεξεργαστούν το συγκεκριμένο στοιχείο.

Επιπλέον, η χρήση της τυποποιημένης ιδιότητας *env:mustUnderstand* σε κατάσταση 'true' για στοιχεία που βρίσκονται στο στοιχείο-επικεφαλίδα υποδεικνύει ότι ο ενδιάμεσος ή τελικός παραλήπτης ενός μηνύματος SOAP υποχρεούται να κατανοεί το περιεχόμενο του συγκεκριμένου στοιχείου προκειμένου να το επεξεργαστεί αλλιώς θα πρέπει να επιστρέψει μήνυμα σφάλματος (βλ. επόμενες παραγράφους). Στο παράδειγμα που δίνεται πιο πάνω βλέπουμε ότι το στοιχείο *h:reference* του στοιχείου-επικεφαλίδα πρέπει να είναι κατανοητό και τα περιεχόμενά του να τα επεξεργαστούν οι κόμβοι οι οποίοι υποδεικνύονται από την τιμή της ιδιότητας *env:role* (στο συγκεκριμένο παράδειγμα όλοι οι ενδιάμεσοι και ο τελικός παραλήπτης).

Η προεπιλεγμένη συμπεριφορά ενός κόμβου που επεξεργάζεται κάποιο στοιχείο επικεφαλίδας ενός SOAP μηνύματος είναι να αφαιρέσει το συγκεκριμένο στοιχείο από την επικεφαλίδα πριν προωθήσει το μήνυμα στον επόμενο κόμβο. Διατηρείται όμως η δυνατότητα να επαναπροσθέσει το στοιχείο που επεξεργάστηκε πριν την προώθηση. Υπάρχουν όμως περιπτώσεις όπου θέλουμε να σταλεί ένα μήνυμα SOAP σε έναν αριθμό κόμβων και μόνον όσοι επιλέγουν να το επεξεργαστούν να αφαιρέσουν το αντίστοιχο στοιχείο επικεφαλίδας αλλιώς να το προωθήσουν αναλλοίωτο. Σε αυτές τις περιπτώσεις μπορεί να γίνει χρήση της *env:relay* ιδιότητας η οποία μπορεί να τεθεί σε κατάσταση 'true'. Εάν όμως παράλληλα γίνει χρήση της ιδιότητας *env:mustUnderstand* σε κατάσταση 'true' ένας κόμβος είναι υποχρεωμένος να επεξεργαστεί ένα στοιχείο επικεφαλίδας εάν το κατανοεί.

Σε απάντηση του μηνύματος που δίνεται στο πιο πάνω παράδειγμα ο τελικός παραλήπτης μπορεί να απαντήσει με ένα άλλο SOAP μήνυμα όπως το εξής:

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <h:reference xmlns:h="http://stockinfo.example.com/reference"
```

```

    env:role="http://www.w3.org/2003/05/soap-envelope/role/next"
    env:mustUnderstand="true">
      <h:number>123456</h:number>
    </h:reference>
  </env:Header>
  <env:Body>
    <b:quoteresponse xmlns:b="http://stockinfo.example.com/stockupdate">
      <b:stockexchange> WXYZ </b:stockexchange>
      <b:stock> ABCD </b:stock>
      <b:price> 12.3 </b:price>
      <b:currency> EUR </b:currency>
      <b:quotetime> 2006-07-08T13:51:00.000+02:00 </b:quotetime>
    </b:quoteresponse>
  </env:Body>
</env:Envelope>

```

Αυτό το μήνυμα βλέπουμε ότι περιλαμβάνει τα ίδια στοιχεία στο στοιχείο-επικεφαλίδα και επιτρέπει το συσχετισμό του αρχικού μηνύματος με το απαντητικό μήνυμα με βάση το νούμερο αναφοράς (h:reference). Επίσης βλέπουμε ότι μαζί με τον κωδικό χρηματιστηρίου και μετοχής επιστρέφεται η τιμή της μετοχής (b:price) και η χρονική στιγμή κατά την οποία σημειώθηκε η συγκεκριμένη τιμή μετοχής (b:quotetime).

Δομή Μηνυμάτων Σφάλματος

Προβλέπεται συγκεκριμένη δομή και για τα μηνύματα σφάλματος τα οποία μπορούν να σημειωθούν κατά την επεξεργασία του στοιχείου-επικεφαλίδα ή του στοιχείου-σώμα ενός μηνύματος SOAP από τον τελικό παραλήπτη ή τους ενδιάμεσους.

Το κανάλι επικοινωνίας για την γνωστοποίηση του σφάλματος μπορεί να είναι το κανονικό κανάλι που διατίθεται από το πρωτόκολλο μεταφοράς ή το κανάλι σφάλματος του πρωτοκόλλου αναφοράς.

Τα μηνύματα σφάλματος επεξεργασίας του στοιχείου-σώματος περιλαμβάνουν ένα στοιχείο-σφάλμα (env:Fault) στο στοιχείο-σώμα του απαντητικού SOAP μηνύματος το οποίο περιέχει τυποποιημένα στοιχεία τα οποία καθορίζουν τον κωδικό λάθους (env:Code), την αιτία (env:Reason) και περαιτέρω λεπτομέρειες (env:Detail).

Μηνύματα σφάλματος που αφορούν την επεξεργασία στοιχείων του στοιχείου-επικεφαλίδα περιέχουν ένα στοιχείο-σφάλμα (env:Fault) πάλι στο στοιχείο-σώμα του μηνύματος σφάλματος αλλά επίσης υποδεικνύουν το στοιχείο του στοιχείου-επικεφαλίδα που προκάλεσε το σφάλμα.

Κωδικοποίηση Μηνυμάτων

Το SOAP προβλέπει ένα συγκεκριμένο τρόπο για την κωδικοποίηση δομημένων δεδομένων που μεταφέρονται μέσω της ανταλλαγής μηνυμάτων που υποστηρίζει. Με αυτό τον τρόπο το SOAP παρέχει ένα τρόπο σειριοποίησης (serialisation). Η χρήση αυτής της κωδικοποίησης είναι προαιρετική. Για αυτό το σκοπό το SOAP προβλέπει χρήση της ιδιότητας env:encodingStyle για στοιχεία που δηλώνονται στο στοιχείο-επικεφαλίδα ή στο στοιχείο-σώμα ενός SOAP μηνύματος. Με βάση την τιμή της ιδιότητας αυτής ο παραλήπτης ενός μηνύματος θα γνωρίζει εάν και πώς μπορεί να αποκωδικοποιηθεί το περιεχόμενο του στοιχείου.

Ο W3C προβλέπει επίσης τρόπους για τη βελτιστοποίηση της μετάδοσης μηνυμάτων SOAP ιδιαίτερα για τη μεταφορά περιεχομένου κωδικοποιημένου σε base64 (βλ. <http://www.w3.org/TR/soap12-mtom/>). Το base64 επιτρέπει την κωδικοποίηση δυαδικών δεδομένων (binary data) σε δεδομένα κειμένου (text data, ASCII data) ώστε να

είναι δυνατή η μεταφορά τους μέσω ηλεκτρονικού ταχυδρομείου ή άλλων εφαρμογών μεταφοράς κειμένου.

Συσχετισμός με το HTTP πρωτόκολλο (HTTP Binding)

Ο συσχετισμός του SOAP με το HTTP πρωτόκολλο έχει τυποποιηθεί από τον W3C. Σύμφωνα με το πρότυπο είναι δυνατή η υποστήριξη του μοντέλου αμφίδρομης επικοινωνίας για ανταλλαγή μηνυμάτων (request-response) μέσω του συσχετισμού του SOAP με τη μέθοδο HTTP POST. Σε αυτή την περίπτωση το αρχικό μήνυμα SOAP αποστέλλεται από τον κόμβο του Αρχικού Αποστολέα σε μία HTTP POST ΑΙΤΗΣΗ (request) ενώ το απαντητικό μήνυμα αποστέλλεται από τον Τελικό Παραλήπτη στην HTTP POST ΑΠΑΝΤΗΣΗ (response). Έτσι είναι δυνατός άμεσα ο συσχετισμός αρχικού μηνύματος και απαντητικού μηνύματος.

Έτσι, στο παράδειγμα με την τιμή μετοχής που παρατίθεται πιο πάνω εάν τα μηνύματα SOAP ανταλλάσσονται στα πλαίσια συσχετισμού του SOAP με τη μέθοδο HTTP POST, HTTP POST η αίτηση θα μπορούσε να είναι ως εξής:

```
POST /Reservations HTTP/1.1
Host: stockinfo.example.com
Content-Type: application/soap+xml; charset="utf-8"
Content-Length: nnn

<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <env:Header>
    <h:reference xmlns:h="http://stockinfo.example.com/reference"
      env:role="http://www.w3.org/2003/05/soap-envelope/role/next"
      env:mustUnderstand="true">
      <h:number>123456</h:number>
    </h:reference>
  </env:Header>
  <env:Body>
    <b:quoterequest xmlns:b="http://stockinfo.example.com/stockupdate">
      <b:stockexchange> WXYZ </b:stockexchange>
      <b:stock> ABCD </b:stock>
    </b:quoterequest>
  </env:Body>
</env:Envelope>
```

Η HTTP POST απάντηση με τη σειρά της θα μπορούσε να είναι ως εξής:

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset="utf-8"
Content-Length: nnn

<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <h:reference xmlns:h="http://stockinfo.example.com/reference"
    env:role="http://www.w3.org/2003/05/soap-envelope/role/next"
    env:mustUnderstand="true">
    <h:number>123456</h:number>
  </h:reference>
</env:Header>
<env:Body>
  <b:quoteresponse xmlns:b="http://stockinfo.example.com/stockupdate">
    <b:stockexchange> WXYZ </b:stockexchange>
    <b:stock> ABCD </b:stock>
    <b:price> 12.3 </b:price>
    <b:currency> EUR </b:currency>
    <b:quotetime> 2006-07-08T13:51:00.000+02:00 </b:quotetime>
  </b:quoteresponse>
</env:Body>
</env:Envelope>
```

Είναι επίσης δυνατή η ανταλλαγή μηνυμάτων SOAP μέσω του συσχετισμού του με τη μέθοδο HTTP GET. Ο συσχετισμός αυτός προβλέπεται όταν το αρχικό μήνυμα SOAP δεν περιλαμβάνει σημαντικό όγκο πληροφορίας αφού αποσκοπεί στο να ζητήσει την

αποστολή πληροφορίας. Έτσι, όποιες παράμετροι χρειάζεται να αποσταλούν από τον Αρχικό Αποστολέα κωδικοποιούνται ως ζεύγη τύπου-τιμής (type-value pairs) στο URL της HTTP GET ΑΙΤΗΣΗΣ (HTTP GET Request).

1.4 Η γλώσσα περιγραφής υπηρεσιών WSDL

Έχοντας περιγράψει τη λειτουργία του πρωτοκόλλου SOAP το οποίο χρησιμοποιείται για την επικοινωνία μεταξύ υπηρεσιών ιστού στη συνέχεια θα δούμε τα χαρακτηριστικά της γλώσσας περιγραφής υπηρεσιών ιστού Web Service Definition Language (WSDL).

Η WSDL αποτελεί το προϊόν τυποποίησης στον οργανισμό W3C και η πιο πρόσφατη έκδοσή της είναι αυτή του 2001 WSDL 1.1 (<http://www.w3.org/TR/wsdl>) ενώ ο W3C εργάζεται για μία νέα έκδοση την WSDL 2.0 (<http://www.w3.org/TR/wsdl20/>) η οποία αναμένεται να οριστικοποιηθεί άμεσα.

Η WSDL περιγράφει μια υπηρεσία ιστού με δομημένο τρόπο. Κάθε υπηρεσία ιστού προσφέρει μία ή περισσότερες Πόρτες Υπηρεσίας (ports). Η περιγραφή της κάθε Πόρτας Υπηρεσίας αποτελείται από τρία κύρια μέρη:

- Τύπος Πόρτας Υπηρεσιών (portType), που αναφέρεται στις Λειτουργίες (operations) που προσφέρονται από τον συγκεκριμένο τύπο πόρτας. Κάθε λειτουργία περιγράφεται ως ανταλλαγή μηνυμάτων η οποία μπορεί να γίνει μονόδρομα ή αμφίδρομα ή σύμφωνα με άλλα μοντέλα. Για κάθε μήνυμα που ανταλλάσσεται πρέπει να γίνει γνωστός ο τύπος των δεδομένων που μεταφέρει. Ο τρόπος περιγραφής των τύπων δεδομένων είναι συνήθως η χρήση του XMLSchema (<http://www.w3.org/XML/Schema>).
- Συσχετισμός (binding), που αναφέρεται στο συσχετισμό ενός τύπου πόρτας υπηρεσιών σε συγκεκριμένο πρωτόκολλο επικοινωνίας και σε συγκεκριμένο τρόπο αναπαράστασης των δεδομένων που ανταλλάσσονται. Το πιο σύνηθες πρωτόκολλο επικοινωνίας με το οποίο συσχετίζονται οι τύποι πόρτας που ορίζονται στην WSDL είναι το πρωτόκολλο SOAP.
- Διεύθυνση (address), που αναφέρεται στην ηλεκτρονική διεύθυνση από την οποία οι Λειτουργίες της συγκεκριμένης πόρτας υπηρεσιών είναι διαθέσιμες.

Με βάση τα παραπάνω θα ήταν αναμενόμενο ότι μία περιγραφή υπηρεσίας ιστού σε WSDL θα αποτελείται από δύο μέρη: ένα μέρος που δίνει μια γενική περιγραφή της υπηρεσίας ιστού και ένα άλλο μέρος με τις πόρτες υπηρεσιών για κάθε μία από τις οποίες προσδιορίζεται ο τύπος πόρτας υπηρεσιών, ο συσχετισμός και η διεύθυνση. Στην πραγματικότητα όμως, κάτι τέτοιο θα οδηγούσε σε αδικαιολόγητες επαναλήψεις σε μία περιγραφή σε WSDL. Αυτό θα οφείλονταν στο γεγονός ότι ένας ή περισσότεροι τύποι δεδομένων μπορούν να επαναχρησιμοποιηθούν σε διαφορετικά μηνύματα διαφορετικών λειτουργιών. Επίσης ένας συγκεκριμένος τύπος πόρτας υπηρεσιών μπορεί να επαναχρησιμοποιηθεί με τη σειρά του για τον ορισμό διαφορετικών πορτών υπηρεσιών όπου ο συγκεκριμένος τύπος συσχετίζεται με εναλλακτικούς συσχετισμούς με εναλλακτικά πρωτόκολλα επικοινωνίας σε διαφορετικές διευθύνσεις. Για αυτό το λόγο η δομή μιας περιγραφής υπηρεσίας ιστού σε WSDL είναι διαφορετική.

Μια περιγραφή σε WSDL αποτελεί ένα κείμενο XML το οποίο σέβεται τον ορισμό ενός συγκεκριμένου XML Schema (WSDL Schema definition). Σύμφωνα με αυτό ένα κείμενο XML έχει ένα στοιχείο-ρίζα το wsdl:definitions όπου το 'wsdl' αντιστοιχεί στον ονοματοχώρο (namespace) <http://schemas.xmlsoap.org/wsdl/>. Το στοιχείο wsdl:definitions περιέχει έναν αριθμό στοιχείων τα οποία μπορούν να είναι τα εξής:

- Τύποι δεδομένων (wsdl:types), που ορίζουν τους τύπους δεδομένων που ανταλλάσσονται μέσω των μηνυμάτων όταν χρησιμοποιούνται οι διάφορες λειτουργίες που προσφέρει μία υπηρεσία ιστού μέσω των πορτών υπηρεσιών που παρουσιάζει.
- Μηνύματα (wsdl:message), που ορίζουν τη δομή των μηνυμάτων που ανταλλάσσονται όταν χρησιμοποιούνται οι λειτουργίες των πορτών υπηρεσίας ιστού.
- Τύποι Πορτών Υπηρεσιών (wsdl:portType), που περιλαμβάνουν τις λειτουργίες (στοιχεία wsdl:operation) που μπορεί να προσφέρει ο συγκεκριμένος τύπος πόρτας υπηρεσιών. Με τη σειρά τους τα στοιχεία wsdl:operation προσδιορίζουν ως στοιχεία εισόδου (wsdl:input) και εξόδου (wsdl:output) τα μηνύματα που ανταλλάσσονται όπως αυτά έχουν οριστεί ξεχωριστά ως 'wsdl:message'.
- Συσχετισμοί (wsdl:binding), οι οποίοι για κάθε τύπο πόρτας υπηρεσιών (όπως έχει οριστεί ξεχωριστά ως wsdl:portType) προσδιορίζουν πως κάθε ανταλλασσόμενο μήνυμα στα πλαίσια κάθε λειτουργίας μπορεί να συσχετιστεί με ένα πρωτόκολλο ανταλλαγής μηνυμάτων. Το σύνθετο πρωτόκολλο ανταλλαγής μηνυμάτων είναι το πρωτόκολλο SOAP μέσω του πρωτοκόλλου HTTP.
- Υπηρεσίες (wsdl:service), οι οποίες περιλαμβάνουν ένα στοιχείο τεκμηρίωσης (wsdl:document) και στοιχεία πορτών υπηρεσιών (wsdl:port). Ένα στοιχείο πόρτας υπηρεσιών με τη σειρά του προσδιορίζει τον συσχετισμό που υλοποιεί όπως έχει οριστεί ξεχωριστά ως wsdl:binding όπως και τη διεύθυνση από την οποία είναι διαθέσιμες οι λειτουργίες της συγκεκριμένης πόρτας (wsdl:address).

Αν και η αυτή η δομή περιγραφών WSDL ίσως φαίνεται πιο πολύπλοκη εξασφαλίζει περιγραφές υπηρεσιών ιστού αποφεύγοντας επαναλήψεις. Ένα παράδειγμα περιγραφής μιας υπηρεσίας ιστού σε WSDL είναι το ακόλουθο:

```
<?xml version="1.0"?>

<definitions name="StockUpdateWebService"
targetNamespace="http://stockinfo.example.com/stockupdate.wsdl"
  xmlns:tns="http://stockinfo.example.com/stockupdate.wsdl"
  xmlns:xsd="http://stockinfo.example.com/stockupdate.xsd"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns="http://schemas.xmlsoap.org/wsdl/">

  <types>
    <schema targetNamespace="http://stockinfo.example.com/stockupdate.xsd"
      xmlns="http://www.w3.org/2000/10/XMLSchema">
      <element name="stockrequest">
        <complexType>
          <all>
            <element name="stockexchange" type="string"/>
            <element name="stock" type="string"/>
          </all>
        </complexType>
      </element>
      <element name="stockresponse">
        <complexType>
          <all>
            <element name="stockexchange" type="string"/>
            <element name="stock" type="string"/>
            <element name="price" type="float"/>
            <element name="currency" type="string"/>
            <element name="quotetime" type="dateTime"/>
          </all>
        </complexType>
      </element>
    </schema>
  </types>
```

```

</types>

<message name="StockUpdateInput">
  <part name="body" element="xsd1:stockrequest"/>
</message>

<message name="StockUpdateOutput">
  <part name="body" element="xsd1:stockresponse"/>
</message>

<portType name="StockUpdatePortType">
  <operation name="StockUpdate">
    <input message="tns:StockUpdateInput"/>
    <output message="tns:StockUpdateOutput"/>
  </operation>
</portType>

<binding name="StockUpdateSoapBinding" type="tns:StockUpdatePortType">
  <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="StockUpdate">
    <soap:operation soapAction="http://stockinfo.example.com/stockupdate"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>
</binding>

<service name="StockUpdateService">
  <documentation>Live Stock Price Updates from the Markets</documentation>
  <port name="StockUpdatePort" binding="tns:StockUpdateSoapBinding">
    <soap:address location="http://stockinfo.example.com/stockquote"/>
  </port>
</service>

</definitions>

```

Σε αυτό το παράδειγμα βλέπουμε τον ορισμό της υπηρεσίας ιστού *StockUpdate* σε WSDL. Πρόκειται για ένα κείμενο XML το οποίο στο στοιχείο-ρίζα `wsdl:definitions`¹ περιλαμβάνει στοιχεία ορισμού τύπων δεδομένων (`wsdl:types`), μηνυμάτων (`wsdl:message`), τύπου πόρτας υπηρεσιών (`wsdl:portType`) και λειτουργίας (`wsdl:operation`), συσχετισμού (`wsdl:binding`) και υπηρεσίας (`wsdl:service`). Παρατηρούμε ότι δηλώνονται δύο τύποι δεδομένων `xsd1:stockrequest` και `xsd1:stockresponse` οι οποίοι χρησιμοποιούνται στα μηνύματα `tns:StockUpdateInput` και `tns:StockUpdateOutput` αντίστοιχα. Τα μηνύματα αυτά με τη σειρά τους ορίζονται ως μηνύματα εισόδου και εξόδου αντίστοιχα της λειτουργίας `tns:StockUpdate` η οποία προσφέρεται στον τύπο πόρτας υπηρεσιών `tns:StockUpdatePortType`. Ακολούθως δηλώνεται ο συσχετισμός του συγκεκριμένου τύπου πόρτας υπηρεσιών με το πρωτόκολλο SOAP μέσω HTTP με τη δήλωση του στοιχείου `soap:binding`. Τέλος ορίζεται η υπηρεσία `tns:StockUpdateService` στην οποία δηλώνεται ως πόρτα υπηρεσιών ο συσχετισμός του `tns:StockUpdatePortType` με το πρωτόκολλο SOAP και με το στοιχείο `soap:address` δίνεται η διεύθυνση από την οποία η συγκεκριμένη πόρτα είναι διαθέσιμη.

Εάν παρατηρήσουμε πιο προσεκτικά τη δήλωση του στοιχείου `soap:binding` στο συγκεκριμένο παράδειγμα βλέπουμε ότι δηλώνει δύο ιδιότητες: `wsdl:style` και `wsdl:transport` όπως φαίνεται στο συγκεκριμένο τμήμα του κώδικα του παραδείγματος:

```

<soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>

```

¹ Σημειώστε ότι η χρήση του προθέματος 'wsdl' για τον ονοματοχώρο <http://schemas.xmlsoap.org/wsdl/> δεν είναι απαραίτητη αφού αυτός ορίζεται ως ο προεπιλεγμένος ονοματοχώρος με τη δήλωση `xmlns` στον ορισμό του στοιχείου `wsdl:definitions`.

Η ιδιότητα `wsdl:style` έχει να κάνει με τον τρόπο με τον οποίο πρέπει να κωδικοποιηθούν τα μηνύματα SOAP που θα ανταλλαχθούν όταν κληθεί η συγκεκριμένη πόρτα υπηρεσιών. Έτσι, μπορεί να πάρει τη τιμή *document*, όπως στο συγκεκριμένο παράδειγμα, όπου το μήνυμα SOAP που θα σταλεί για την χρήση της συγκεκριμένης πόρτας υπηρεσιών θα είναι δομημένο σύμφωνα με αυτά που ορίζονται στο στοιχείο `wsdl:types`. Δηλαδή στο στοιχείο-σώμα του μηνύματος θα περιέχονται τα απαιτούμενα στοιχεία για τη χρήση της πόρτας. Μπορεί επίσης να πάρει την τιμή *rpc* όπου όμως το μήνυμα SOAP θα πρέπει να είναι δομημένο με ένα συγκεκριμένο τρόπο όπως ορίζεται από το πρότυπο SOAP. Δηλαδή το στοιχείο-σώμα του μηνύματος θα περιέχει ένα στοιχείο που θα αντιστοιχεί στο όνομα της καλούμενης λειτουργίας της πόρτας (για το παράδειγμά μας αυτό θα ήταν *StockUpdate* όπως ορίζεται στο `wsdl:operation`). Αυτό το στοιχείο με τη σειρά του περιλαμβάνει στοιχεία που αντιστοιχούν στις παραμέτρους κλήσης της λειτουργίας.

Η ιδιότητα `wsdl:transport` έχει να κάνει με το πρωτόκολλο μέσω του οποίου θα μεταφερθούν τα μηνύματα SOAP. Στο συγκεκριμένο παράδειγμα βλέπουμε ότι ορίζεται ως τέτοιο πρωτόκολλο το HTTP με τη χρήση του URI <http://schemas.xmlsoap.org/soap/http>. Άλλα URI μπορούν να δηλώσουν χρήση άλλων πρωτοκόλλων μεταφοράς όπως το SMTP.

Τέλος παρατηρούμε το εξής τμήμα στη δήλωση του στοιχείου συσχετισμού (`wsdl:binding`) του παραπάνω παραδείγματος:

```
<input>
  <soap:body use="literal"/>
</input>
<output>
  <soap:body use="literal"/>
</output>
```

Η χρήση της ιδιότητας *'use'* στη δήλωση των στοιχείων `soap:body` μπορεί να πάρει την τιμή *literal* όπως στο συγκεκριμένο παράδειγμα. Αυτό δηλώνει ότι οι παράμετροι στα μηνύματα εισόδου και εξόδου κωδικοποιούνται σύμφωνα με ένα συγκεκριμένο XML Schema (το οποίο στο παράδειγμα αυτό ορίζεται στο στοιχείο `wsdl:types`). Αυτός ο τρόπος ενδείκνυται για παραμέτρους με δενδρικές δομές.

Η ιδιότητα *'use'* μπορεί επίσης να πάρει την τιμή *encoded*. Σε αυτή την περίπτωση οι παράμετροι των μηνυμάτων κωδικοποιούνται σύμφωνα με κανόνες που ορίζει το πρότυπο SOAP και δεν είναι απαραίτητο να συμφωνούν με ένα XML Schema. Αυτός ο τρόπος ενδείκνυται για δεδομένα με μη δενδρικές δομές (κυκλικές δομές).

1.5 Το πρωτόκολλο ανακάλυψης υπηρεσιών UDDI

Το πρωτόκολλο UDDI (<http://www.uddi.org/>) άρχισε να αναπτύσσεται το 2001 από τον οργανισμό OASIS Open, Organization for the Advancement of Structured Information Standards (<http://www.oasis-open.org>). Η εργασίες για το UDDI ξεκίνησαν πριν ακόμα ξεκινήσουν οι εργασίες για την WSDL από τον W3C. Έτσι, υπάρχουν κάποια προβλήματα και ασυμβατότητες μεταξύ του UDDI και της WSDL.

Μέχρι στιγμής το UDDI δεν έχει την επιτυχία που έχουν γνωρίσει το SOAP και η WSDL και για ορισμένους δεν συγκαταλέγεται στα βασικά πρωτόκολλα για ανάπτυξη Αρχιτεκτονικών Υπηρεσιών με Υπηρεσίες Ιστού (Newcomer, Lomow, 2005). Ένας από τους λόγους που δεν έχει γνωρίσει αυτή την επιτυχία ίσως είναι ότι οι επιχειρήσεις φάνηκαν απρόθυμες να το χρησιμοποιήσουν για να ξεκινήσουν συνεργασία με επιχειρήσεις τις οποίες δεν γνώριζαν από πριν (Newcomer, Lomow 2005). Όμως η λειτουργία της ανακάλυψης υπηρεσιών ιστού είναι αρκετά βασική στις Αρχιτεκτονικές Υπηρεσιών και για πολλούς οι λειτουργίες που προσφέρει το UDDI απαραίτητες. Ο OASIS Open το 2005 ανακοίνωσε την τυποποίηση μιας

νεότερης έκδοσης του πρωτοκόλλου UDDI v3. Με τις βελτιώσεις που παρέχει η νέα έκδοση είναι δυνατό να τύχει ευρύτερης αποδοχής.

Το UDDI παρέχει έναν τρόπο ανακάλυψης υπηρεσιών ιστού που μπορεί να προσφέρονται από έναν αριθμό επιχειρήσεων προκειμένου να υπάρξει συνεργασία ιδιαίτερα στα πλαίσια του ηλεκτρονικού επιχειρείν. Έτσι, προσφέρεται πρόσβαση σε έναν κατάλογο που περιλαμβάνει τις επιχειρήσεις, τις υπηρεσίες ιστού που αυτές προσφέρουν και τον τρόπο με τον οποίο οι υπηρεσίες αυτές θα μπορούσαν να χρησιμοποιηθούν. Πιο συγκεκριμένα, το μοντέλο δεδομένων που αποθηκεύει το UDDI περιλαμβάνει τους ακόλουθους τύπου στοιχείων:

- *businessEntity*. Που περιγράφει επιχειρήσεις ή οργανισμούς που προσφέρουν υπηρεσίες ιστού. Στην ουσία μπορεί να περιγράψει οποιαδήποτε οντότητα προσφέρει υπηρεσίες ιστού σε διαφορετικές γλώσσες εάν απαιτείται και να παρέχει λεπτομέρειες για το πως μπορεί να επικοινωνήσει κάποιος μαζί της, κλπ. Τέτοια στοιχεία μπορούν να κατηγοριοποιηθούν με διάφορους τρόπους και μπορούν να σχετιστούν με έναν αριθμό υπηρεσιών, οι οποίες περιγράφονται ως στοιχεία *businessService*.
- *businessService*: Τέτοια στοιχεία περιγράφουν μια συλλογή σχετικών μεταξύ τους υπηρεσιών ιστού που προσφέρονται από μία επιχείρηση ή οργανισμό η περιγραφή του οποίου δίνεται ως περιγραφή *businessEntity*. Επίσης, στοιχεία *businessService* μπορούν να σχετιστούν με έναν αριθμό στοιχείων *bindingTemplate*.
- *bindingTemplate*: Αυτά τα στοιχεία παρέχουν τις τεχνικές λεπτομέρειες προκειμένου να χρησιμοποιηθεί μια υπηρεσία ιστού. Περιλαμβάνουν τις δικτυακές διευθύνσεις όπου οι υπηρεσίες είναι διαθέσιμες όπως επίσης και Ψηφιακές Υπογραφές (digital signatures). Επίσης στοιχεία *bindingTemplate* μπορούν να σχετιστούν με έναν αριθμό στοιχείων *tModel*.
- *tModel*: Αυτού του είδους τα στοιχεία αποτελούν επαναχρησιμοποιήσιμες έννοιες που προσφέρουν το τεχνικό μοντέλο (technical model) περιγραφής διαφορετικών στοιχείων όπως ένας τύπος υπηρεσίας ιστού, ένα πρωτόκολλο για επικοινωνία με υπηρεσίες ιστού ή ένα σύστημα κατηγοριοποίησης. Έτσι, στοιχεία *tModel* μπορούν να αναπαριστούν πρωτόκολλα μεταφοράς δεδομένων όπως HTTP ή SMTP, φορμά ταχυδρομικών διευθύνσεων, κ.α.
- *publisherAssertion*: Που περιγράφουν σχέσεις μεταξύ στοιχείων *businessEntity*. Έτσι, είναι δυνατό να ορισθεί ότι μία επιχείρηση είναι θυγατρική κάποιας άλλης, κλπ. Το UDDI ορίζει έναν αριθμό πιθανών σχέσεων.
- *subscription*: Που περιγράφουν αιτήματα για ενημέρωση (ειδοποίηση) σχετικά με ενημερώσεις στοιχείων ή τύπων στοιχείων στους καταλόγους.

Οι τύποι στοιχείων *businessEntity*, *businessService*, *bindingTemplate*, *tModel* και *publisherAssertion* μπορούν να έχουν τις δικές τους ψηφιακές υπογραφές (digital signatures) σύμφωνα με την έκδοση UDDI v3.

Μία αρχιτεκτονική που να υποστηρίζει τη φιλοξενία και χρήση καταλόγων σύμφωνα με το UDDI μπορεί να περιλαμβάνει *UDDI Κόμβους (nodes)*, οι οποίοι παρέχουν καταλόγους χρησιμοποιώντας το μοντέλο δεδομένων που περιγράφεται πιο πάνω και *Πελάτες UDDI Κόμβων (clients)*. Τουλάχιστον ένας κόμβος πρέπει να υλοποιείται σε κάθε αρχιτεκτονική UDDI. Οι πελάτες κόμβων μπορούν να χρησιμοποιούν τις υπηρεσίες κόμβων UDDI. Είναι δυνατή η χρήση υπηρεσιών ενός κόμβου UDDI από κάποιο άλλο κόμβο, όπως επίσης και η χρήση υπηρεσιών του πελάτη ενός κόμβου

UDDI από τον κόμβο (π.χ. για τη μετάδοση ειδοποιήσεων για τις οποίες ο πελάτης έχει συνάψει συνδρομές).

Το πρότυπο UDDI προβλέπει ένα σύνολο από Διεπαφές Προγραμματισμού Εφαρμογών (Application Programming Interface, API) προκειμένου να χρησιμοποιηθούν οι υπηρεσίες που προσφέρονται από UDDI Κόμβους αλλά και Πελάτες UDDI Κόμβων. Ένας κόμβος παρέχει τις εξής ομάδες διεπαφών σύμφωνα με το UDDI v3:

- *UDDI Inquiry*. Με λειτουργίες αναζήτησης στοιχείων στον κατάλογο που τηρεί ο κόμβος και απόκτηση πληροφορίας από τον κατάλογο.
- *UDDI Publication*. Με λειτουργίες που επιτρέπουν την πρόσθεση, ενημέρωση ή διαγραφή στοιχείων από τον κατάλογο που διατηρεί ο κόμβος.
- *UDDI Security*. Με λειτουργίες που εξυπηρετούν προβλεπόμενους μηχανισμούς Επικύρωσης (authentication).
- *UDDI Custody Transfer*. Με λειτουργίες που επιτρέπουν την απόκτηση ή μεταφορά ενός ή περισσότερων στοιχείων businessEntity και των σχετικών businessService στοιχείων όπως επίσης και στοιχεία tModel από ένα κόμβο σε έναν άλλο.
- *UDDI Subscription*. Που υποστηρίζει τις συνδρομές για ειδοποίηση όταν λαμβάνουν χώρα αλλαγές στον κατάλογο που υποστηρίζει ο κόμβος.

Με τη σειρά του, ένας Πελάτης Κόμβου UDDI σύμφωνα με το UDDI v3 προσφέρει τις εξής ομάδες διεπαφών:

- *UDDI Subscription Listener*. Που επιτρέπει στον Κόμβο UDDI να ειδοποιεί τον πελάτη όταν γίνονται αλλαγές στον κατάλογο για τις οποίες ο πελάτης έχει ζητήσει να ενημερώνεται μέσω συνδρομής (χρησιμοποιώντας τις λειτουργίες UDDI Subscription που προσφέρει ο κόμβος).
- *UDDI Value Set*. Με λειτουργίες σχετικές με κατηγοριοποίηση και συσχετισμό των εννοιών που χρησιμοποιούνται.

Η επικοινωνία μεταξύ πελατών και UDDI κόμβων γίνεται με χρήση SOAP μέσω του πρωτοκόλλου HTTP. Έτσι, σε μια αρχιτεκτονική υπηρεσιών μια υπηρεσία ιστού-πελάτης μπορεί να αναζητήσει άλλες υπηρεσίες ιστού στον κατάλογο του UDDI με το να χρησιμοποιήσει τις λειτουργίες UDDI Inquiry του τοπικού κόμβου. Η περιγραφή των υπηρεσιών που αποκτώνται από τον κατάλογο και οι λεπτομέρειες για τον τρόπο επικοινωνίας μαζί τους περιγράφονται από τα στοιχεία του καταλόγου.

Γίνεται έτσι φανερό η ανάγκη να ορισθούν τρόποι συσχετισμού του τρόπου περιγραφής μιας υπηρεσίας ιστού στο UDDI με τον τρόπο περιγραφής σύμφωνα με την WSDL. Αυτό επιτρέπει σε υπηρεσίες ιστού που περιγράφονται σε WSDL να προστεθούν στους καταλόγους UDDI αλλά και σε πελάτες κόμβων UDDI να αποκτήσουν πρόσβαση σε WSDL περιγραφή των υπηρεσιών ιστού που ανακαλύπτουν. Οι εργασίες που έχουν γίνει μέχρι στιγμής στα πλαίσια του OASIS Open προβλέπουν αντιστοίχιση των στοιχείων portType, binding, service και port της WSDL με τα στοιχεία tModel, tModel businessService και bindingTemplate του UDDI αντίστοιχα.

Κατάλογοι UDDI μπορούν να υποστηριχθούν από κόμβους και να είναι διαθέσιμοι μόνο στο εσωτερικά στο δίκτυο μιας επιχείρησης (προκειμένου να εξυπηρετήσουν λειτουργίες της αρχιτεκτονικής υπηρεσιών εσωτερικά) ή και εξωτερικά προκειμένου

να εξυπηρετήσουν συναλλαγές με τρίτους. Είναι επίσης δυνατή και η ανάπτυξη υβριδικών μοντέλων όπου ορισμένα στοιχεία των καταλόγων UDDI είναι διαθέσιμα εσωτερικά ενώ διαφορετικά στοιχεία είναι διαθέσιμα εξωτερικά.

1.6 Σύνθεση Υπηρεσιών Ιστού

Ένας σημαντικός αριθμός προτύπων για το συνδυασμό υπηρεσιών ιστού ο οποίος παρουσιάζεται είτε ως *Ενορχήστρωση* ή ως *Χορογραφία Υπηρεσιών Ιστού*. Η ενορχήστρωση υπηρεσιών ιστού (Web Service Orchestration) προϋποθέτει ότι κάποιος είναι σε θέση κεντρικά να προσδιορίσει τις υπηρεσίες ιστού που πρόκειται να συνδυαστούν και να παρακολουθεί και να ελέγχει την εξέλιξη της συνεργασίας τους. Η γλώσσα Web Services Business Process Execution Language ή WSBPEL (http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel) προσφέρεται για τον ορισμό προδιαγραφών ενορχήστρωσης υπηρεσιών ιστού. Η γλώσσα WSBPEL αναπτύσσεται από τον οργανισμό OASIS, Organization for the Advancement of Structured Information Standards (<http://www.oasis-open.org>), ο οποίος προωθεί τη δημιουργία προτύπων για ηλεκτρονικό επιχειρείν.

Η χορογραφία υπηρεσιών ιστού διαφέρει από την ενορχήστρωση στο ότι δεν υπάρχει κεντρικός έλεγχος αλλά οι υπηρεσίες ιστού επικοινωνούν μεταξύ του σε βάση ίσος-προς-ίσο (peer-to-peer) και συνεργάζονται σύμφωνα με «χορογραφίες» επικοινωνίας και ανταλλαγής μηνυμάτων που έχει η κάθε υπηρεσία στη διάθεσή της. Ο οργανισμός W3C έχει αναπτύξει ειδική ομάδα εργασίας με θέμα τη χορογραφία υπηρεσιών ιστού (<http://www.w3.org/2002/ws/chor/>) ενώ υπάρχει δραστηριότητα για την τυποποίηση μιας γλώσσας χορογραφίας, της Web Service Choreography Description Language (WS-CDL) (<http://www.w3.org/TR/2006/WD-ws-cdl-10-primer-20060619/>).

Έτσι, όταν πρόκειται για σύνθεση υπηρεσιών ιστού υπάρχουν προσεγγίσεις με βάση την ενορχήστρωση υπηρεσιών (orchestration-centric) και προσεγγίσεις με βάση τη χορογραφία υπηρεσιών (choreography-centric) από τις οποίες υποστηρίζεται ότι οι πιο χαλαρές σύζευξης είναι αυτές που στηρίζονται σε χορογραφία (Newcomer, Lomow, 2005).

2 Σημασιολογικές Υπηρεσίες Ιστού (Semantic Web Services)

Όπως προκύπτει από την ανάλυση των τεχνολογιών για υπηρεσίες ιστού και αρχιτεκτονικές υπηρεσιών τα υπάρχοντα πρότυπα ορίζουν τον τρόπο με τον οποίο είναι δυνατό να αναζητηθούν υπηρεσίες ιστού και να χρησιμοποιηθούν. Η αναζήτηση γίνεται με βάση κάποιες ιδιότητές τους, ενώ η περιγραφή των λειτουργιών που προσφέρουν με βάση τα μηνύματα που μπορούν να λαμβάνουν και να στέλνουν. Επίσης ορίζονται τα πρότυπα που αφορούν την υλοποίηση του πλαισίου ανταλλαγής μηνυμάτων μεταξύ υπηρεσιών ιστού.

Ας φανταστούμε ένα περιβάλλον όπου μια πληθώρα παρόχων προσφέρουν υπηρεσίες ιστού οι οποίες θα είναι συνδυάσιμες με υπηρεσίες ιστού του ιδίου ή και άλλων παρόχων προκειμένου να προσφέρουν προηγμένες και εξατομικευμένες υπηρεσίες στους χρήστες και να υλοποιήσουν τις πιο σύνθετες εταιρικές διαδικασίες που απαιτούνται για αυτό το σκοπό (όπως στα παραδείγματα που παρουσιάζονται στην ενότητα 1). Έξυπνοι διαμεσολαβητές αναλαμβάνουν τα καθήκοντα εύρεσης σχετικών υπηρεσιών ιστού και διαπραγμάτευσης προκειμένου να γίνει συμφωνηθεί και να υλοποιηθεί η συνεργασία δυναμικά και με τρόπο (ημι)-αυτοματοποιημένο. Ένα τέτοιο σενάριο συνοψίζει το όραμα πίσω από την ιδέα των αρχιτεκτονικών υπηρεσιών σε πλήρη ανάπτυξη.

Θα διαπιστώσουμε όμως ότι τα υπάρχοντα πρότυπα δεν επαρκούν για να υλοποιηθεί αυτό το όραμα άμεσα. Ένας σημαντικός λόγος για αυτή την ανεπάρκεια είναι η απουσία περιγραφής των λειτουργιών που προσφέρονται από τις υπηρεσίες ιστού με τρόπο που να είναι ακριβής και να μπορεί να είναι επεξεργάσιμος από προγράμματα αναζήτησης και χρήσης υπηρεσιών.

Η δυναμική ανακάλυψη υπηρεσιών ιστού γίνεται δύσκολη ή λιγότερο αποδοτική αφού οι τρόποι αναζήτησης που προβλέπει αυτή τη στιγμή το πρότυπο UDDI δεν περιλαμβάνουν σημασιολογική περιγραφή των υπηρεσιών που βρίσκονται στον κατάλογο, των προϋποθέσεων ή των τρόπων χρήσης τους και της επίδρασης που κάθε υπηρεσία έχει στον πραγματικό κόσμο. Για παράδειγμα, ένα πάροχος χρηματιστηριακών υπηρεσιών που επιθυμεί να δώσει μια σημασιολογική περιγραφή της υπηρεσίας *buyStocks* που προσφέρει και να περιγράψει το αποτέλεσμα αυτής της συναλλαγής στον πραγματικό κόσμο (αφαίρεση απαιτούμενου ποσού αγοράς από το λογαριασμό του αγοραστή και πρόσθεση μετοχών στο χαρτοφυλάκιό του) δεν υποστηρίζεται από τα παραπάνω πρότυπα. Επίσης, ο πάροχος χρηματιστηριακών υπηρεσιών που προσφέρει την προηγμένη υπηρεσία έξυπνου διαμεσολαβητή για διαχείριση χαρτοφυλακίου δεν είναι σε θέση να την διαφημίσει και να την περιγράψει με μια σημασιολογική περιγραφή. Με αυτό τον τρόπο περιορίζεται το εύρος χρήσης υπηρεσιών ιστού και το εύρος υλοποίησης του οράματος αρχιτεκτονικών υπηρεσιών.

Για αυτό το λόγο, η ερευνητική κοινότητα τόσο στα ακαδημαϊκά ιδρύματα όσο και στη βιομηχανία, όπως επίσης και οι οργανισμοί τυποποίησης εργάζονται σε τρόπους περιγραφής, ανακάλυψης και χρήσης Σημασιολογικών Υπηρεσιών Ιστού (Semantic Web Services) τα οποία παρουσιάζονται στις επόμενες παραγράφους.

2.1 Τι είναι οι Σημασιολογικές Υπηρεσίες Ιστού

Οι *Σημασιολογικές Υπηρεσίες Ιστού* (Semantic Web Services) αποτελούν υπηρεσίες ιστού που περιλαμβάνουν *σημασιολογική περιγραφή* η οποία επιτρέπει την δυναμική ανακάλυψη, χρήση ή συνδυασμό τους από έξυπνους διαμεσολαβητές υλοποιώντας έτσι ένα σημαντικό μέρος του οράματος των αρχιτεκτονικών υπηρεσιών.

Συγκεκριμένα, οι Σημασιολογικές Υπηρεσίες Ιστού (ΣΥΙ) επιτρέπουν μέσω της σημασιολογικής περιγραφής τους:

- *Χρήση της ΣΥΙ από έξυπνους διαμεσολαβητές.* Η σημασιολογική περιγραφή μιας ΣΥΙ περιγράφει με ακρίβεια πως ένας πελάτης της ΣΥΙ μπορεί να χρησιμοποιήσει τις υπηρεσίες της. Έτσι, είναι δυνατή η χρήση της υπηρεσίας από έξυπνους διαμεσολαβητές με τρόπο που να μην απαιτεί ανθρώπινη παρέμβαση. Ένας έξυπνος διαμεσολαβητής θα είναι σε θέση για παράδειγμα να βρει τους τύπους των παραμέτρων κλήσης των λειτουργιών της ΣΥΙ, να βρει τις τιμές των παραμέτρων που πρέπει να χρησιμοποιηθούν, την περιγραφή της υπηρεσίας σε WSDL και τον μοντέλο ανταλλαγής μηνυμάτων που υποστηρίζεται από την υπηρεσία. Μετά την χρήση της ΣΥΙ ο έξυπνος διαμεσολαβητής μπορεί αντίστοιχα να επεξεργαστεί τα αποτελέσματα.
- *Ανακάλυψη ΣΥΙ από έξυπνους διαμεσολαβητές.* Κάθε υπηρεσία ιστού έχει κάποιες *Δυνατότητες* (capabilities). Αυτές οι δυνατότητες περιγράφονται σημασιολογικά με τρόπο ώστε να είναι δυνατή η ανακάλυψη των υπηρεσιών που τις υποστηρίζουν από έξυπνους διαμεσολαβητές μέσω του *Ταιριάσματος* (matchmaking) των δυνατοτήτων που απαιτούνται από το χρήστη (και τον διαμεσολαβητή) και των δυνατοτήτων που παρέχουν οι διάφορες ΣΥΙ. Η περιγραφή των δυνατοτήτων των ΣΥΙ μπορεί να γίνει με δύο τρόπους. Ο ένας τρόπος είναι με τη χρήση μιας οντολογίας που προσφέρει ταξινομίες για υπηρεσίες ιστού. Ο άλλος τρόπος είναι προσφέροντας μια σημασιολογική

περιγραφή της κάθε λειτουργίας της ΣΥΙ ως *Μετασχηματισμό Καταστάσεων* (*state transformation*). Δηλαδή μια λειτουργία λαμβάνει κάποια δεδομένα και μετασχηματίζει την τρέχουσα κατάσταση των πραγμάτων αντίστοιχα. Για παράδειγμα, λαμβάνει τις ημερομηνίες πτήσης ενός πελάτη και την πιστωτική του κάρτα και προκαλεί μια μετάβαση σε μια νέα κατάσταση όπου ο πελάτης βρίσκεται στον προορισμό του και έχει γίνει η αντίστοιχη χρέωση στην πιστωτική κάρτα του. Ο συγκεκριμένος τρόπος σημασιολογικής περιγραφής λειτουργιών των ΣΥΙ είναι ιδιαίτερα χρήσιμος για εφαρμογές που χρησιμοποιούν γλώσσες *Σχεδιασμού Ενεργειών Τεχνητής Νοημοσύνης* (AI Planning).

- *Ενορχήστρωση ΣΥΙ από έξυπνους διαμεσολαβητές*. Μια υπηρεσία ιστού που έχει αποτελέσματα στον πραγματικό κόσμο όπως αυτή που περιγράφεται στο προηγούμενο παράδειγμα μπορεί δέχεται κάποια Είσοδο, να επιστρέφει κάποια Έξοδο και με δεδομένες Προϋποθέσεις έχει ορισμένα Αποτελέσματα στον πραγματικό κόσμο. Τέτοιες υπηρεσίες τις ονομάζουμε *Υπηρεσίες ΕΕΠΑ* από τις λέξεις Είσοδος, Έξοδος, Προϋποθέσεις, Αποτελέσματα. Υπάρχουν όμως και υπηρεσίες οι οποίες είναι απλά πληροφοριακού χαρακτήρα οι οποίες δέχονται δεδομένα ως Είσοδο και επιστρέφουν δεδομένα ως Έξοδο τις οποίες ονομάζουμε *Υπηρεσίες ΕΕ* αντίστοιχα. Οι σημασιολογικές υπηρεσίες ιστού προσφέρουν σημασιολογική περιγραφή όλων αυτών των στοιχείων η οποία επιτρέπει σε έξυπνους διαμεσολαβητές με βάση τα δεδομένα που έχουν διαθέσιμα, τα δεδομένα που περιμένουν να αποκτήσουν, την υπάρχουσα κατάσταση (προϋποθέσεις) και την επιθυμητή κατάσταση στον πραγματικό κόσμο (αποτελέσματα) να συνδυάσουν την χρήση ενός ή περισσότερων ΣΥΙ. Αν και υπάρχουν γλώσσες που περιγράφουν την Ενορχήστρωση Υπηρεσιών Ιστού όπως η WSBPEL (που περιγράφεται στην ενότητα 1.6) ο τρόπος με τον οποίο επιλέγονται οι ΣΥΙ που λαμβάνουν μέρος στην ενορχήστρωση υπηρεσιών είναι ξεχωριστό αντικείμενο μελέτης και υπάρχουν διάφορες προτάσεις βασισμένες σε Σχεδιασμό Ενεργειών οι οποίες παρατίθενται στη συνέχεια.

Δεδομένων των προκλήσεων τόσο στον τομέα τον τεχνολογικό όσο και στον τομέα της τυποποίησης που συνεπάγεται η ανάπτυξη, χρήση και υποστήριξη ΣΥΙ έχει καταβληθεί αρκετή προσπάθεια αρχικά για την ανάπτυξη μιας γλώσσας σημασιολογικής περιγραφής υπηρεσιών ιστού και έπειτα για τεχνολογίες που υποστηρίζουν με τον καλύτερο τρόπο όλες τις λειτουργίες που παρατίθενται πιο πάνω (Pandis 2004).

Ο οργανισμός W3C στεγάζει ομάδα ειδικού ενδιαφέροντος (Semantic Web Services Interest Group, SWS-IG) για σημασιολογικές υπηρεσίες ιστού η οποία είναι ανοιχτή στο κοινό και όπου συζητούνται διάφορες πτυχές και προτάσεις για ΣΥΙ από την ακαδημαϊκή κοινότητα και τη βιομηχανία (W3C Semantic Web Services Interest Group <http://www.w3.org/202/ws/swsiq/>).

Διάφορες γλώσσες σημασιολογικής περιγραφής υπηρεσιών ιστού όπως η OWL-S παρουσιάζονται σε αυτή την ενότητα ενώ ακολουθούν τεχνολογίες Σχεδιασμού Ενεργειών για τη χρήση και το συνδυασμό ΣΥΙ.

2.2 Η γλώσσα περιγραφής σημασιολογικών υπηρεσιών ιστού OWL-S

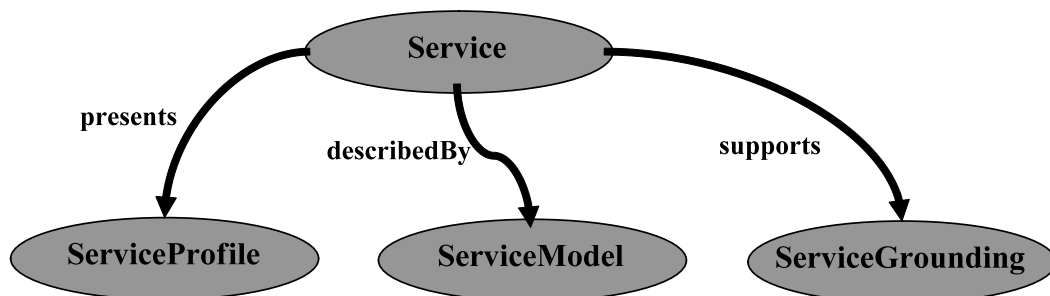
Η OWL-S (OWL Web Ontology Language for Services) είναι γλώσσα περιγραφής σημασιολογικών υπηρεσιών ιστού. Η OWL-S διαδέχεται την γλώσσα DAML for Services (DAML-S) (<http://www.daml.org/services>) η οποία αναπτύχθηκε στα πλαίσια του προγράμματος (DARPA Agent Markup Language) που χρηματοδοτήθηκε από την

DARPA (<http://www.daml.org/>). Πρόσφατη έκδοση της OWL-S² έχει υποβληθεί στον οργανισμό W3C για τυποποίηση ενώ αποτελεί αντικείμενο συζητήσεων στα πλαίσια της ομάδας SWS-IG του W3C.

Η γλώσσα OWL-S χρησιμοποιεί μια *Ανώτερη Οντολογία* (Upper Ontology) για σημασιολογική περιγραφή υπηρεσιών ιστού. Ο σκοπός της οντολογίας είναι να προσφέρει τον τρόπο περιγραφής μιας υπηρεσίας ώστε να απαντώνται ερωτήματα όπως: (α) τι είναι σε θέση να προσφέρει μια υπηρεσία στους πελάτες της, (β) πως μπορεί να χρησιμοποιηθεί μια υπηρεσία και (γ) πως γίνεται η αλληλεπίδραση με την υπηρεσία.

Έτσι, αυτή η οντολογία προβλέπει αρχικά την κλάση *Service* η οποία αναπαριστά μια υπηρεσία ιστού. Για κάθε στιγμιότυπο της κλάσης *Service* υπάρχει και μια υλοποίηση της αντίστοιχης υπηρεσίας. Η *Service* σχετίζεται με τρεις άλλες:

- *Κλάση ServiceProfile*. Η οποία αντιπροσωπεύει το τι κάνει μια υπηρεσία. Έτσι δίνεται η δυνατότητα σε έναν έξυπνο διαμεσολαβητή να ανακαλύψει πληροφορίες σχετικά με το τι κάνει η υπηρεσία με ακρίβεια και αυτοματοποιημένα. Στην ανώτερη οντολογία η κλάση *Service* συνδέεται με την κλάση *ServiceProfile* με τη σχέση *presents*.
- *Κλάση ServiceModel*. Η οποία αντιπροσωπεύει το πως μπορεί μια υπηρεσία να χρησιμοποιηθεί και το διεργασιακό μοντέλο (*process model*) που ακολουθείται λεπτομερώς. Δίνεται έτσι η δυνατότητα σε ένα έξυπνο διαμεσολαβητή να αποκτήσει πληροφορίες σχετικά με την είσοδο και την έξοδο που απαιτεί η υπηρεσία, τις προϋποθέσεις και τα αποτελέσματα της χρήσης της. Είναι δυνατή επίσης η παροχή λεπτομερειών σχετικά με τα βήματα που ακολουθούνται ώστε η υπηρεσία να παράγει τα συγκεκριμένα αποτελέσματα. Έτσι, ένας έξυπνος διαμεσολαβητής μπορεί να γνωρίζει με μεγαλύτερη ακρίβεια εάν η υπηρεσία καλύπτει συγκεκριμένες ανάγκες, εάν μπορεί να συνδυαστεί η χρήση της και με τη χρήση άλλων υπηρεσιών και για να παρακολουθήσει την εξέλιξη μιας υπηρεσίας κατά τη χρήση της. Η κλάση *Service* συνδέεται με την κλάση *ServiceModel* με τη σχέση *describedBy*.
- *Κλάση ServiceGrounding*. Η οποία αντιπροσωπεύει το πως μπορεί να γίνει αλληλεπίδραση με την υπηρεσία ιστού. Ένας έξυπνος διαμεσολαβητής θα είναι σε θέση να έχει περιγραφή του πρωτοκόλλου επικοινωνίας που πρέπει να χρησιμοποιηθεί και του φορμά των μηνυμάτων που πρέπει να ανταλλαχθούν για τη χρήση της υπηρεσίας. Επίσης, για κάθε στοιχείο εισόδου ή εξόδου που ορίζεται στην κλάση *ServiceModel* ορίζεται η τεχνική σειριοποίησης (*serialization*) που πρέπει να ακολουθηθεί. Η κλάση *Service* συνδέεται με την κλάση *ServiceGrounding* με τη σχέση *supports*.



Σχήμα 7 Η ανώτερη οντολογία της OWL-S (βασισμένη στην τεκμηρίωση της OWL-S 1.1 – <http://www.w3.org/Submission/OWL-S/>)

² OWL-S Έκδοση 1.1

Η ανώτερη οντολογία για σημασιολογικές υπηρεσίες ιστού της OWL-S απεικονίζεται στο Σχήμα 7. Σύμφωνα με την OWL-S για ένα στιγμιότυπο Service υπάρχει το πολύ ένα στιγμιότυπο ServiceModel με το οποίο να σχετίζεται μέσω της describedBy, ενώ κάθε στιγμιότυπο ServiceGrounding συνδέεται μέσω της σχέσης supports με ακριβώς ένα στιγμιότυπο της Service. Είναι επιθυμητό για κάθε στιγμιότυπο της Service να υπάρχουν σχετιζόμενα στιγμιότυπα των ServiceProfile και ServiceModel αλλά όχι υποχρεωτικό σύμφωνα με την OWL-S 1.1.

Όπως αναφέρεται στις προηγούμενες παραγράφους, μια υπηρεσία ιστού μπορεί να περιγραφεί από δύο διαφορετικές οπτικές γωνίες: ως τρόπος μετασχηματισμού δεδομένων Εισόδου σε δεδομένα Εξόδου ή ως τρόπος μετασχηματισμού καταστάσεων από Προϋποθέσεις σε Αποτελέσματα. Είναι φυσικό αυτοί οι μετασχηματισμοί να περιγράφονται αναλυτικά στη σημασιολογική περιγραφή μιας υπηρεσίας ιστού ώστε ένας έξυπνος διαμεσολαβητής να μπορεί αφενός να διαγνώσει εάν μια υπηρεσία εξυπηρετεί συγκεκριμένες ανάγκες και αφετέρου να γνωρίζει τα ακολουθούμενα βήματα και τα δεδομένα Εισόδου όπως και τις Προϋποθέσεις για να τη χρησιμοποιήσει.

Επακόλουθα, θα ήταν αναμενόμενο οι Είσοδος, Έξοδος, Προϋποθέσεις και Αποτελέσματα (ΕΕΠΑ) μιας υπηρεσίας να αποτελούν αντικείμενο τόσο της περιγραφής που περιέχει η κλάση ServiceProfile προκειμένου μπορεί ένας διαμεσολαβητής να ανακαλύψει την υπηρεσία, όσο και της περιγραφής που περιέχει η κλάση ServiceModel ώστε να είναι σε θέση ένας διαμεσολαβητής να χρησιμοποιήσει την υπηρεσία. Ταυτόχρονα, θα περίμενε ίσως κανείς ότι η περιγραφή ΕΕΠΑ σε επίπεδο ServiceProfile θα είναι ένα υποσύνολο της περιγραφής σε επίπεδο ServiceModel αφού η τελευταία θα πρέπει να είναι αρκετά πιο λεπτομερής προκειμένου να γίνεται σωστή χρήση της υπηρεσίας από έξυπνους διαμεσολαβητές.

Γίνεται επίσης αντιληπτός ο κίνδυνος να υπάρχουν ασυμφωνίες μεταξύ των περιγραφών ΕΕΠΑ της ServiceProfile και της ServiceModel. Η OWL-S 1.1 όμως δεν επιβάλλει κάποιους κανόνες ή περιορισμούς για αυτό αφήνοντας αποκλειστικά την ευθύνη σε αυτόν που δημιουργεί την περιγραφή μιας σημασιολογικής υπηρεσίας ιστού.

Υπάρχουν αρκετά παραδείγματα σημασιολογικών περιγραφών ακόμα και για υπάρχουσες και ευρέως χρησιμοποιούμενες υπηρεσίες ιστού. Μια σειρά από τέτοια παραδείγματα είναι διαθέσιμα στις ιστοσελίδες του προγράμματος MINDSWAP (<http://www.mindswap.org/>). Ας δούμε όμως ένα παράδειγμα σημασιολογικής περιγραφής μιας υπηρεσίας ιστού για χρηματιστηριακές συναλλαγές. Ένα πρώτο μέρος της περιγραφής μιας ΣΥΙ σύμφωνα με OWL-S περιλαμβάνει συντομεύσεις, ονοματοχώρους και εισαγωγή στοιχείων που αφορούν την ανώτερη οντολογία που ορίζει η OWL-S. Σύμφωνα με το στυλ γραφής υπηρεσιών ιστού του προγράμματος MINDSWAP μπορούν να γίνουν οι εξής δηλώσεις:

```
<?xml version="1.0" encoding="UTF-8"?>
<!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 service "http://www.daml.org/services/owl-s/1.1/Service.owl">
  <!ENTITY profile "http://www.daml.org/services/owl-s/1.1/Profile.owl">
  <!ENTITY process "http://www.daml.org/services/owl-s/1.1/Process.owl">
  <!ENTITY grounding "http://www.daml.org/services/owl-s/1.1/Grounding.owl">
  <!ENTITY groundingWSDL "http://stockinfo.example.com/stockupdate.wsdl">
  <!ENTITY concepts "http://stockinfo.example.com/concepts.owl">
  <!ENTITY profileH "http://www.daml.org/services/owl-s/1.1/ProfileHierarchy.owl">
]>

<rdf:RDF
```

```

xmlns:rdf="&rdf;#"
xmlns:rdfs="&rdfs;#"
xmlns:owl="&owl;#"
xmlns:xsd="&xsd;#"
xmlns:service="&service;#"
xmlns:profile="&profile;#"
xmlns:process="&process;#"
xmlns:grounding="&grounding;#"
xml:base="http://stockinfo.example.com/stockupdate.owl
>

<owl:Ontology rdf:about="">
  <owl:imports rdf:resource="&service;"/>
  <owl:imports rdf:resource="&profile;"/>
  <owl:imports rdf:resource="&process;"/>
  <owl:imports rdf:resource="&grounding;"/>
  <owl:imports rdf:resource="&concepts;"/>
</owl:Ontology>

```

Σε αυτό το παράδειγμα έχουμε τα τέσσερα διαφορετικά μέρη μιας σημασιολογικής περιγραφής σε OWL-S 1.1 η οποία είναι όπως και η OWL βασισμένη στην RDF. Έτσι μετά τις γενικές δηλώσεις έχουμε την περιγραφή των τεσσάρων τμημάτων: Service, Profile, Process (Service) Model, Grounding τα οποία περιγράφουν το περιεχόμενο των αντίστοιχων τεσσάρων κλάσεων στην ανώτερη οντολογία που ορίζει η OWL-S 1.1:

```

<!-- Service description -->
<service:Service rdf:ID="StockUpdateService">
  <service:presents rdf:resource="#StockUpdateProfile"/>

  <service:describedBy rdf:resource="#StockUpdateProcess"/>

  <service:supports rdf:resource="#StockUpdateGrounding"/>
</service:Service>

```

- Το τμήμα περιγραφής της *Υπηρεσίας* (Service) δίνει ένα όνομα στην υπηρεσία και τη συσχετίζει με τα αντίστοιχες περιγραφές *Προφίλ* υπηρεσίας (Profile), *Διεργασιακό Μοντέλο* (Process Model) και *Συσχετισμού* με συγκεκριμένο πρωτόκολλο (Grounding).

```

<!-- Profile description -->
<profileH:StockInformation rdf:ID="StockUpdateProfile">
  <service:presentedBy rdf:resource="#StockUpdateService"/>

  <profile:serviceName xml:lang="en">Stock Update</profile:serviceName>
  <profile:textDescription xml:lang="en">This service returns stock quotes.
</profile:textDescription>

  <profile:hasInput rdf:resource="#StockInfo"/>

  <profile:hasOutput rdf:resource="#StockPrice"/>
</profileH:StockInformation>

```

- Το τμήμα περιγραφής του *Προφίλ* υπηρεσίας εκτός από μια συνοπτική περιγραφή σε κείμενο του τι κάνει η υπηρεσία δίνει και μια συνοπτική εικόνα του μετασχηματισμού δεδομένων που επιτυγχάνει με τον ορισμό των σχέσεων *hasInput* και *hasOutput*. Το προφίλ μιας υπηρεσίας μπορεί να περιλαμβάνει αρκετά ακόμα στοιχεία όπως πληροφορίες επικοινωνίας με τον φορέα που την προσφέρει ή και πληροφορίες μετασχηματισμού καταστάσεων με τη χρήση των σχέσεων *hasPrecondition* και *hasResult*. Επίσης το προφίλ υπηρεσίας μπορεί να δώσει μια κατηγοριοποίηση της υπηρεσίας (για παράδειγμα σε σχέση με μια OWL οντολογία κατά NAICS – North American Industry Classification System, <http://www.census.gov/naics/>) ή του προϊόντος

που μπορεί να προσφέρει (όπως για παράδειγμα σε σχέση με μια OWL οντολογία κατα UNSPSC – United Nations Standard Products and Services Code, <http://www.unspsc.org/>). Το προφίλ υπηρεσίας μπορεί να αποτελεί μέρος μιας ιεραρχίας προφίλ υπηρεσιών που τηρεί ένας πάροχος.

```
<!-- Process Model description -->
<process:AtomicProcess rdf:ID="StockUpdateProcess">
  <service:describes rdf:resource="#StockUpdateService"/>

  <process:hasInput rdf:resource="#StockInfo"/>

  <process:hasOutput rdf:resource="#StockPrice"/>
</process:AtomicProcess>

<process:Input rdf:ID="StockInfo">
  <process:parameterType rdf:datatype="xsd:anyURI">& concepts; #StockUpdateInput
</process:parameterType>
  <rdfs:label>Stock Information</rdfs:label>
</process:Input>

<process:Output rdf:ID="StockPrice">
  <process:parameterType rdf:datatype="xsd:anyURI">& concepts; #StockUpdateOutput
</process:parameterType>
  <rdfs:label>Stock Price</rdfs:label>
</process:Output>
```

- Η περιγραφή του *Διεργασιακού Μοντέλου* (Process Model) δίνει μια πιο λεπτομερή περιγραφή των μετασχηματισμών που επιτυγχάνει η υπηρεσία σε δεδομένα ή σε καταστάσεις (στο συγκεκριμένο παράδειγμα σε δεδομένα) περιλαμβάνοντας τους τύπους των δεδομένων που μετασχηματίζονται και την περιγραφή τους. Το πιο σημαντικό όμως χαρακτηριστικό της περιγραφής του διεργασιακού μοντέλου είναι η δυνατότητα ορισμού ατομικών (atomic), απλών (simple) και σύνθετων (composite) διεργασιών μετασχηματισμού δεδομένων ή καταστάσεων.
 - Οι *Ατομικές* διεργασίες μπορούν να κληθούν όπως ορίζεται σε αντίστοιχους συσχετισμούς με πρωτόκολλα επικοινωνίας (grounding), δεν περιλαμβάνουν υπο-διεργασίες και εκτελούνται σε ένα βήμα (single step).
 - Οι *Απλές* διεργασίες εκτελούνται σε ένα βήμα επίσης, όμως δεν μπορούν να κληθούν και δεν έχουν αντίστοιχους συσχετισμούς με πρωτόκολλα επικοινωνίας αφού αφορούν ένα αφαιρετικό επίπεδο (abstract level) στο οποίο μπορούν να περιγραφούν διεργασίες. Έτσι, μια απλή διεργασία μπορεί να *υλοποιηθεί* (realizedBy) από μια ατομική διεργασία ή να *επεκταθεί* (expandsTo) στην υλοποίηση μιας σύνθετης διεργασίας.
 - Οι *Σύνθετες* διεργασίες αποτελούν σύνθεση άλλων ατομικών ή σύνθετων διεργασιών ενώ ο τρόπος που συντίθεται ορίζεται μέσω δομών ελέγχου (control structures) όπως if-then-else, sequence, split, choice, repeat, κλπ.

Στο παραπάνω παράδειγμα σημασιολογικής περιγραφής έχουμε τον ορισμό μιας ατομικής διεργασίας *StockUpdateProcess* που μετασχηματίζει δεδομένα του τύπου *StockInfo* σε δεδομένα του τύπου *StockPrice*.

```
<!-- Grounding description -->
<grounding:WsdgGrounding rdf:ID="StockUpdateGrounding">
  <service:supportedBy rdf:resource="#StockUpdateService"/>
```



```

    <grounding:hasAtomicProcessGrounding
rdf:resource="#StockUpdateProcessGrounding"/>
</grounding:WsdLGrounding>

<grounding:WsdLAtomicProcessGrounding rdf:ID="StockUpdateProcessGrounding">
  <grounding:owlsProcess rdf:resource="#StockUpdateProcess"/>
  <grounding:wsdlDocument
rdf:datatype="&xsd:anyURI">&groundingWSDL;</grounding:wsdlDocument>
    <grounding:wsdlOperation>
      <grounding:WsdLOperationRef>
        <grounding:portType
rdf:datatype="&xsd:anyURI">&groundingWSDL;#StockUpdatePortType</grounding:portType>
          <grounding:operation
rdf:datatype="&xsd:anyURI">&groundingWSDL;#StockUpdate</grounding:operation>
            </grounding:WsdLOperationRef>
          </grounding:wsdlOperation>

      <grounding:wsdlInputMessage
rdf:datatype="&xsd:anyURI">&groundingWSDL;#StockUpdateInput</grounding:wsdlInputMessage
>
        <grounding:wsdlInput>
          <grounding:WsdLInputMessageMap>
            <grounding:owlsParameter rdf:resource="#StockInfo"/>
            <grounding:wsdlMessagePart
rdf:datatype="&xsd:anyURI">&groundingWSDL;#stockrequest</grounding:wsdlMessagePart>
              </grounding:WsdLInputMessageMap>
            </grounding:wsdlInput>

          <grounding:wsdlOutputMessage
rdf:datatype="&xsd:anyURI">&groundingWSDL;#StockUpdateOutput</grounding:wsdlOutputMessa
ge>
            <grounding:wsdlOutput>
              <grounding:WsdLOutputMessageMap>
                <grounding:owlsParameter rdf:resource="#StockPrice"/>
                <grounding:wsdlMessagePart
rdf:datatype="&xsd:anyURI">&groundingWSDL;#stockresponse</grounding:wsdlMessagePart>
                  </grounding:WsdLOutputMessageMap>
                </grounding:wsdlOutput>
              </grounding:wsdlOutputMessage>
            </grounding:WsdLAtomicProcessGrounding>
  </rdf:RDF>

```

- Η περιγραφή του *Συσχετισμού* (grounding) με πρωτόκολλα επικοινωνίας είναι παρόμοια με αυτή του συσχετισμού τύπων πόρτας υπηρεσιών με πρωτόκολλα επικοινωνίας στην WSDL (binding) όπως περιγράφεται στην ενότητα 1.4. Με αυτό τον τρόπο είναι δυνατή η ανάπτυξη πελατών για τις υπηρεσίες που προσφέρονται από τον προγραμματιστή ή από έναν έξυπνο διαμεσολαβητή. Στην ουσία προκειμένου να χρησιμοποιηθεί μια υπηρεσία απαιτείται τόσο η περιγραφή του Συσχετισμού (grounding) σε OWL-S όσο και η περιγραφή σε WSDL. Για την υπηρεσία του παραδείγματος η περιγραφή σε WSDL θα μπορούσε να είναι αυτή που παρατίθεται στην ενότητα 1.4. Στο συγκεκριμένο παράδειγμα βλέπουμε ότι ο συσχετισμός της ατομικής διεργασίας (StockUpdateProcess) που ορίζεται στο διεργασιακό μοντέλο γίνεται με τον ορισμό σε WSDL της υπηρεσίας *StockUpdateWebService*. Έτσι, γίνεται συσχετισμός με τον τύπο πόρτας υπηρεσιών *StockUpdatePortType* και με τη λειτουργία *StockUpdate* ενώ οι τύποι παραμέτρων εισόδου *StockInfo* και εξόδου *StockPrice* συσχετίζονται με τους τύπους δεδομένων που ορίζονται στην WSDL περιγραφή *stockrequest* και *stockresponse* αντίστοιχα.

Η OWL-S είναι ένας τρόπος σημασιολογικής περιγραφής υπηρεσιών ιστού ο οποίος έχει τύχει προσοχής από την ερευνητική κοινότητα τα τελευταία χρόνια. Υπάρχουν όμως και άλλες προτάσεις σχετικά με το πως μπορεί να γίνει η σημασιολογική περιγραφή υπηρεσιών ιστού οι οποίες αναλύονται στην επόμενη ενότητα.

Ταυτόχρονα, ορισμένα χαρακτηριστικά της OWL-S αποτελούν αντικείμενο κριτικής. Για παράδειγμα, πολλοί ερευνητές αμφιβάλλουν το κατά πόσο μπορεί να γίνεται αποδοτικά η ανακάλυψη υπηρεσιών με βάση OWL-S περιγραφές. Πιο συγκεκριμένα, το *ταίριασμα* (matchmaking) μεταξύ επιθυμητών υπηρεσιών σε OWL-S (όπου οι

υπηρεσίες μπορούν να περιγραφούν ως μετασχηματισμοί ΕΕΠΑ) με υπάρχουσες υπηρεσίες απαιτεί να γίνουν πολλοί έλεγχοι ως προς το κατά πόσο ταιριάζουν οι επιθυμητές είσοδοι-έξοδοι και προϋποθέσεις-αποτελέσματα με αυτά της κάθε προσφερόμενης υπηρεσίας.

Από την άλλη μεριά η OWL-S λαμβάνει υπόψη τα υπάρχοντα πρότυπα WSDL ενώ μπορεί να χρησιμοποιηθεί και σε συνδυασμό με το UDDI για ταίριασμα δυνατοτήτων υπηρεσιών ιστού (McIlraith, Martin, 2003).

2.3 Περιγραφή σημασιολογικών υπηρεσιών ιστού με WSMO

Μια εναλλακτική πρόταση για τη σημασιολογική περιγραφή υπηρεσιών ιστού αποτελεί αυτή της Web Service Modelling Ontology (WSMO), η οποία έχει προταθεί στον οργανισμό W3C (<http://www.w3.org/Submission/WSMO>) στα πλαίσια της δραστηριότητας τυποποίησης που έχει αναλάβει. Η WSMO είναι βασικό μέρος του πλαισίου Web Services Modelling Framework (WSMF) (Fensel, Bussler, 2002) μέρος του οποίου αποτελεί τόσο η WSMO που περιγράφει ένα πλαίσιο για την ανάπτυξη οντολογιών για υπηρεσίες ιστού όσο και η γλώσσα περιγραφής τους, η WSML (Web Service Modelling Language, <http://www.wsmo.org/wsml/>).

Σε υψηλό επίπεδο, για την περιγραφή σημασιολογικών υπηρεσιών ιστού η WSMO προτείνει τέσσερις βασικές έννοιες:

- *Οντολογίες (ontologies)*. Οι οποίες προσφέρουν την ορολογία που μπορεί να χρησιμοποιείται από άλλα στοιχεία της WSMO για τον ορισμό εννοιών (concepts) και σχέσεων (relations). Οι έννοιες και οι σχέσεις μπορούν να περιγράψουν ένα συγκεκριμένο πεδίο που αφορά κοινότητες χρηστών και δραστηριότητες. Π.χ. τις έννοιες και τις σχέσεις που έχουν να κάνουν με δραστηριότητες στη χρηματιστηριακή αγορά. Για την περιγραφή των ιδίων των οντολογιών χρησιμοποιείται ως βάση η ορολογία DublinCore (IETF RFC 2413, <http://www.ietf.org/rfc/rfc2413.txt?number=2413>) ενώ προβλέπεται η περιγραφή και μη-λειτουργικών (non-functional) ιδιοτήτων των οντολογιών.
- *Υπηρεσίες Ιστού (Web Services)*. Οι οποίες περιγράφουν την υπηρεσίες που προσφέρονται ως υλοποίηση. Έτσι, περιγράφεται οι ικανότητες, οι διεπαφές και η συμπεριφορά υπηρεσιών ιστού.
- *Στόχοι (goals)*. Οι οποίοι περιγράφουν τις επιθυμίες των χρηστών τις οποίες έρχονται να καλύψουν μεμονωμένες ή συνδυασμένες υπηρεσίες ιστού. Με αυτό τον τρόπο αναπαρίσταται η οπτική γωνία των χρηστών όσον αφορά τη χρήση υπηρεσιών ιστού.
- *Μεσάζοντες (mediators)*. Οι οποίοι αποτελούν βασικά στοιχεία προκειμένου να αντιμετωπισθούν θέματα σχετικά με την διαλειτουργικότητα διεργασιών, πρωτοκόλλων ή οντολογιών και να υποστηριχθούν συνδυασμοί υπηρεσιών ιστού.

Η WSMO διαφοροποιείται από την OWL-S σε αρκετά σημεία (Polleres, Lara, 2005). Μία διαφορά της πρώτης από τη δεύτερη είναι ότι υπάρχει σαφής διαχωρισμός στον τρόπο περιγραφής των υπηρεσιών που επιθυμεί ο χρήστης από τον τρόπο περιγραφής των υπηρεσιών που είναι διαθέσιμες. Αυτό επιτυγχάνεται με τον ορισμό στόχων από μεριάς του χρήστη το οποίο υποστηρίζεται ότι μπορεί να επιτύχει πιο αποτελεσματικό ταίριασμα με σχετικές υπηρεσίες ιστού.

Μία άλλη διαφορά είναι ότι η WSMO προβλέπει τυποποιημένο τρόπο περιγραφής των μη-λειτουργικών ιδιοτήτων υπηρεσιών ιστού βασισμένο κυρίως στην DublinCore ή στην ορολογία FOAF (<http://xmlns.com/foaf/0.1/>) του προγράμματος FOAF (Friend Of A

Friend, <http://www.foaf-project.org/>). Στην OWL-S η περιγραφή μη-λειτουργικών ιδιοτήτων περιορίζεται στην κλάση Profile.

Μία σημαντική διαφορά επίσης είναι ότι η WSMO προβλέπει όσον αφορά την περιγραφή υπηρεσιών ιστού ορισμό *χορογραφιών* (choreography) που μπορεί να εκτελέσει η υπηρεσία ιστού και *συνθέσεων* (orchestration). Με τις χορογραφίες μπορεί να περιγραφεί με ακρίβεια η συμπεριφορά μιας υπηρεσίας ιστού μεμονωμένα όπως μπορεί να παρατηρηθεί από κάποιο εξωτερικό παρατηρητή. Με τις συνθέσεις περιγράφονται τρόποι με τους οποίους μια συγκεκριμένη υπηρεσία μπορεί να συνδυαστεί με άλλες από ένα κεντρικό σημείο ελέγχου (προγραμματιστή ή έξυπνο διαμεσολαβητή) για την παροχή σύνθετων υπηρεσιών ιστού.

2.4 Άλλες προτάσεις για περιγραφή σημασιολογικών υπηρεσιών ιστού

Εκτός από την OWL-S και την WSMO έχουν προταθεί και άλλα πλαίσια για τη σημασιολογική περιγραφή υπηρεσιών ιστού.

Μια από αυτές της προτάσεις αποτελεί η WSDL-S (Web Service Semantics) η προέρχεται από το ερευνητικό πρόγραμμα METEOR-S και έχει προταθεί στον οργανισμό W3C για τυποποίηση (<http://www.w3.org/Sumbission/WSDL-S>, Sivashanmugam et al, 2005, Patil et al, 2004). Η φιλοσοφία στην οποία στηρίζεται η WSDL-S είναι ότι αν και υπάρχουν διαφορετικές οντολογίες και γλώσσες που προσφέρουν τρόπους σημασιολογικής περιγραφής υπηρεσιών ιστού αυτό δεν πρέπει να συνεπάγεται ότι μία από αυτές θα πρέπει να επιλεγεί από έναν οργανισμό τυποποίησης και να προωθηθεί για χρήση από όλους. Είναι θεμιτό να υπάρχει η ελευθερία χρήσης κάποιας προτιμώμενης γλώσσας κατά περίπτωση. Για αυτό το λόγο προτείνεται ένας τρόπος πρόσθεσης σημασιολογικών περιγραφών υπηρεσιών ιστού χρησιμοποιώντας τις δομές της υπάρχουσας γλώσσας περιγραφής υπηρεσιών ιστού, της WSDL.

Προτείνεται λοιπόν η χρήση επιπρόσθετων στοιχείων σε περιγραφές WSDL τα οποία μπορούν σε κάποια επιλεγμένη γλώσσα οντολογίας να περιγράψουν την κατηγορία στην οποία ανήκει η υπηρεσία ιστού και να προσφέρουν σημασιολογική περιγραφή των δεδομένων εισόδου και εξόδου, προϋποθέσεων και αποτελεσμάτων.

Έτσι δεν απαιτείται η εισαγωγή νέων προτύπων για τη σημασιολογική περιγραφή υπηρεσιών ιστού. Τα υπάρχοντα πρότυπα όπως η WSDL αλλά και το UDDI μπορούν να συμπεριλάβουν σημασιολογικές περιγραφές των υπηρεσιών που πραγματεύονται. Στα πλαίσια του προγράμματος METEOR-S έχουν προταθεί αλγόριθμοι που επιτρέπουν ανακάλυψη και το συνδυασμό υπηρεσιών με βάση την σημασιολογική περιγραφή τους (Sivashanmugam et al).

Μία άλλη πρόταση είναι αυτή για το πλαίσιο SWSF (Semantic Web Services Framework) όπως έχει υποβληθεί στον οργανισμό W3C για τυποποίηση (<http://www.w3.org/Submission/SWSF>). Το πλαίσιο αυτό περιλαμβάνει τη γλώσσα σημασιολογικής περιγραφής υπηρεσιών ιστού SWSL (Semantic Web Services Language) και την οντολογία SWSO (Semantic Web Services Ontology).

Η γλώσσα SWSL αποτελείται από δύο υπο-γλώσσες, την SWSL-FOL (First-Order Logic) και την SWSL-Rules. Η πρώτη είναι βασισμένη σε λογική πρώτου βαθμού και μπορεί να περιγράψει την οντολογία που αφορά υπηρεσίες ιστού. Η δεύτερη μπορεί να υποστηρίξει Αιτιολόγηση (reasoning) με βάση τη σημασιολογική περιγραφή υπηρεσιών. Η SWSL είναι μία γενικού σκοπού γλώσσα η οποία όμως έχει σχεδιαστεί λαμβάνοντας υπόψη τις ανάγκες για την περιγραφή σημασιολογικών υπηρεσιών ιστού χρησιμοποιώντας URI, ονοματοχώρους (namespaces) και τύπους δεδομένων σε XML. Η οντολογία SWSO προσφέρει το μοντέλο για την περιγραφή υπηρεσιών ιστού και περιλαμβάνει τον ορισμό αξιωμάτων σε λογική πρώτου βαθμού.

2.5 Σύνθεση Σημασιολογικών Υπηρεσιών Ιστού

Στην ενότητα 1.2 έγινε αναφορά στους διαφορετικούς τρόπους με τους οποίους είναι δυνατός ο συνδυασμός υπηρεσιών ιστού. Με τέτοιους τρόπους μπορούν να συντίθενται και να προσφέρονται σύνθετες υπηρεσίες, οι οποίες βασισμένες στο συνδυασμό των κατάλληλων υπηρεσιών μπορούν να υποστηρίξουν εξατομίκευση.

Έχει επίσης αναφερθεί ότι η ανάγκη βελτίωσης της αποτελεσματικότητας της ανακάλυψης, χρήσης και συνδυασμού υπηρεσιών ιστού είναι ένας από τους βασικούς λόγους έρευνας στην περιοχή των σημασιολογικών υπηρεσιών ιστού.

Έχοντας παρουσιάσει τους τρόπους με τους οποίους είναι αυτή τη στιγμή δυνατή η πρόσθεση σημασιολογικού περιεχομένου στις υπηρεσίες ιστού, στις επόμενες παραγράφους θα περιγράψουμε τους τρόπους με τους οποίους η σημασιολογική αυτή περιγραφή αξιοποιείται για τη *σύνθεση* υπηρεσιών ιστού. Η χρήση της λέξης «σύνθεση» σε αυτό το σημείο είναι σκόπιμο να πούμε ότι χρησιμοποιείται από την ερευνητική κοινότητα μάλλον με το νόημα της «ενορχήστρωσης» παρά της «χορογραφίας» που παρουσιάστηκε στην ενότητα 1.2 και έπειτα αναφέρθηκε ως χαρακτηριστικό της WSMO στην ενότητα 2.3. Η σύνθεση υπηρεσιών ιστού με βάση τις σημασιολογικές περιγραφές τους έχει προσελκύσει σημαντική ερευνητική προσπάθεια τα τελευταία χρόνια.

Φαίνεται ότι μια σημαντική προσπάθεια έχει επενδυθεί στη χρήση του *Σχεδιασμού Ενεργειών Τεχνητής Νοημοσύνης* (AI Planning) (Hendler et al, 1990) για *αυτοματοποιημένη σύνθεση* υπηρεσιών ιστού αποκλειστικά από έξυπνους διαμεσολαβητές ή για *ημι-αυτοματοποιημένη σύνθεση* η οποία απαιτεί κάποια ενέργειες και από χρήστες σε συνεργασία με έξυπνους διαμεσολαβητές.

Ο Σχεδιασμός Ενεργειών Τεχνητής Νοημοσύνης (Σχεδιασμός Ενεργειών TN) στο συγκεκριμένο τομέα των υπηρεσιών ιστού βασίζεται ακριβώς στον ορισμό μιας υπηρεσίας ως διαδικασία μετασχηματισμού δεδομένων (από δεδομένα εισόδου σε δεδομένα εξόδου) αλλά και καταστάσεων (από προϋποθέσεις σε αποτελέσματα).

Για παράδειγμα, μπορεί να εκφράσει κανείς με ακρίβεια την κατάσταση από την οποία θέλει να ξεκινήσει και την κατάσταση στην οποία θέλει να φθάσει και με τη χρήση τεχνικών Σχεδιασμού Ενεργειών TN ένα πρόγραμμα Σχεδιαστή Ενεργειών (AI Planner) μπορεί να ανακαλύψει τους κατάλληλους συνδυασμούς υπηρεσιών που μπορούν να υποστηρίξουν (πιθανότατα συνεργαζόμενες μεταξύ τους) τις απαιτήσεις για υπηρεσίες, να επιλέξει συνδυασμό, να αποφασίσει για τα δεδομένα που πρέπει να ανταλλάξουν οι υπηρεσίες ιστού για αυτό το σκοπό όπως και τα πρωτόκολλα που πρέπει να χρησιμοποιηθούν.

Μια ερευνητική κατεύθυνση για συνδυασμό υπηρεσιών ιστού χρησιμοποιώντας τεχνικές Σχεδιασμού Ενεργειών TN είναι αυτή που στηρίζεται στη γλώσσα ConGogol (McIlraith, Son, 2002). Η ConGogol βασίζεται στη γλώσσα λογικού προγραμματισμού (logic programming) με προγραμματισμό καταστάσεων (situation calculus). Οι επεκτάσεις που φέρνει η ConGolog εστιάζονται στη δυνατότητα περιγραφής των προτιμήσεων και περιορισμών που επιβάλλει ο χρήστης του σχεδιασμού όπως και με την περιγραφή απαιτήσεων για επιπλέον βήματα σχεδιασμού ενεργειών που μπορεί να απαιτηθούν για παράδειγμα εάν ένα επιλεγμένο σχέδιο συνδυασμού ενεργειών παρουσιάσει προβλήματα κατά την εκτέλεσή του.

Με βάση τις προϋποθέσεις και την επιθυμητή κατάσταση ένας σχεδιαστής ενεργειών βασισμένος στην ConGogol μπορεί να επιλέξει τις κατάλληλες ενέργειες και τη σειρά με την οποία θα πρέπει να εκτελεστούν προκειμένου να φθάσουμε στα επιθυμητά αποτελέσματα. Κάθε τέτοια ενέργεια μπορεί να είναι *απλή* ή *σύνθετη*. Οι απλές

ενέργειες μπορούν να εκτελούνται από κάποια υπηρεσία ιστού και είτε να αποτελούν ενέργειες συλλογής περαιτέρω πληροφορίας είτε ενέργειες καταστάσεων που μπορεί να μας φέρουν πιο κοντά στο επιθυμητό αποτέλεσμα του σχεδιασμού.

Εκτός από την ConGogol ενδιαφέρον επικεντρώνει και ο τρόπος σχεδιασμού ενεργειών γνωστός ως Hierarchical Task Network (HTN) (Erol et al, 1994). Ο HTN σχεδιασμός δεν ξεκινάει με μια αρχική κατάσταση αλλά με ένα καθήκον (task) το οποίο πρέπει να εκτελεστεί και μεθόδους που μπορούν να χρησιμοποιηθούν για τον επιμερισμό τους. Έτσι, ένα καθήκον επιμερίζεται σε υπο-καθήκοντα (sub-tasks) τα οποία με τη σειρά τους μπορούν να διαιρούνται σε περαιτέρω υπο-καθήκοντα ώσπου θα φθάσουμε σε ένα επίπεδο καθηκόντων που θα δώσουν μία ταξινομημένη λίστα ενεργειών που πρέπει να σχεδιαστούν. Ο HTN σχεδιασμός εμφανίζεται να είναι πολλά υποσχόμενος δεδομένου ότι έχει χρησιμοποιηθεί επιτυχώς για την υλοποίηση σχεδιαστών ενεργειών (AI Planners) στην περίπτωση του σχεδιαστή SHOP (<http://www.cs.umd.edu/projects/shop>) (Nau et al, 2003) ο οποίος μάλιστα κέρδισε ένα από τα 4 πρώτα βραβεία σε διαγωνισμό για σχεδιαστές ενεργειών³.

Υπάρχει ένας σημαντικός ακόμα αριθμός προτάσεων για το πως μπορεί να γίνει σύνθεση υπηρεσιών ιστού χρησιμοποιώντας σημασιολογικές περιγραφές. Για παράδειγμα η εργασία των Medjahed et al (2003), όπου χρησιμοποιείται σχεδιασμός ενεργειών με χρήση κανόνων (rule-based planning) και κάποιες υλοποιήσεις (Ponnekanti, Fox, 2002).

Άλλες μέθοδοι κάνουν χρήση τεχνολογιών αναπαράστασης και διαχείρισης γνώσης προκειμένου να επιτύχουν έναν δυναμικό σχεδιασμό Ροής Εργασιών (workflow) και να υλοποιήσουν σύνθεση υπηρεσιών ιστού όπως το PPM (Polymorphic Process Model) (Laukkanen, Helin, 2003) έχουν επίσης προταθεί.

Υπάρχει ακόμα πληθώρα προτάσεων για τη σύνθεση σημασιολογικών υπηρεσιών ιστού. Ταυτόχρονα όμως αρκετές προκλήσεις παραμένουν ανοιχτές όσον αφορά τη σύνθεση υπηρεσιών ιστού. Οι προκλήσεις αυτές έχουν να κάνουν με την κλίμακα υλοποίησης, με την εύρεση της κατάλληλης ισορροπίας μεταξύ διαθέσιμης υπολογιστικής δύναμης για τη σύνθεση υπηρεσιών και τον βαθμό αυτοματοποίησης της διαδικασίας αυτής, με θέματα ασφάλειας, με θέματα τυποποίησης σημασιολογικής περιγραφής υπηρεσιών, θέματα συνυπολογισμού πολιτικών (policy) όσον αφορά τη χρήση υπηρεσιών ιστού από τους παρόχους τους, θέματα αξιοπιστίας υπηρεσιών, σχεδιασμού πλαισίων διαπραγμάτευσης για την παροχή υπηρεσιών ιστού, κλπ.

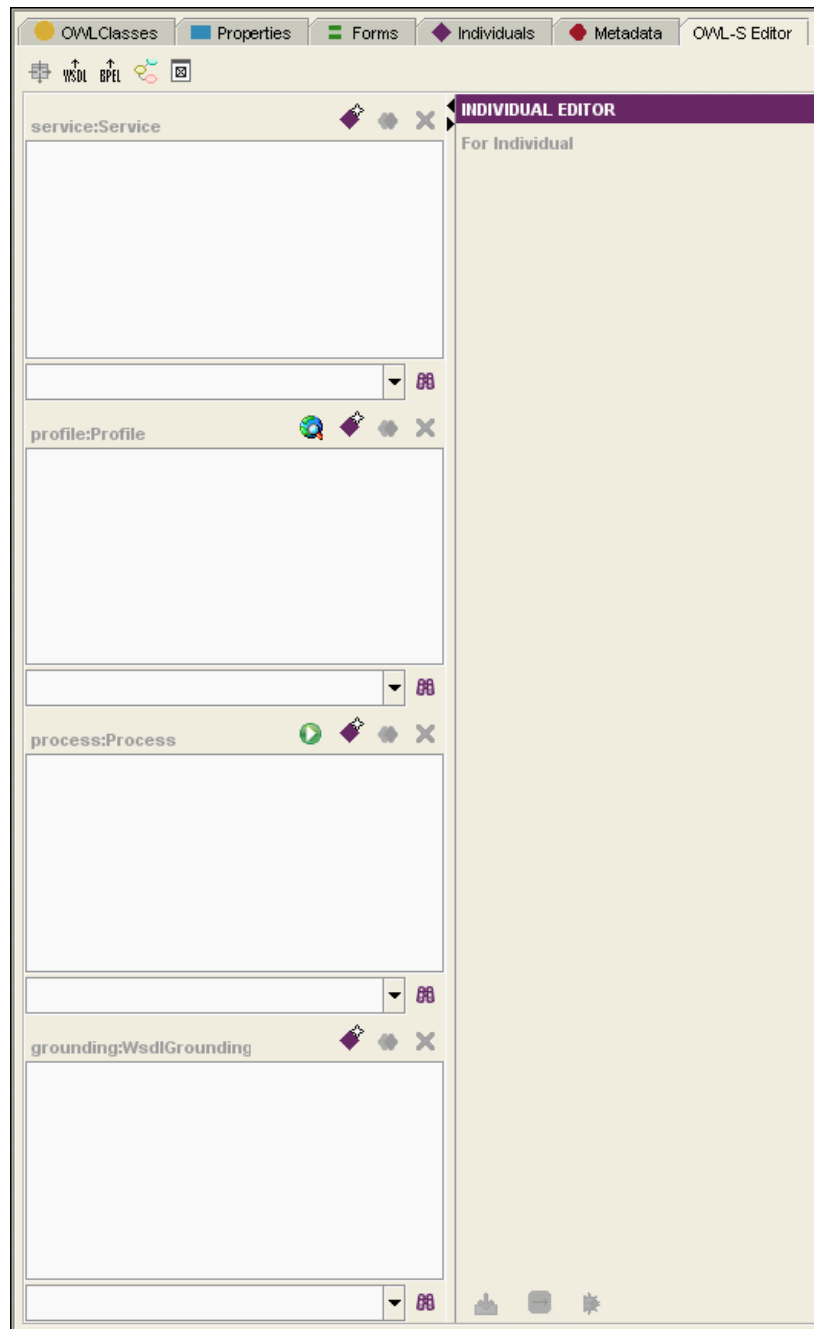
3 Εφαρμογές για Σημασιολογικές Υπηρεσίες Ιστού

Υπάρχει ήδη ένας σημαντικός αριθμός εφαρμογών που σχετίζονται με τη δημιουργία, υποστήριξη ή προσφορά σημασιολογικών υπηρεσιών ιστού μερικές από τις οποίες παρουσιάζονται στις επόμενες παραγράφους. Οι πιο πολλές από αυτές είναι ανοιχτού πηγαίου κώδικα (open source).

3.1 Εργαλεία χαρακτηρισμού σημασιολογικών υπηρεσιών ιστού

Υπάρχει ένας αριθμός εφαρμογών για το χαρακτηρισμό υπηρεσιών ιστού ώστε να αποκτήσουν σημασιολογική περιγραφή σύμφωνα με τις διάφορες προτεινόμενες γλώσσες όπως παρουσιάζονται στην ενότητα 2.

³ Ιστοσελίδα διαγωνισμού <http://planning.cis.strath.ac.uk/competition/>.



Σχήμα 8 Ο εκδότης OWL-S (OWL-S Editor) στο εργαλείο Protégé (από την ιστοσελίδα <http://owlseditor.semwebcentral.org/documentation.shtml>).

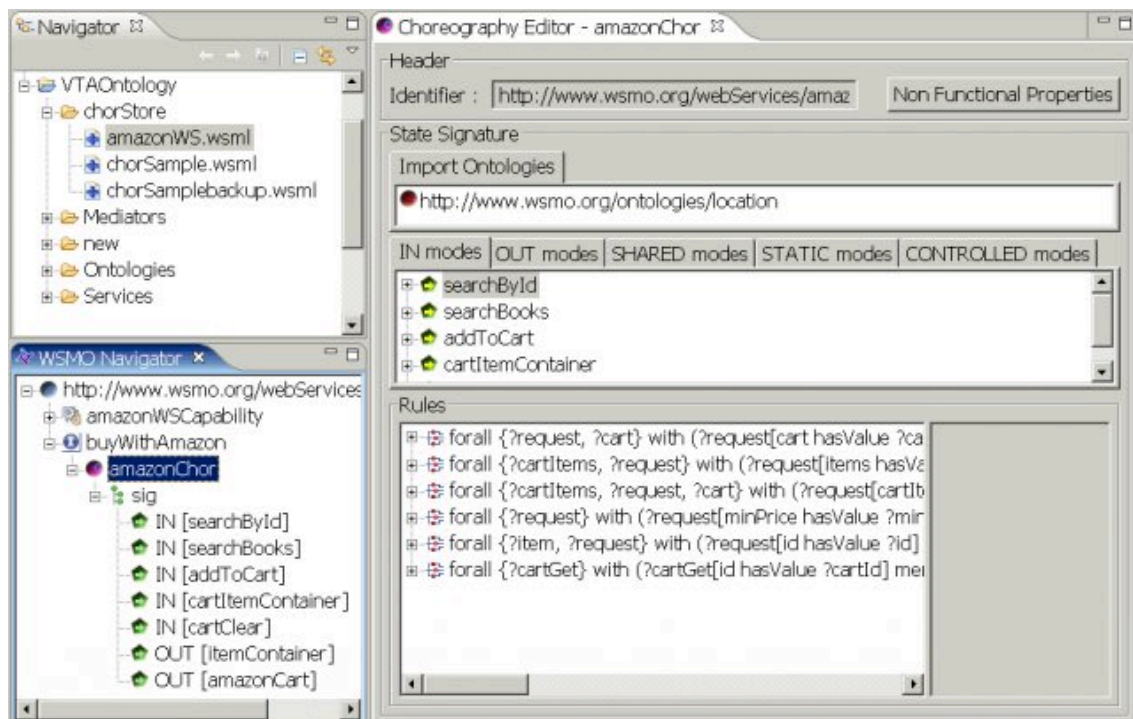
Η γλώσσα σημασιολογικής περιγραφής υπηρεσιών ιστού OWL-S που παρουσιάστηκε στην ενότητα 2.2 είναι αρκετά δημοφιλής στην ερευνητική κοινότητα. Έτσι, έχει αναπτυχθεί ένας εκδότης OWL-S (OWL-S Editor, <http://owlseditor.semwebcentral.org/>) ο οποίος μπορεί να ολοκληρωθεί με το εργαλείο Protégé (<http://protege.stanford.edu>) ως επέκτασή του. Στο Σχήμα 8 φαίνεται η απεικόνιση του OWL-S εργαλείου μέσα από το Protégé.

Ο εκδότης OWL-S έχει τη δυνατότητα επίσης να εισάγει ένα WSDL αρχείο και να δημιουργήσει τον αντίστοιχο σκελετό μιας OWL-S περιγραφής του την οποία μπορεί να χρήστης να συμπληρώσει με σημασιολογικό περιεχόμενο.

Επίσης η ομάδα που στηρίζει το WSMO (<http://www.wsmo.org>) έχει αναπτύξει έναν εκδότη WSMO (WSMO Studio) που διατίθεται στη μορφή επιπρόσθετων (plugin) εφαρμογών στο πλαίσιο Eclipse (<http://www.eclipse.org>).

Στο Σχήμα 9 φαίνεται το χαρακτηριστικό Choreography Editor του WSMO Studio.

Ο εκδότης WSMO επιτρέπει τη σημασιολογική περιγραφή υπηρεσιών ιστού όπως παρουσιάζεται στην ενότητα 2.3 και μπορεί επίσης να υποστηρίξει την περιγραφή σημασιολογικών υπηρεσιών ιστού σε WSDL-S η οποία παρουσιάζεται στην ενότητα 2.4. Ένα ακόμα ενδιαφέρον μέρος του WSMO Studio είναι ο Σχεδιαστής Χορογραφιών (choreography designer) που επιτρέπει το σχεδιασμό χορογραφιών για υπηρεσίες ιστού όπως προβλέπεται από το WSMO (ενότητα 2.3).



Σχήμα 9 WSMO Choreography Editor (από τις ιστοσελίδες <http://www.wsmo.org>).

3.2 Εργαλεία ανάπτυξης σημασιολογικών υπηρεσιών ιστού

Αρκετά εργαλεία σχετιζόμενα με την ανάπτυξη εφαρμογών για σημασιολογικές υπηρεσίες ιστού προέρχονται από το πρόγραμμα MINDSWAP (Maryland Information and Network Dynamics lab Semantic Web Agents Project, <http://www.mindswap.org>) του οποίου η ιστοσελίδα δίνεται στο Σχήμα 10.

Το πρόγραμμα MINDSWAP έχει υλοποιήσει μία διεπαφή προγραμματισμού εφαρμογών (API) για την ανάπτυξη και χρήση υπηρεσιών ιστού χρησιμοποιώντας τη γλώσσα OWL-S. Είναι έτσι δυνατό να προγραμματισθούν κλήσεις ατομικών διεργασιών όπως περιγράφονται σε OWL-S ή και σύνθετων διεργασιών με τη σειρά που προσδιορίζεται σε OWL-S. Εκτός από αυτό έχει αναπτύξει μια σειρά άλλων ευκολιών όπως ο μετατροπέας περιγραφών WSDL σε σκελετούς περιγραφών OWL-S, προγράμματα αιτιολόγησης (reasoner) ενώ εργάζεται για την ανάπτυξη σημασιολογικών πυλών παγκοσμίου ιστού (semantic portals) κλπ.



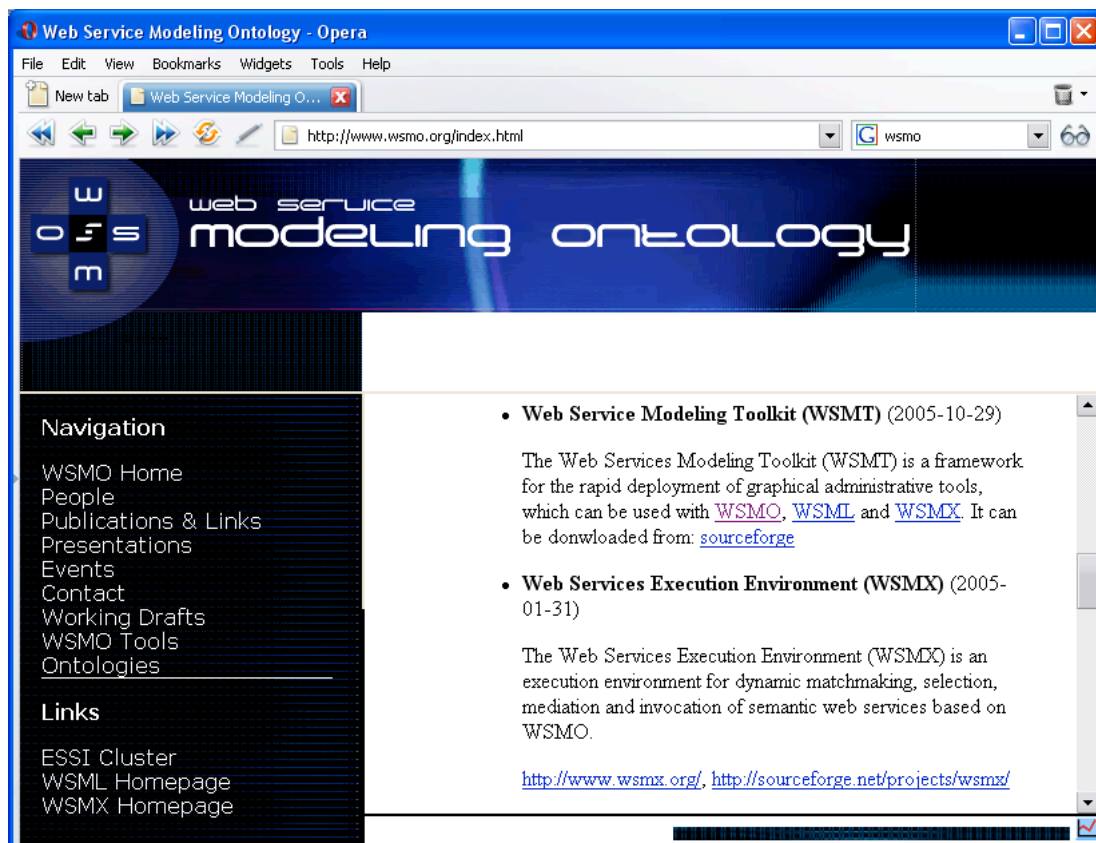
Σχήμα 10 Εργαλεία ανάπτυξης OWL-S από την ιστοσελίδα του προγράμματος MINDSWAP.

Αντίστοιχη δραστηριότητα έχει αναπτυχθεί και για σημασιολογικές υπηρεσίες ιστού που μοντελοποιούνται σύμφωνα με το WSMO.

Συγκεκριμένα αναπτύσσεται το Web Service Modelling Toolkit (WSMT) που προσφέρει ένα πλαίσιο ανάπτυξης εφαρμογών μοντελοποίησης ΣΥΙ βασισμένων στην WSMML (Web Services Modelling Language) όπως περιγράφεται στην ενότητα 2.3. Σημαντική είναι επίσης η ανάπτυξη του Web Services Execution Environment (WSMX) που προσφέρει ένα περιβάλλον για δυναμικό ταίριασμα (matchmaking), διαμεσολάβηση (mediation) και κλήση υπηρεσιών ιστού που έχουν μοντελοποιηθεί κατά την WSMO. Το Σχήμα 11 δείχνει την ιστοσελίδα με τα εργαλεία που αναπτύσσονται στα πλαίσια της προσπάθειας της ομάδας της WSMO.

Στα πλαίσια του προγράμματος METEOR-S (<http://lsdis.cs.uga.edu/projects/meteor-s/>) αναπτύσσονται εργαλεία για ανάπτυξη υπηρεσιών ιστού χαρακτηρισμένων με βάση την WSDL-S που περιγράφεται στην ενότητα 2.4. Μέχρι στιγμής μπορεί να υποστηριχθεί ο χαρακτηρισμός υπηρεσιών ιστού και ταίριασμα (matchmaking). Τα εργαλεία αυτά προσφέρονται ως τμήμα του πλαισίου ανάπτυξης εφαρμογών Emerging Technologies Toolkit (ETTK, <http://www.alphaworks.ibm.com/ettk>) της IBM.

Περαιτέρω εργασίες για την ανάπτυξη εργαλείων που έχουν εφαρμογή στο Σχεδιασμό Ενεργειών Τεχνητής Νοημοσύνης (όπως παρουσιάζεται στην ενότητα 2.5) βρίσκονται σε εξέλιξη. Είναι πιθανό να υπάρξουν εξελίξεις στα πλαίσια των εργασιών για το Σχεδιαστή Ενεργειών SHOP2 (<http://www.cs.umd.edu/projects/shop>) και των εργασιών για το πλαίσιο ανάπτυξης σύνθετων υπηρεσιών ιστού SWORD (Ponnekanti, Fox, 2002) το οποίο βασίζεται σε κανόνες για τη δημιουργία σχεδίων (rule-based planning).



Σχήμα 11 Εργαλεία ανάπτυξης με βάση το WSMO από την ιστοσελίδα του οργανισμού..

4 Μελλοντικές κατευθύνσεις

Οι εργασίες που έχουν γίνει μέχρι τώρα στην περιοχή των αρχιτεκτονικών υπηρεσιών ιστού και τις υπηρεσίες ιστού φαίνεται να εξασφαλίζουν τυποποιημένους τρόπους επικοινωνίας μέσω μηνυμάτων και περιγραφής υπηρεσιών ιστού. Ιδιαίτερα τα πρωτόκολλα SOAP και WSDL φαίνεται να τυγχάνουν ευρείας αποδοχής και αρκετά προϊόντα τα χρησιμοποιούν.

Ο τομέας που μένει να καλυφθεί επαρκώς είναι αυτός της ανακάλυψης και σύνθεσης υπηρεσιών ιστού λαμβάνοντας υπόψη ότι αυτό πρέπει να γίνεται με ακρίβεια όσον αφορά το μετασχηματισμό δεδομένων και καταστάσεων και σε ευρεία κλίμακα. Για αυτό το λόγο η ερευνητική κοινότητα έχει στραφεί προς μια πληθώρα κατευθύνσεων που αφορούν τη σημασιολογική περιγραφή υπηρεσιών ιστού και επίσης την ανακάλυψη υπηρεσιών ιστού και το ταίριασμα επιθυμητών στόχων και μετασχηματισμών με ΣΥΙ. Ακόμα όμως υπάρχουν σημαντικά προβλήματα όσον αφορά την απόδοση των προτεινόμενων τεχνικών και την κλίμακα στην οποία θα μπορούν να έχουν εφαρμογή.

Όσον αφορά το ταίριασμα (matchmaking) με υπηρεσίες ιστού (σύνθετες ή απλές) αυτή τη στιγμή δεν φαίνεται κάποια συγκεκριμένη τεχνική να κερδίζει έδαφος έναντι των άλλων. Επίσης, η σημασιολογική περιγραφή των υπηρεσιών ιστού κληρονομεί όλα τα προβλήματα που έχουν να κάνουν με διαλειτουργικότητα και συσχέτιση εννοιών από διαφορετικές οντολογίες.

Μεγαλύτερο βάρος έχει δοθεί μέχρι σήμερα στην ανάπτυξη του θεωρητικού υπόβαθρου για όλα αυτά τα θέματα των ΣΥΙ. Ταυτόχρονα όμως φαίνεται να ωριμάζουν αντίστοιχες πλατφόρμες και πλαίσια ανάπτυξης ΣΥΙ οι οποίες θα είναι σε θέση επιτρέψουν δοκιμές να δώσουν περισσότερα πειραματικά αποτελέσματα όσον αφορά την απόδοση και την κλίμακα ανάπτυξης αυτών των τεχνολογιών.

Οι κυρίαρχες τάσεις που αναμένεται να επικρατήσουν το επόμενο διάστημα θα εστιάζονται σε δυο κατευθύνσεις. Η μία κατεύθυνση αναμένεται να είναι η ανάπτυξη προηγμένων και πλούσιας περιγραφής ΣΥΙ των οποίων η αξία θα πρέπει να αποδειχθεί σε εφαρμογές που αφορούν μικρή κλίμακα, δηλ. μεταξύ ενός πολύ μικρού αριθμού παρόχων.

Η άλλη κατεύθυνση αναμένεται να αφορά την ανάπτυξη απλούστερων ΣΥΙ των οποίων η αξία θα πρέπει να δοκιμαστεί και να εδραιωθεί με χρήση σε ευρεία κλίμακα, δηλ. με τη συμμετοχή ενός μεγαλύτερου αριθμού παρόχων. Αυτή η κατεύθυνση απαιτεί και τους κατάλληλους χειρισμούς από τους οργανισμούς τυποποίησης αφού η ευρεία κλίμακα εφαρμογής απαιτεί συνεννόηση και συνεργασία σε επίπεδο προτύπων.



Αναφορές

- Erol, K., Nau, D., Hendler, J. (1994). U MCP: A Sound and Complete Planning Procedure for Hierarchical Task-Network Planning. In AIPS-94, Chicago, June, 1994.
- Fensel, D., Bussler, C. (2002). *The Web Service Modeling Framework WSMF*, Electronic Commerce: Research and Applications, 1 (2), pp. 113-137.
- Hendler, J., Tate, A., and Drummond, M. 1990. *AI planning: systems and techniques*. AI Mag. 11, 2 (Apr. 1990), 61-77.
- Krafzig, D., Banke, K., Slama, D. (2005). *Enterprise SOA-Service-Oriented Architecture Best Practices*, Pearson Education.
- Laukkanen, M., Helin, H. (2003). *Composing Workflows of Semantic Web Services*. In Proceedings of AAMAS Workshop on Web Services and Agent-Based Engineering, Melbourne, Australia, 2003.
- McIlraith, S., Martin, D. (2003) *Bringing Semantics to Web Services*. In IEEE Intelligent Systems 18(1).
- McIlraith S., Son, T. (2002) *Adapting Golog for Composition of Semantic Web Services*. In Proceedings of the Eighth International Conference on Knowledge Representation and Reasoning (KR2002), pp. 482-493.
- Medjahed, B., Bouguettaya, A., Elmagarmid, A. (2003). *Composing Web Services on the Semantic Web*. The VLDB Journal, 12(4), November 2003.
- Nau, D. S., Au, T. C., Ilghami, O., Kuter, U., Murdock, J. W., Wu, D., Yaman, F. (2003). *SHOP2: An HTN planning system*. Journal of Artificial Intelligence Research (JAIR) 20, pp. 379–404.
- Newcomer, E., Lomow, G. (2005). *Understanding SOA with Web Services*, Pearson Education.
- Pandis, I. (2004) "A survey on Semantic Web Services, state-of-the-art, current work, and challenges", Independent Study, Athens Information Technology Centre, December 2004.
- Patil, A., Oundhakar, S., Sheth, A., Verma, K. (2004). *METEOR-S Web Service Annotation Framework*, In Proceedings International WWW Conference, New York, USA.
- Polleres, R., Lara, R. (Eds) (2005), *A Conceptual Comparison between WSMO and OWL-S*, WSMO Working Group working draft.
- Ponnekanti, S. R., Fox, A. (2002). *SWORD: A Developer Toolkit for Web Service Composition*. In Eleventh World Wide Web Conference (Web Engineering Track), Honolulu, Hawaii, May 2002.
- Sivashanmugam, K., Verma, K., Sheth, A., and Miller, J. (2003). *Adding Semantics to Web Services Standards*. In Proceedings of the 1st International Conference on Web Services (ICWS'03), Las Vegas, Nevada (June 2003) pp. 395 - 401.
- Weerawarana, S., Curbera, F., Leymann, F., Storey, F., Ferguson, D. F. (2005). *Web Services Platform Architecture*, Pearson Education.

Διευθύνσεις URL

1. Η ιστοσελίδα του οργανισμού W3C: <http://www.w3.org/>
2. Η ιστοσελίδα του οργανισμού OASIS Open: <http://www.oasis-open.org>
3. Η ιστοσελίδα του οργανισμού W3C για την ομάδα εργασίες σε αρχιτεκτονική υπηρεσιών ιστού: <http://www.w3.org/TR/ws-arch/>
4. Η ιστοσελίδα του οργανισμού W3C για το πρωτόκολλο SOAP: <http://www.w3.org/TR/soap/>
5. Η ιστοσελίδα του οργανισμού W3C για τη γλώσσα WSDL 1.1: <http://www.w3.org/TR/wsdl/>
6. Η ιστοσελίδα του οργανισμού W3C για τη γλώσσα WSDL 2.0: <http://www.w3.org/TR/wsdl20/>
7. Η ιστοσελίδα του οργανισμού OASIS Open για το πρωτόκολλο UDDI: <http://www.uddi.org>
8. Η ιστοσελίδα του οργανισμού OASIS Open για τη γλώσσα σύνθεσης υπηρεσιών ιστού BPEL: http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel
9. Η ιστοσελίδα της ομάδας του W3C που εργάζεται για τη χορογραφία (choreography) υπηρεσιών ιστού: <http://www.w3.org/2002/ws/chor/>
10. Η ιστοσελίδα του οργανισμού W3C για την ομάδα ενδιαφέροντος σε σημασιολογικές υπηρεσίες ιστού: <http://www.w3.org/2002/ws/swsig/>
11. Η ιστοσελίδα του οργανισμού WSMO: <http://www.wsmo.org/>
12. Η ιστοσελίδα για την προτεινόμενη γλώσσα OWL-S: <http://www.w3.org/Submission/OWL-S/>
13. Η ιστοσελίδα για το προτεινόμενο πρότυπο WSMO: <http://www.w3.org/Submission/WSMO/>
14. Η ιστοσελίδα για την προτεινόμενη γλώσσα WSDL-S: <http://www.w3.org/Submission/WSDL-S/>
15. Η ιστοσελίδα για το προτεινόμενο πρότυπο SWSF: <http://www.w3.org/Submission/SWSF/>
16. Η ιστοσελίδα του προγράμματος MINDSWAP: <http://www.mindswap.org/>
17. Η ιστοσελίδα του εκδότη OWL-S (OWL-S Editor): <http://owlseeditor.semwebcentral.org/>
18. Η ιστοσελίδα του προγράμματος METEOR-S: <http://lsdis.cs.uqa.edu/projects/meteor-s/>
19. Η ιστοσελίδα του πλαισίου ανάπτυξης εφαρμογών ETTK (Emerging Technologies Toolkit) της IBM: <http://www.alphaworks.ibm.com/ettk/>
20. Η ιστοσελίδα του προγράμματος SHOP: <http://www.cs.umd.edu/projects/shop/>

Ακρωνύμια – Συντομεύσεις

EE	Είσοδος Έξοδος
ΕΕΠΑ	Είσοδος Έξοδος Προϋποθέσεις Αποτελέσματα
ΣΥΙ	Σημασιολογικές Υπηρεσίες Ιστού
ΦΠΥ	Φορείς Παροχής Υπηρεσιών
API	Application Programming Interface
DAML-S	DARPA Agent Markup Language for Services
FOAF	Friend Of A Friend
FOL	First Order Logic
HTN	Hierarchical Task Network
NAICS	North American Industry Classification System
OASIS	Organisation for the Advancement of Structured Information Standards
OWL-S	Web Ontology Language for Services
PPM	Polymorphic Process Model
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol (στην αρχική έκδοση του προτύπου SOAP)
SWS	Semantic Web Services (συντόμευση)
SWSF	Semantic Web Services Framework
SWSL	Semantic Web Services Language

SWWS Semantic Web-enabled Web Services
UDDI Universal Description, Discovery and Integration
UNSPSC United Nations Standard Products and Services Code
W3C World Wide Web Consortium
WS Web Services (συντόμηση)
WSBPEL Web Services Business Process Execution Language
WC-CDL Web Service Choreography Description Language
WSDL Web Services Description Language
WSDL-S Web Service Semantics
WSMF Web Services Modeling Framework
WSML Web Services Modeling Language
WSMO Web Service Modeling Ontology