

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΕΝΣΥΡΜΑΤΗΣ ΤΗΛΕΠΙΚΟΙΝΩΝΙΑΣ

ΣΧΕΔΙΑΣΗ ΚΑΙ ΑΝΑΠΤΥΞΗ WEB SERVICE

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΠΑΠΑΡΓΥΡΗ ΤΡΙΑΝΤΑΦΥΛΛΙΑ
Του ΚΩΝΣΤΑΝΤΙΝΟΥ
ΦΟΙΤΗΤΡΙΑΣ ΤΟΥ ΤΜΗΜΑΤΟΣ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ
ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ

ΕΠΙΒΛΕΠΩΝ: ΕΠΙΚΟΥΡΟΣ ΚΑΘΗΓΗΤΗΣ Μ. ΚΟΥΚΙΑΣ

ΑΡΙΘΜΟΣ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ: / 2012

ΠΑΤΡΑ
ΙΟΥΝΙΟΣ 2012

ΠΙΣΤΟΠΟΙΗΣΗ

Πιστοποιείται ότι η διπλωματική εργασία με θέμα:

ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΑΝΑΠΤΥΞΗ WEB SERVICE

Της φοιτήτριας του Τμήματος Ηλεκτρολόγων Μηχανικών και
Τεχνολογίας Υπολογιστών

Παπαργύρη Τριανταφυλλιάς του Κωνσταντίνου
(AM 4611)

Παρουσιάστηκε δημόσια και εξετάστηκε στο τμήμα Ηλεκτρολόγων
Μηχανικών και Τεχνολογίας Υπολογιστών στις .../.../2012

Ο Επιβλέπων
Καθηγητής

Ο Διευθυντής του Τομέα Τ&ΤΠ
Καθηγητής

ΤΙΤΛΟΣ : ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΑΝΑΠΤΥΞΗ WEB SERVICE

Φοιτήτρια: Παπαργύρη Τριανταφυλλιά

Επιβλέπων: Μ. Κουκιάς

Περίληψη

Η ύπαρξη Διαδικτύου δημιούργησε την ιδιότητα του <<Παγκόσμιου Πολίτη>>, δίνοντας την δυνατότητα σε όλους τους χρήστες του Διαδικτύου να μπορούν να έχουν πρόσβαση σε μια πληθώρα πληροφοριών, να επικοινωνούν μεταξύ τους, αλλά και να μπορούν και οι ίδιοι να μετατρέπονται από χρήστες του Διαδικτύου σε Παρόχους Υπηρεσιών μέσω του Διαδικτύου. Την σημερινή εποχή αναγνωρίζεται ως μια από τις σημαντικότερες εφαρμογές του Διαδικτύου και μάλιστα όχι άδικα, η τεχνολογία των Web Services, η οποία επιτρέπει στις εφαρμογές να επικοινωνούν μεταξύ τους ανεξαρτήτως πλατφόρμας και γλώσσας προγραμματισμού. Τα πρώτα web services σκόπευαν να είναι πηγές πληροφορίας τις οποίες μπορεί κανείς πολύ εύκολα να ενσωματώσει στις εφαρμογές του. Στην συνέχεια εξελίχθηκαν, προσφέροντας όλα τα δυναμικά τους χαρακτηριστικά ώστε να συνδυάζονται υπηρεσίες σε εφαρμογές.

Στην παρούσα διπλωματική εργασία γίνεται μια εκτενής αναφορά στον σχεδιασμό και στην ανάπτυξη των Web Services. Αρχικά παρουσιάζονται οι βασικές τεχνολογίες των Web Services, η γλώσσα XML, το πρωτόκολλο SOAP, καθώς και οι WSDL και UDDI, οι οποίες αποτελούν τον <<δομικό σκελετό>> των Υπηρεσιών Διαδικτύου. Στην συνέχεια αντιμετωπίζουμε μια Υπηρεσία Web από την πλευρά του Παρόχου Υπηρεσίας. Αναφερόμαστε δηλαδή στους τρόπους υλοποίησης μιας τέτοιας εφαρμογής και στα βασικά εργαλεία προγραμματισμού της. Στο τελευταίο κομμάτι της παρούσας διπλωματικής παρουσιάζονται οι τελευταίες απαιτούμενες τεχνολογίες, ώστε να μπορούμε να μιλάμε για μια ολοκληρωμένη κατανεμημένη τεχνολογία των Υπηρεσιών Διαδικτύου.

Πιο συγκεκριμένα το κεφάλαιο 1 αποτελεί την εισαγωγή της παρούσας διπλωματικής, όπου παρουσιάζονται οι βασικές έννοιες του Διαδικτύου, μια ιστορική αναδρομή που αφορά την δημιουργία και εξέλιξη του, ενώ ολοκληρώνεται με μια σχετικά εκτενή αναφορά στο World Wide Web, τον Παγκόσμιο Ευρύ Ιστό, ο οποίος αποτελεί ουσιαστικά και τον πυρήνα του Internet.

Στην συνέχεια στο κεφάλαιο 2 παρουσιάζονται οι βασικές έννοιες των Υπηρεσιών Διαδικτύου και οι λόγοι που τις καθιστούν από τις πιο σημαντικές εφαρμογές του σημερινού τεχνολογικού κόσμου. Περιγράφεται συνοπτικά η βασική αρχιτεκτονική των web services και τα επίπεδα από τα οποία αποτελείται. Τέλος στο παρόν κεφάλαιο γίνεται μια αρχική εισαγωγή στα βασικότερα πρότυπα, τα οποία επιτρέπουν στους υπεύθυνους, την ανάπτυξη και υλοποίηση των Υπηρεσιών Web.

Στο κεφάλαιο 3 γίνεται μια εκτενής παρουσίαση στην περιγραφική γλώσσα XML. Όλα τα δεδομένα που μεταφέρονται κατά την διαδικασία εκτέλεσης και σχεδιασμού Web Service μετατρέπονται σε μορφή XML εγγράφου, το οποίο

συντάσσεται βασισμένο στην XML. Ταυτόχρονα στο παρόν κεφάλαιο παρουσιάζεται το DTD, και μια πιο προηγμένη προδιαγραφή, το XML Schema, οι οποίες καθορίζουν το πλαίσιο δομής των XML εγγράφων ώστε να θεωρούνται έγκυρα. Τέλος γίνεται μια συνοπτική παρουσίαση του λεκτικού αναλυτή XML Parser.

Στο κεφάλαιο 4 περιγράφεται το πρότυπο δικτυακό πρωτόκολλο SOAP, το οποίο ορίζει την ανταλλαγή δομημένων πληροφοριών σε οποιοδήποτε κατανεμημένο περιβάλλον και κατά επέκταση και στις Υπηρεσίες Διαδικτύου.

Όλες οι εφαρμογές που παρέχουν Web Services, διαφημίζουν τις διάφορες υπηρεσίες τους χρησιμοποιώντας ένα συγκεκριμένο πρότυπο γλώσσας. Στο κεφάλαιο 5 παρουσιάζεται η πρότυπη περιγραφική γλώσσα WSDL, η οποία κάνει χρήση της XML και μέσω ενός ειδικού συνόλου ετικετών περιγράφει τις Υπηρεσίες Διαδικτύου, τις λειτουργίες που αυτές παρέχουν κλπ. Ο <<χώρος>> στον οποίο όλες οι εφαρμογές παροχής Web Service διαφημίζουν τις διάφορες υπηρεσίες τους είναι το μητρώο Υπηρεσιών UDDI. Στο κεφάλαιο 6 γίνεται μια εκτενής αναφορά στο συγκεκριμένο πρότυπο, το οποίο ουσιαστικά αποτελεί τον <<χρυσό οδηγό>> των Web Services.

Σε αυτό το σημείο ολοκληρώνεται η αναλυτική περιγραφή των τριών βασικών τεχνολογιών των Web Services, οι οποίες παρέχουν και τις βασικές λειτουργίες των Web Services καθώς και της περιγραφικής γλώσσας XML στην οποία βασίζονται. Στο Κεφάλαιο 7 ασχολούμαστε με τους τρόπους υλοποίησης των Υπηρεσιών Διαδικτύου αναφερόμενοι στα βασικά εργαλεία προγραμματισμού. Τέλος στο τελευταίο κεφάλαιο δεν θα μπορούσαμε να μην αναφερθούμε στις απειλές που δέχεται κάθε Υπηρεσία Web από πιθανούς "εισβολείς", στους βασικότερους τρόπους αντιμετώπισης τους καθώς και στην συνεργασία που μπορεί να υπάρχει μεταξύ περισσότερων της μιας Υπηρεσιών Διαδικτύου

ABSTRACT

The existence of the Internet created the position of << World Citizen>>, giving the opportunity to all users to be able to access a wealth of information, to communicate with each other, but also can be converted from users of Internet to Internet Service Providers. At this moment recognized as one of the most important applications of the Internet and not unjustly, the technology of Web Services, which allows applications to communicate regardless of platform and programming language. The initial web services intended to be sources of information which can easily incorporate into applications. Then web services were developed by offering all dynamic characteristics to combine services to applications.

In this diploma become a comprehensive reference to the design and development of Web Services. Originally presented Initially the basic technologies of Web Services, the language XML, the protocol SOAP, WSDL and the UDDI, which is the <<structural frame>> of Web Services. Then we have a Web Service from the service provider side. We refer to the ways of implementation of such an application and on the basic development tools. In the last part of this thesis presented the latest technologies required so we can talk about a comprehensive distributed technology of the Web Services.

More specifically, Chapter 1 is an introduction to this diploma, which presents the basic concepts of the Internet, a throwback to Internet's creation and evolution, and concludes with a comprehensive report on the World Wide Web, which is essentially the core of the Internet

Then Chapter 2 presents the basic concepts of Web Services and the reasons that make them the most important applications of modern technological world. It is described the basic architecture of web services and their levels. Finally in this chapter presents an initial introduction to the basic standards that allow managers to develop and implement the Web Service.

In chapter 3 become an extensive presentation of descriptive language XML. All transferred data during the process of planning and executing Web Service converted to XML document, drawn up based on XML language. At the same time in this chapter presents the DTD, and a more advanced specification, the XML Schema, which set the framework structure of XML documents to be considered valid. Finally presents an overview of the word analyzer XML Parser

Chapter 4 describes the standard network protocol SOAP, which defines the exchange of structured information in any distributed environment and extending to Web Services.

All applications that provide Web Services, advertise their various services, by using a specific template language. Chapter 5 presents the standard descriptive language WSDL, which use XML language and by using a special set of tags describes Web services, the functions they provide etc. The <<space>>, in which all applications for the Web Service advertise their various services, is the service registry UDDI. In chapter 6 presents this template, which essentially is the <<yellow pages>> of the Web Services.

At this point completes the detailed description of three basic technologies of Web Services and XML descriptive language in which they are based and which provide the basic functions of Web Services. Chapter 7 deals with the ways of implementation of the Web Services referred to the basic tools. Finally at the last chapter we could not mention the threats to each Web Service from "potential intruders", the main ways to confront them and the collaboration that can exist between more than one Web Services.

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΕΧΟΜΕΝΑ	9
1 ΕΙΣΑΓΩΓΗ	11
1.1 ΕΙΣΑΓΩΓΗ ΣΤΟ ΔΙΑΔΙΚΤΥΟ	11
1.2 Η ΕΞΕΛΙΞΗ ΤΟΥ ΔΙΑΔΙΚΤΥΟΥ	11
1.3 WORLD WIDE WEB (WWW)	13
2 WEB SERVICES	15
2.1 ΟΡΙΣΜΟΣ ΤΩΝ WEB SERVICES	15
2.2 Η ΔΥΝΑΜΗ ΤΩΝ WEB SERVICES	16
2.3 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΩΝ WEB SERVICES (SOA)	17
2.4 ΤΕΧΝΟΛΟΓΙΕΣ ΤΩΝ WEB SERVICES	18
3 EXTENSIBLE MARKUP LANGUAGE - XML	21
3.1 ΕΙΣΑΓΩΓΗ ΣΤΗΝ XML	21
3.2 ΒΑΣΙΚΗ ΟΡΟΛΟΓΙΑ ΤΗΣ XML	22
3.3 DTD-DOCUMENT TYPE DEFINITION	24
3.3.1 XML Parsers	25
3.4 XML SCHEMAS	25
4 SOAP-SIMPLE OBJECT ACCESS PROTOCOL	27
4.1 ΕΙΣΑΓΩΓΗ ΣΤΙΣ ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ	27
4.2 ΜΗΝΥΜΑ SOAP	29
4.2.1 Envelope (Φάκελος)	30
4.2.2 Header (Επικεφαλίδα)	30
4.2.3 Header (Σώμα)	31
4.3 ΜΟΝΤΕΛΟ ΕΠΕΞΕΡΓΑΣΙΑΣ SOAP	32
4.4 ΜΟΝΤΕΛΟ ΑΝΤΑΛΛΑΓΗΣ ΜΗΝΥΜΑΤΩΝ	33
4.4.1 Απομακρυσμένες Κλήσεις Διαδικασιών	33
4.4.2 HTTP Binding	34
4.5 ΠΑΡΑΔΕΙΓΜΑ ΕΝΟΣ SOAP ΜΗΝΥΜΑΤΟΣ	35
5 WSDL - WEB SERVICES DESCRIPTION LANGUAGE	37
5.1 ΕΙΣΑΓΩΓΗ ΣΤΙΣ ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ	37
5.2 ΤΕΧΝΙΚΗ ΠΕΡΙΓΡΑΦΗ ΤΗΣ WSDL	38
6 UDDI - UNIVERSAL DESCRIPTION, DISCOVERY AND INTEGRATION	41
6.1 ΕΙΣΑΓΩΓΗ ΣΤΙΣ ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ	41
6.2 ΟΡΙΣΜΟΣ ΔΟΜΩΝ ΔΕΔΟΜΕΝΩΝ	43
7 ΥΛΟΠΟΙΗΣΗ WEB SERVICE	45
7.1 ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΑ ΕΡΓΑΛΕΙΑ ΣΥΣΤΗΜΑΤΟΣ	45
7.1.1 Περιβάλλον Πλατφόρμας	45
VISUAL STUDIO	45
TWISTER	46
7.1.2 NET Framework	46
Common Language Runtime (CLR)	47
Βιβλιοθήκες Κλάσεων	47
Γλώσσες προγραμματισμού	48
ASP-NET	48
7.1.3 Web server	48
7.1.4 Apache Axis	48
7.1.5 Συστήματα Διαχείρισης Βάσης Δεδομένων	49

MicrosoftSQL	49
PostgreSQL	50
7.1.6 Τρόποι επικοινωνίας με την Βάση Δεδομένων	51
7.2 ΤΕΧΝΟΛΟΓΙΕΣ	51
7.2.1 Java.....	52
7.2.2 Πακέτο Δημιουργίας Web Service.....	52
7.2.3 Servlet	53
7.2.4 Δημιουργία Δυναμικών Σελίδων.....	53
7.3 ΠΑΡΑΔΕΙΓΜΑΤΑ ΥΛΟΠΟΙΗΣΗΣ WEB SERVICE.....	54
8 ΠΡΟΣΘΕΤΕΣ ΤΕΧΝΟΛΟΓΙΕΣ	59
8.1 ΑΠΕΙΛΕΣ ΣΤΙΣ ΔΙΑΔΙΚΤΥΑΚΕΣ ΥΠΗΡΕΣΙΕΣ	59
8.1.1 Parameter Tampering - SQL Injection	59
8.1.2 Soap messages flooding	60
8.1.3 Schema Poisoning	60
8.1.4 Παρέμβαση κατά την Δρομολόγηση.....	61
8.1.5 Malicious Contents.....	62
8.2 ΕΞΑΣΦΑΛΙΣΗ ΑΣΦΑΛΕΙΑΣ ΣΤΙΣ WEB SERVICES	62
8.2.1 Πιστοποίηση για Σύνδεση πάνω από HTTP	63
8.2.2 Πιστοποίηση για σύνδεση πάνω από HTTPS	63
8.2.3 Ψηφιακή Υπογραφή-XML Signature	63
8.2.4 Ψηφιακή Κρυπτογράφηση-XML Encryption	64
8.2.5 SAML-XACML	64
8.2.6 XKMS	65
8.2.7 Υπό εξέλιξη προδιαγραφές	65
8.3 WEB SERVICES ΚΑΙ ΣΥΝΑΛΛΑΓΕΣ	66
8.3.1 BPEL	66
8.3.1.1 Δομή της γλώσσας	67
8.3.2 Τεχνικές Ενορχήστρωσης και Χορογραφίας	68
9 ΒΙΒΛΙΟΓΡΑΦΙΑ	71
9.1 ΑΝΑΦΟΡΕΣ	71
9.2 ΔΙΕΥΘΥΝΣΕΙΣ URL	72

ΚΕΦΑΛΑΙΟ 1

1 ΕΙΣΑΓΩΓΗ

1.1 ΕΙΣΑΓΩΓΗ ΣΤΟ ΔΙΑΔΙΚΤΥΟ

Με την εμφάνιση οποιουδήποτε νέου μέσου επικοινωνίας, η κοινωνία και οι δομές της αναμφισβήτητα επηρεάζονται. Η επίδραση αυτή πηγάζει κυρίως από την τεχνολογία του νέου μέσου. Είναι πλέον εμφανές πως το διαδίκτυο, ως ένα από τα πλέον πιο σημαντικά μέσα επικοινωνίας στον σύγχρονο κόσμο έχει διεισδύσει σε κάθε γωνιά της υφελίου, δημιουργώντας εικονικούς "χώρους" καθώς και εικονικές "κοινότητες", όπου παύουν να υφίστανται οι γεωγραφικές, κοινωνικές και πολιτιστικές διαχωριστικές γραμμές που υπάρχουν στον πραγματικό κόσμο και που τα παραδοσιακά μέσα επικοινωνίας αδυνατούν να ξεπεράσουν εύκολα.

Με την δημιουργία του διαδικτύου, η επικοινωνία έγινε άμεση και αμφίδρομη. Οι αποστάσεις εκμηδενίστηκαν και δόθηκε η δυνατότητα σε κάθε χρήστη ηλεκτρονικού υπολογιστή συνδεδεμένου στο Διαδίκτυο, να πληροφορηθεί αλλά και να πληροφορήσει ανταλλάσσοντας απόψεις μέσω ενός συμμετοχικού διαύλου επικοινωνίας. Οι χρήστες αποκτούν ολοένα και περισσότερο την ιδιότητα του παγκόσμιου πολίτη. Είναι εμφανείς λοιπόν οι λόγοι για τους οποίους το διαδίκτυο θεωρείται ένα από τα πιο δημοκρατικά σύγχρονα μέσα μαζικής επικοινωνίας καθώς δίνει στον μέσο άνθρωπο τη δυνατότητα πρόσβασης σε μεγάλο όγκο πληροφοριών συγκεντρωμένων σε ένα "χώρο" και την δυνατότητα της προσωπικής επιλογής των πληροφοριών αυτών.

Ωστόσο δε γίνεται να μην αναφερθούμε και στα μειονεκτήματα που προκύπτουν μέσω της ελευθερίας της έκφρασης και της άμεσης πρόσβασης σε ένα μεγάλο όγκο πληροφοριών που υποστηρίζει το Διαδίκτυο. Η παραβίαση πνευματικών δικαιωμάτων, η πορνογραφία, η ψευδοπροσωπία και η προσφορά παρανόμων προϊόντων είναι μερικά μόνο από τα αποτελέσματα της κακής χρήσης των δυνατοτήτων που παρέχει το Διαδίκτυο. Το τεχνικό και νόμιμο framework της λειτουργίας του Internet θα πρέπει να αντιμετωπίσει την αναταραχή της κακής αυτής χρήσης μέσα από την δικιά του διαδικασία εξέλιξης.

1.2 Η ΕΞΕΛΙΞΗ ΤΟΥ ΔΙΑΔΙΚΤΥΟΥ

Στην πιο εξειδικευμένη και περισσότερο χρησιμοποιούμενη μορφή του, με τους όρους Διαδίκτυο (με κεφαλαίο το αρχικό γράμμα) περιγράφεται το παγκόσμιο πλέγμα διασυνδεδεμένων υπολογιστών και των υπηρεσιών και πληροφοριών που παρέχει στους χρήστες του. Η τεχνολογία του είναι κυρίως βασισμένη στην διασύνδεση επιμέρους δικτύων ανά τον κόσμο και πολυάριθμα τεχνολογικά

πρωτόκολλα. Ωστόσο είναι προφανές πως για να φτάσει στην σημερινή του δομή ακολούθησε την δική του διαδρομή εξέλιξης.

Το σημερινό Διαδίκτυο (Internet) αποτελεί εξέλιξη του ARPANET, ενός δικτύου που άρχισε να αναπτύσσεται πειραματικά στα τέλη της δεκαετίας του 60 στις ΗΠΑ. Εγκαταστάθηκε και λειτούργησε για πρώτη φορά το 1969 με 4 κόμβους μέσω των οποίων συνδέονται 4 μίνι υπολογιστές. Μέχρι το 1972 οι συνδεδεμένοι στο ARPANET υπολογιστές είχαν φτάσει τους 23, οπότε και εφαρμόζεται για πρώτη φορά το σύστημα διαχείρισης ηλεκτρονικού ταχυδρομείου (e-mail). Παράλληλα δημιουργήθηκαν και άλλα δίκτυα, που χρησιμοποιούσαν διαφορετικές μεθόδους και τεχνικές (όπως το x.25 και το UUCP) τα οποία συνδέονταν με το ARPANET. Το πρωτόκολλο που χρησιμοποιούσε το ARPANET ήταν το NCP (Network Control Protocol), το οποίο όμως είχε το μειονέκτημα ότι λειτουργούσε μόνο με συγκεκριμένους τύπους υπολογιστών. Έτσι δημιουργήθηκε η ανάγκη στις αρχές του 1970 για ένα πρωτόκολλο που θα ένωνε όλα τα δίκτυα που είχαν δημιουργηθεί μέχρι τότε.

Το 1973 ξεκινά ένα νέο ερευνητικό πρόγραμμα που ονομάζεται Interneting Project (Πρόγραμμα Διαδικτύωσης). Από την έρευνα γεννιέται μια νέα τεχνική, το Internet Protocol (IP) (Πρωτόκολλο Διαδικτύωσης), από την οποία θα πάρει αργότερα το όνομά του το Internet. Διαφορετικά δίκτυα που χρησιμοποιούν το κοινό πρωτόκολλο IP μπορούν να συνδέονται και να αποτελούν ένα διαδίκτυο. Επίσης σχεδιάζεται μια άλλη τεχνική για τον έλεγχο της μετάδοσης των δεδομένων, το Transmission Control Protocol (TCP) (Πρωτόκολλο Ελέγχου Μετάδοσης). Ορίζονται προδιαγραφές για τη μεταφορά αρχείων μεταξύ υπολογιστών (FTP) και για το ηλεκτρονικό ταχυδρομείο (E-mail). Σταδιακά συνδέονται με το ARPANET ιδρύματα από άλλες χώρες, με πρώτα το University College of London (Αγγλία) και το Royal Radar Establishment (Νορβηγία).

Το 1983, το πρωτόκολλο TCP/IP (δηλ. ο συνδυασμός των TCP και IP) αναγνωρίζεται ως πρότυπο από το Υπουργείο Άμυνας των ΗΠΑ. Η έκδοση του λειτουργικού συστήματος Berkeley UNIX το οποίο περιλαμβάνει το TCP/IP συντελεί στη γρήγορη εξάπλωση της δικτύωσης των υπολογιστών. Το 1985 το National Science Foundation (NSF) έκανε ένα σημαντικό βήμα στην ανάπτυξη του Διαδικτύου. Δημιουργεί την πρώτη διαδικτυακή πανεπιστημιακή ραχοκοκαλιά (backbone), το NSFNet, χρησιμοποιώντας το πρωτόκολλο TCP/IP. Στα τέλη της δεκαετίας του '80, όλο και περισσότερες χώρες συνδέονται στο NSFNet (Καναδάς, Γαλλία, Σουηδία, Αυστραλία, Γερμανία, Ιταλία, κ.α.). Χιλιάδες πανεπιστήμια και οργανισμοί δημιουργούν τα δικά τους δίκτυα και τα συνδέουν πάνω στο παγκόσμιο αυτό δίκτυο το οποίο αρχίζει να γίνεται γνωστό σαν INTERNET και να εξαπλώνεται με τρομερούς ρυθμούς σε ολόκληρο τον κόσμο. Το 1990 το ARPANET πλέον καταργείται.

Το παγκόσμιο δίκτυο για όλους είναι πλέον γεγονός. Όλο και περισσότερες χώρες συνδέονται στο NSFNet, μεταξύ των οποίων και η Ελλάδα το 1990. Το 1993 το εργαστήριο CERN στην Ελβετία παρουσιάζει το World Wide Web (WWW) - (Παγκόσμιο Ιστό) που αναπτύχθηκε από τον Tim Berners-Lee. Πρόκειται για ένα σύστημα διασύνδεσης πληροφοριών σε μορφή πολυμέσων (multimedia) που βρίσκονται αποθηκευμένες σε χιλιάδες υπολογιστές του Internet σε ολόκληρο τον κόσμο και παρουσίασης τους σε ηλεκτρονικές σελίδες, στις οποίες μπορεί να περιηγηθεί κανείς χρησιμοποιώντας το ποντίκι. Το γραφικό αυτό περιβάλλον έκανε την εξερεύνηση του Internet προσιτή στον απλό χρήστη. Παράλληλα, εμφανίζονται στο Internet διάφορα εμπορικά δίκτυα που ανήκουν σε εταιρείες παροχής υπηρεσιών

Internet (Internet Service Providers – ISP) και προσφέρουν πρόσβαση στο Internet για όλους. Η ανακάλυψη του WWW σε συνδυασμό με την ευκολία απόκτησης πρόσβασης στο Internet προσέλκυσε έναν μεγάλο αριθμό καινούργιων χρηστών και έφερε την “έκρηξη” που παρακολούθησαμε τα τελευταία χρόνια.

1.3 WORLD WIDE WEB (WWW)

Η απλούστερη περιγραφή που θα μπορούσε να δώσει κανείς για το Internet, η οποία συνοψίζει ταυτόχρονα και την ουσία της ύπαρξης και της λειτουργίας του, είναι ότι αυτό αποτελεί έναν “χώρο” όπου αποθηκεύεται η πληροφορία. Από την ίδια τη φύση της όμως η πληροφορία δεν μπορεί να είναι στατική, διότι τότε χάνει την έννοιά της. Επεκτείνοντας λοιπόν την αρχική περιγραφή, ένας πιο ακριβής ορισμός θα έλεγε ότι το Internet είναι ένας χώρος αποθήκευσης αλλά ταυτόχρονα και ένα μέσο μετάδοσης, επεξεργασίας και μετεξέλιξης της πληροφορίας.

Κατά τη διάρκεια της εξέλιξης του δικτύου, εμφανίστηκαν και δοκιμάστηκαν πολλά εργαλεία, μέθοδοι και τρόποι αποθήκευσης και μετάδοσης της πληροφορίας, αρκετοί από τους οποίους πέτυχαν και κατάφεραν τελικά να καθιερωθούν. Ωστόσο το αποφασιστικό βήμα για την ολοκλήρωση (integration) της αναζήτησης της πληροφορίας στο Internet έγινε με την εισαγωγή μίας νέας υπηρεσίας που ονομάζεται World Wide Web (WWW) ή αλλιώς WEB. Ο Παγκοσμίως Ευρύς Ιστός, με απλά λόγια το Web, είναι ένα δίκτυο υπολογιστών βασισμένο στο Internet, που επιτρέπει στους χρήστες ενός υπολογιστή να έχουν πρόσβαση σε πληροφορίες που είναι αποθηκευμένες σε κάποιο άλλο υπολογιστή μέσω αυτού του παγκόσμιου δικτύου.

Πιο συγκεκριμένα, έστω ότι έχουμε πολλούς υπολογιστές διασυνδεδεμένους μέσω καλωδιώσεων σε ένα είδος δικτύου ευρείας περιοχής WAN (Wide Area Network), που είναι ουσιαστικά η ίδια ανάπτυξη που διακρίνουμε σε όλο το Internet . Κάθε ένας από αυτούς τους υπολογιστές καλείται εξυπηρετητής ή διανομέας Web (Web server) και περιέχει ένα σύνολο από δεδομένα υπερμέσων, τα λεγόμενα Web Documents, τα οποία δεικτοδοτούνται μεταξύ τους μέσω των διευθύνσεων των υπολογιστών στους οποίους περιέχονται συν των σχετικών τους εσωτερικών διευθύνσεων εντός των υπολογιστών αυτών. Πρόκειται, στην ουσία για ένα αλληλεπιδραστικό multimedia περιβάλλον που προσφέρει τη δυνατότητα στον χρήστη να δημιουργήσει εντυπωσιακά ηλεκτρονικά έγγραφα και να τα “δημοσιεύσει” στον Web, καθιστώντας τα έτσι προσβάσιμα για οποιονδήποτε ενδιαφερόμενο.

Ο Web είναι δομημένος σε μία αρχιτεκτονική client / server. Σε μία απλουστευμένη προσέγγιση, ο server όπως είδαμε περιέχει τα λεγόμενα Web Documents, μέσα στα οποία βρίσκονται οι διάφοροι σύνδεσμοι και “διοχετεύει” τις πληροφορίες προς τον client, χειριζόμενος ταυτόχρονα και τις αιτήσεις για επικοινωνία και διασύνδεση με άλλους απομακρυσμένους υπολογιστές (hosts). Στο Web ο client αποκαλείται Web Browser. Ένας Web browser (φυλλομετρητής ιστοσελίδων, πλοηγός Web, πρόγραμμα περιήγησης Web ή περιηγητής Ιστού) είναι ένα λογισμικό που επιτρέπει στον χρήστη του να προβάλλει και να αλληλεπιδρά με όλα τα Web Documents, που είναι συνήθως αναρτημένα σε μια ιστοσελίδα ενός ιστότοπου στον Παγκόσμιο Ιστό ή σε ένα τοπικό δίκτυο. Το πρωτόκολλο που χρησιμοποιείται από τον Web για την εκτέλεση όλων αυτών των διεργασιών είναι το HTTP (Πρωτόκολλο Μεταφοράς Υπερκειμένου), ενώ η HTML ((Γλώσσα

Επισήμανσης Υπερκειμένου) αποτελεί την γλώσσα κατασκευής των Web Documents, της πληροφορίας που ρέει στο Web.

ΚΕΦΑΛΑΙΟ 2

2 WEB SERVICES

2.1 ΟΡΙΣΜΟΣ ΤΩΝ WEB SERVICES

Τα web services εμφανίστηκαν το 2001 όταν άρχισε να γίνεται λόγος για αυτο-περιγραφόμενες και αυτο-προσδιοριζόμενες εφαρμογές οι οποίες μπορούν να ανακαλυφθούν και να προσπελαστούν μέσω διαδικτύου από άλλες εφαρμογές. Οι υπηρεσίες WEB κατάφεραν όχι μόνο να παραμείνουν στο προσκήνιο των εξελίξεων στον χώρο της τεχνολογίας όλη τη δεκαετία, αλλά να αποτελέσουν ένα από τα πιο σημαντικά κεφάλαια της υψηλής τεχνολογίας του Διαδικτύου.

Web service είναι οποιοδήποτε τμήμα λογισμικού ή συστατικό εφαρμογής που είναι διαθέσιμο μέσω διαδικτύου και το οποίο μπορεί να ολοκληρώνει εργασίες και να λύνει προβλήματα. Ωστόσο αν θέλουμε να προσδιορίσουμε πιο σωστά την έννοια των web services, θα μπορούσαμε να χρησιμοποιήσουμε τον ορισμό που έρχεται από την IBM :

- Τα web services είναι μια τεχνολογία που επιτρέπει στις εφαρμογές να επικοινωνούν μεταξύ τους ανεξαρτήτως πλατφόρμας και γλώσσας προγραμματισμού. Ένα web service αποτελεί μια διεπαφή λογισμικού (software interface) που περιγράφει μια συλλογή από λειτουργίες οι οποίες μπορούν να προσεγγιστούν από το δίκτυο μέσω πρότυπων μηνυμάτων XML. Χρησιμοποιεί πρότυπα βασισμένα στη γλώσσα XML για να περιγράψει μια λειτουργία προς εκτέλεση και τα δεδομένα προς ανταλλαγή με κάποια άλλη εφαρμογή. Μια ομάδα από web services οι οποίες αλληλεπιδρούν μεταξύ τους καθορίζει μια εφαρμογή web services.

Με απλά λόγια μια Web service μπορούμε να την αντιληφθούμε σαν ένα λογισμικό που γνωρίζει με ποιόν τρόπο πρέπει να 'μιλήσει' σε διαφορετικού τύπου λογισμικά μέσω δικτύου. Πιο συγκεκριμένα μια υπηρεσία Web πραγματοποιεί τις ακόλουθες λειτουργίες.

- Παρέχει την δυνατότητα σε άλλες εφαρμογές να κατανοήσουν την λειτουργικότητά της, εκθέτοντας και περιγράφοντας την παρουσία της σε αυτές με τη χρήση ενός πρότυπου δικτυακού πρωτοκόλλου.
- Μπορεί να γίνει εύκολα αντιληπτή από άλλες εφαρμογές μέσω ενός καταλόγου στον οποίο βρίσκετε καταχωρημένη.
- Μπορεί να προβληθεί ή να χρησιμοποιηθεί από την εφαρμογή που την χρειάζεται μέσω απλών πρωτοκόλλων.

Στα κεφάλαια που θα ακολουθήσουν θα γίνει μια εκτενής αναφορά στις παραπάνω λειτουργίες, ενώ ταυτόχρονα θα δοθεί ιδιαίτερη έμφαση στα πρωτόκολλα που συναντάμε και χωρίς την παρουσία των οποίων δεν θα μπορούσαμε σήμερα να

μιλάμε για την ύπαρξη των web services. Η δυνατότητα που δίνεται σε οποιαδήποτε σύγχρονη επιχείρηση να παρέχει on line συναλλαγές με τους προμηθευτές της αλλά και με τους πελάτες της μέσω της συγκεκριμένης υπηρεσίας, είναι μόλις μια από τις πολλές εφαρμογές της υπηρεσίας Web. Δεν είναι τυχαία άλλωστε η ραγδαία εξάπλωση της παρά τις αρνητικές πιθανότητες των αρχών της δεκαετίας του 2000, όπου πολλές άλλες προσανατολισμένες εφαρμογές της τεχνολογίας προς το Διαδίκτυο, εξαφανίστηκαν σχεδόν όσο γρήγορα εμφανίστηκαν.

2.2 Η ΔΥΝΑΜΗ ΤΩΝ WEB SERVICES

Οι web services χαρακτηρίζονται από τη ανεξάρτητη και ουδέτερη φύση τους. Σε αντίθεση με τις παλαιότερες εφαρμογές λογισμικού που βασίζονταν σε κλειστά πρότυπα, οι υπηρεσίες Web βασίζονται σε ευρέως αποδεκτά πρότυπα του κλάδου. Με την χρήση μιας γλώσσας ανοιχτού προτύπου, την XML, σαν το μόνο πρότυπο στα web services, το πρόβλημα της διαλειτουργικότητας από το οποίο έπασχαν όλες οι προηγούμενες τεχνολογίες έπαψε να υφίσταται. Η XML ερμηνεύεται από οποιαδήποτε γλώσσα προγραμματισμού, καθιστώντας έτσι τις υπηρεσίες Web ουδέτερες και τις υπηρεσίες που τις υιοθετούν να μην χρειάζεται να ανησυχούν για θέματα συμβατότητας μεταξύ λειτουργικών συστημάτων.

Ταυτόχρονα, η ευκολία επικοινωνίας μεταξύ συμβατών και ασυμβατών συστημάτων μεταξύ τους εξαιτίας της XML, μας οδήγησε σε εφαρμογές με χαλαρή συνδεσιμότητα, ευελιξία και προσαρμοστικότητα. Σε τεχνικό επίπεδο τα web services χειρίζονται τα δεδομένα με πολύ πιο εύκολο τρόπο από παλαιότερες τεχνολογίες καθώς λειτουργούν σε ένα πιο αφηρημένο επίπεδο. Επιπλέον οι δυναμικές συνεργασίες μεταξύ συστημάτων και εταιρών είναι πλέον γεγονός. Οι περισσότερες τεχνολογίες χρησιμοποιούσαν μη πρότυπες πόρτες, με αποτέλεσμα η συνεργασία μεταξύ εταιρειών να απαιτεί άνοιγμα οπών στα τείχη προστασίας, κάτι το οποίο ήταν μη αποδεκτό από τους ανθρώπους της πληροφορικής. Η δυνατότητα ωστόσο των web services να χρησιμοποιούν πέρα άλλων και το πρωτόκολλο μεταφοράς HTTP, μετέτρεψε αυτό το πρόβλημα σε παρελθόν. Τα περισσότερα τείχη προστασίας επιτρέπουν την πρόσβαση του HTTP μέσω συγκεκριμένης θύρας και έτσι οδηγηθήκαμε σε ευκολότερες συνεργασίες μεταξύ εταιρειών.

Η απλότητα υποδομής είναι ένας ακόμα λόγος που καθιστά τις web services τόσο σημαντικές και ευρέως διαδεδομένες. Τα web services λειτουργούν με πολύ πιο απλά πρωτόκολλα επικοινωνίας (SOAP), τα πρωτόκολλα HTTP-TCP/IP, καθώς όπως προαναφέρθηκε χρησιμοποιούν την γλώσσα ανοιχτού προτύπου XML. Αυτά τα πρότυπα με την απλότητα τους οδηγούν σε ταχύτερη ανάπτυξη εφαρμογών με πολύ πιο χαμηλά κόστη. Η πλειονότητα των εταιρειών διέθεταν στην ουσία ήδη την δικτυακή υποδομή αλλά και την απαραίτητη γνώση, ώστε να μπορέσουν να υποστηρίξουν άμεσα την καινούργια αυτή μετάβαση στον 'κόσμο' των web services. Η προτυποποίηση, η ανοιχτή αρχιτεκτονική, η ευκολία επικοινωνίας συμβατών και ασυμβατών συστημάτων μεταξύ τους καθώς και η μεγάλη εξοικονόμηση κόστους λόγω των παραπάνω, καθιστούν τις web services ικανές να βοηθήσουν στην βελτίωση της αποτελεσματικότητας επιτρέποντας στις επιχειρήσεις και όχι μόνο να ρυθμίζουν τις διαδικασίες μεταξύ τους και των πελατών τους..

2.3 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΩΝ WEB SERVICES (SOA)

Κάθε υπηρεσία για να μπορεί να λειτουργήσει αποδοτικά, αλλά ταυτόχρονα και να εξελιχθεί πρέπει να βασίζεται σε κάποια συγκεκριμένη δομή. Να ακολουθεί ένα συγκεκριμένο μοντέλο πάνω στο οποίο θα αναπτύσσονται όλες οι λειτουργίες και τα πρωτόκολλα της αντίστοιχης υπηρεσίας. Το μοντέλο που χρησιμοποιείται στις μέρες μας για τις Υπηρεσίες Διαδικτύου είναι το λεγόμενο SOA-Service Oriented Architecture. Αντιπροσωπεύει μια αρχιτεκτονική προσανατολισμένη σε υπηρεσίες. Σε αυτό το μοντέλο συναντάμε ξεχωριστές δομικές μονάδες, οι οποίες μπορούν να διανεμηθούν πάνω από ένα δίκτυο, να συνδυαστούν και να ξαναχρησιμοποιηθούν έτσι ώστε να δημιουργήσουν επιχειρησιακές εφαρμογές. Στη συνέχεια γίνεται μια ανάλυση στις δομικές μονάδες της συγκεκριμένης αρχιτεκτονικής καθώς και στις λειτουργίες που εκτελεί κάθε δομική μονάδα είτε ανεξάρτητα είτε σε συνδυασμό με κάποια άλλη.

- **Service Provider (Πάροχος Υπηρεσίας):**

Αποτελεί την οντότητα, η οποία συντάσσει την αντίστοιχη υπηρεσία στο Διαδίκτυο. Είναι υπεύθυνος για την επιλογή των εφαρμογών που θα παρέχονται στους χρήστες της συγκεκριμένης web service, το επίπεδο ασφάλειας και προσβασιμότητας καθώς και τον τρόπο κοστολόγησης. Η πιο σημαντική αρμοδιότητα του Παρόχου Υπηρεσίας ωστόσο είναι να δημιουργήσει τις κατάλληλες προϋποθέσεις ώστε η συγκεκριμένη υπηρεσία να είναι ορατή από όλους τους χρήστες του Διαδικτύου. Για την συγκεκριμένη αρμοδιότητα απαιτείται η συνεργασία του με μια άλλη δομική μονάδα, την Service Registry (Υπηρεσία Καταλόγου). Ο Πάροχος υπηρεσίας περιγράφει τη υπηρεσία του ενώ παρέχει και κάποιες πληροφορίες πρόσβασης σε αυτή, στον κατάλογο υπηρεσίας. Ταυτόχρονα κατηγοριοποιεί την υπηρεσία μέσα στον κατάλογο και ορίζει τι είδους συμφωνίες απαιτούνται για να μπορεί κανείς να προσπελάσει την συγκεκριμένη λειτουργία.

- **Service Registry(Υπηρεσία Καταλόγου):**

Η υπηρεσία καταλόγου αποτελεί μια σύγχρονη <<Βιβλιοθήκη>>. Αρμοδιότητα της είναι η καταχώρηση μαζί με τις αντίστοιχες περιγραφές όλων των υπαρχόντων υπηρεσιών Διαδικτύου. Ο υπεύθυνος του κάθε καταλόγου αποφασίζει για το φάσμα χρηστών στους οποίους θα είναι ορατός ο κατάλογος, τη ποσότητα των καταχωρήσεων που θα υποστηρίζει καθώς και το επίπεδο ασφαλείας που θα παρέχει. Υπάρχουν δημόσιοι αλλά και ιδιωτικοί κατάλογοι, μερικοί καλύπτουν ένα ευρύ φάσμα υπηρεσιών ενώ άλλοι ειδικεύονται σε καταχωρήσεις συγκεκριμένων κλάδων. Η συγκεκριμένη μονάδα αποτελεί στην ουσία τον συνδετικό κρίκο ανάμεσα στον Πάροχο και στον Χρήστη Υπηρεσίας. Εδώ δημοσιοποιείται η Περιγραφή της εκάστοτε Υπηρεσίας από τον Πάροχο, η οποία ταυτόχρονα αποτελεί το αποτέλεσμα της αναζήτησης του χρήστη στον κατάλογο.

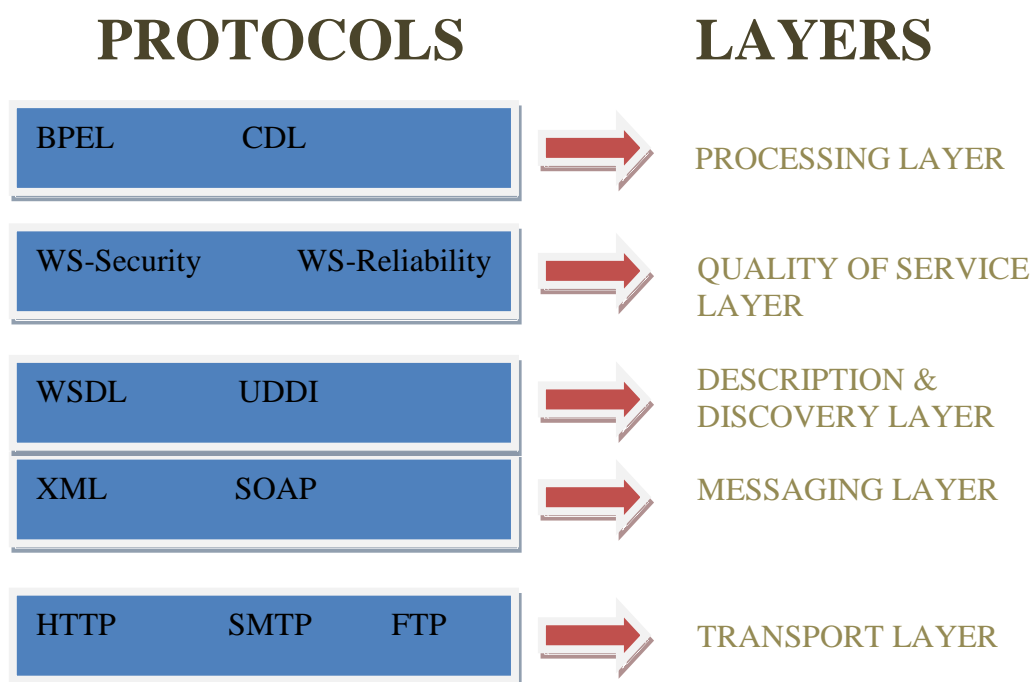
- **Service Requestor(Αιτούμενος Υπηρεσίας):**

Αποτελεί τον αιτούμενο της κάθε υπηρεσίας. Αρχικά αναζητά καταχωρήσεις μέσα από έναν αριθμό καταλόγων αντλώντας την περιγραφή της κάθε υπηρεσίας και καταλήγει στον εντοπισμό της ζητούμενης καταχώρησης καθώς και της διεύθυνσης της. Εδώ λήγει και ο ρόλος της ενδιάμεσης δομικής μονάδας. Ο χρήστης συνδέεται με τον πάροχο και καλεί το web service που εκείνος δημοσιεύει.

Σε αυτό το σημείο θα πρέπει να σημειωθεί ότι οι αρμοδιότητες των παραπάνω δομικών μονάδων δεν είναι καθορισμένες. Ο χρήστης ανά πάσα στιγμή μπορεί να γίνει πάροχος και αντίστροφα. Σε αυτό το σημείο λοιπόν μπορούμε να πούμε ότι έχουμε καλύψει τις βασικές λειτουργίες που εκτελεί κάθε δομική μονάδα ξεχωριστά και σε συνεργασία. Ωστόσο μιλάμε για υπολογιστικές εφαρμογές. Για να πραγματοποιηθούν όλες οι παραπάνω λειτουργίες χρησιμοποιούνται συγκεκριμένα πρότυπα, τα οποία είναι ανεξάρτητα από λειτουργικές πλατφόρμες και γλώσσες προγραμματισμού. Στο κεφάλαιο που ακολουθεί γίνεται μια αναφορά στα συγκεκριμένα πρότυπα.

2.4 ΤΕΧΝΟΛΟΓΙΕΣ ΤΩΝ WEB SERVICES

Η αρχιτεκτονική που αναλύθηκε πιο πάνω ακολουθεί με απλά λόγια το μοντέλο δημοσίευση, εύρεση και σύνδεση. Ο πάροχος της υπηρεσίας δημοσιεύει την υπηρεσία στον κατάλογο υπηρεσιών. Ο χρήστης ψάχνει μέσα από τον κατάλογο την υπηρεσία η οποία καλύπτει τις ανάγκες του και στην συνέχεια συνδέεται με αυτήν για να την εκτελέσει. Είναι προφανές λοιπόν ότι οι δομικές μονάδες της αρχιτεκτονικής SOA πρέπει να είναι σε θέση να βρίσκουν η μια την άλλη, να ανακαλύπτουν πληροφορίες που τις επιτρέπουν να αλληλοσυνδέονται, να καθορίζουν ποιες είναι οι αναμενόμενες μορφές αλληλοσύνδεσης και να διαπραγματεύονται ποιότητες υπηρεσίας όπως η ασφάλεια και η αξιόπιστη επικοινωνία. Για να πραγματοποιηθούν όλες οι παραπάνω λειτουργίες οι Web Services δεν χρησιμοποιούν συγκεκριμένη τεχνολογία, αλλά ένα σύνολο από από καθιερωμένα πρωτόκολλα επικοινωνίας πολλών επιπέδων, που περιλαμβάνουν το HTTP, την XML, το SOAP, το UDDI και το WSDL. Σχηματικά μπορούμε να το φανταστούμε σαν μια κατανομή επιπέδων, όπου η πληροφορία περνάει από το ένα στρώμα στο άλλο, αποκρύπτοντας τις πολλές λεπτομέρειες που δε χρειάζονται στα άλλα στρώματα.



Σχ.1 Στοιβά Πρωτοκόλλων των Web Services

Πιο αναλυτικά έχουμε τα εξής επίπεδα:

- TRANSPORT LAYER(ΕΠΙΠΕΔΟ ΜΕΤΑΦΟΡΑΣ)

Σε αυτό το επίπεδο συναντάμε την υπηρεσία μεταφοράς, υπεύθυνη για την μεταφορά των διάφορων δεδομένων από τον πάροχο στον χρήστη και αντίστροφα. Εδώ περιλαμβάνονται κάποια από τα πιο διαδεδομένα πρωτόκολλα μεταφοράς όπως το HTTP, το SMTP καθώς και το FTP. Τα συγκεκριμένα πρωτόκολλα τα συναντάμε όχι μόνο στις Υπηρεσίες Web αλλά και σε πολλές άλλες υπηρεσίες που διατίθενται μέσω Internet καθώς αποτελούν κάποια από τα πιο εύχρηστα πρωτόκολλα μεταφοράς χρησιμοποιώντας πόρτες μη περιορισμένες από firewalls.

- MESSAGING LAYER(ΕΠΙΠΕΔΟ ΜΗΝΥΜΑΤΩΝ)

Για να είναι δυνατή η μεταφορά δεδομένων σε ένα συγκεκριμένο υπολογιστικό περιβάλλον πρέπει να χρησιμοποιούνται συγκεκριμένα πρωτόκολλα καθορισμού του τρόπου μορφοποίησης και αναπαράστασης των δεδομένων ώστε στη συνέχεια να μπορούν να μεταβούν στο επίπεδο μεταφοράς. Στο συγκεκριμένο επίπεδο συναντάμε την υπηρεσία μηνυμάτων, η οποία βασισμένη σε συγκεκριμένα πρωτόκολλα, παρέχει το υπόβαθρο στις Υπηρεσίες Web για μεταφορά πληροφορίας. Πιο συγκεκριμένα συναντάμε την γλώσσα σήμανσης XML, καθώς και το SOAP – Πρωτόκολλο Απλής Προσπέλασης Αντικειμένου. Η XML αποτελεί μια περιγραφική γλώσσα με καλά καθορισμένη σύνταξη και σημασιολογία παρέχοντας μια πρότυπη και κοινή δομή για την διανομή των μηνυμάτων. Το SOAP από την πλευρά του

αποτελεί το πλαίσιο της επικοινωνίας μεταξύ μιας εφαρμογής-προμηθευτή web service και μιας εφαρμογής πελάτη σύμφωνα με τη γλώσσα XML.

- DESCRIPTION & DISCOVERY LAYER
(ΕΠΙΠΕΔΟ ΠΕΡΙΓΡΑΦΗΣ & ΑΝΑΚΑΛΥΨΗΣ)

Για να έχουν ισχύ οι Υπηρεσίες Web θα πρέπει ο κάθε χρήστης να είναι σε θέση να αναζητάει από ένα κατάλογο όλες τις υπάρχουσες web services και με κάποια κριτήρια να συνδέεται με τον πάροχο της Υπηρεσίας που τον ενδιαφέρει. Από το στάδιο αυτό και έπειτα αναλαμβάνουν τα προαναφερθέντα επίπεδα. Τα πρωτόκολλα που συναντάμε σε αυτό το επίπεδο αφορούν την πραγματοποίηση του Service Registry (δες Υποκεφάλαιο.2.3). Δυο βασικά πρωτόκολλα συναντάμε εδώ, τη περιγραφική γλώσσα WSDL και το UDDI. Η WSDL (Web Services Description Language) χρησιμοποιώντας ένα σύνολο ετικετών (tags) περιγράφει ένα web service, τις υπηρεσίες που παρέχονται, την διεύθυνση του κλπ. Το UDDI από την πλευρά του αποτελεί τον <<Χρυσό Οδηγό>> των web services. Όλες οι εφαρμογές που παρέχουν web services παρατίθενται σε έναν κατάλογο από παρόχους υπηρεσιών χρησιμοποιώντας το UDDI.

- QUALITY OF SERVICE LAYER
(ΕΠΙΠΕΔΟ ΠΟΙΟΤΗΤΑΣ)

Βασικά στοιχεία που πρέπει να εξασφαλίζονται σε κάθε Υπηρεσία Web πρέπει να είναι η διαθεσιμότητα, η αξιοπιστία και η ασφάλεια. Τα πρότυπα που συναντάμε σε αυτό το επίπεδο είναι το WS-Security και το WS-Reliability υπεύθυνα για την εξασφάλιση του απαραίτητου QoS σε μια Web Service.

- PROCESSES LAYER (ΕΠΙΠΕΔΟ ΔΙΕΡΓΑΣΙΩΝ)

Πρόκειται για το επίπεδο που είναι υπεύθυνο για το συνδυασμό ήδη υλοποιημένων υπηρεσιών ώστε να προκύψουν νέες σύνθετες υπηρεσίες. Επιτρέπει στους χρήστες να περιγράφουν αλλά και να υλοποιούν δραστηριότητες μέσω web services ορίζοντας τον τρόπο με τον οποίο θα συνεργαστούν για να πραγματοποιήσουν διάφορες λειτουργίες. Στο συγκεκριμένο επίπεδο συναντάμε τα εξής δυο πρωτόκολλα. Το BPEL (Business Process Execution Language), το οποίο συνθέτει τις νέες υπηρεσίες χρησιμοποιώντας τις ήδη υπάρχουσες. Στην ουσία αποτελεί μια γλώσσα <<ενορχήστρωσης>> που καθορίζει τον ρόλο της κάθε υπηρεσίας στην νέα σύνθεση. Το CDL (Choreography Description Language) από την άλλη πλευρά αποτελεί μια choreography γλώσσα ,η οποία καθορίζει την επικοινωνία ανάμεσα στις υπηρεσίες .

Στα επόμενα κεφάλαια θα ακολουθήσει μια πιο αναλυτική περιγραφή των σημαντικότερων πρωτοκόλλων που παρουσιάστηκαν στο παρόν κεφάλαιο και χωρίς την παρουσία των οποίων δεν θα μπορούσαμε να μιλάμε σήμερα για την ύπαρξη και εξέλιξη μιας από τις πιο σημαντικές εφαρμογές του Διαδικτύου, τις Web Services.

ΚΕΦΑΛΑΙΟ 3

3 EXTENSIBLE MARKUP LANGUAGE - XML

Η HTML αποτελεί την βασική γλώσσα κατασκευής των Web Documents, της πληροφορίας που ρέει στο Web. Στο χώρο του Διαδικτύου υπάρχουν συνολικά πάνω από 1 δισεκατομμύριο ιστοσελίδες βασισμένες στην συγκεκριμένη γλώσσα. Υποστηρίζεται από χιλιάδες εφαρμογές συμπεριλαμβανομένων των πλοηγών Διαδικτύου, των επεξεργαστών κειμένου κλπ. Είναι γλώσσα προγραμματισμού που λέει στον web browser τι πληροφορίες να προβάλει, πώς να εντοπίσει αρχεία πολυμέσων και πώς να τα εμφανίσει στην ιστοσελίδα.

Ωστόσο αν και η συγκεκριμένη γλώσσα είναι μια επιτυχημένη γλώσσα σήμανσης έχει κάποια σημαντικά μειονεκτήματα, τα οποία καλείται να ξεπεράσει μια άλλη γλώσσα σήμανσης, η XML. Η συγκεκριμένη γλώσσα δεν σχεδιάστηκε με σκοπό να εξαλείψει την HTML, αλλά για να υποστηρίξει καλύτερα τη δημιουργία και τη διαχείριση δυναμικού περιεχομένου. Επιπλέον δίνει στα έγγραφα ένα μεγαλύτερο επίπεδο προσαρμοστικότητας στη μορφή και τη δομή από αυτό που παρέχει η HTML. Πιο συγκεκριμένα η HTML εστιάζει στην παρουσίαση και εμφάνιση των δεδομένων. Αντίθετα η XML σχεδιάστηκε για να μεταφέρει και να αποθηκεύει τα δεδομένα δίνοντας έμφαση στο τι δεδομένα είναι, ενώ τα περιγράφει σε μια τυποποιημένη μορφή που μπορεί εύκολα να μεταφερθεί με την χρήση πρωτοκόλλων διαδικτύου (HTTP).

3.1 ΕΙΣΑΓΩΓΗ ΣΤΗΝ XML

Η XML είναι μια markup γλώσσα για έγγραφα που περιέχουν δομημένες πληροφορίες, τα λεγόμενα XML έγγραφα. Με τον όρο <<markup γλώσσα>> αναφερόμαστε σε ένα μηχανισμό που καθορίζει δομές σε ένα έγγραφο, δηλαδή το περιεχόμενο και κάποιες διευκρινίσεις για το ρόλο του περιεχομένου. Στην ουσία δηλαδή περιγράφει μια συγκεκριμένη κατηγορία αντικειμένων δεδομένων, τα XML έγγραφα και εν μέρει την συμπεριφορά των προγραμμάτων υπολογιστών που τα επεξεργάζονται.

Η γλώσσα XML αναπτύχθηκε από μια Ομάδα Εργασίας κάτω από την επίβλεψη του διεθνούς οργανισμού World Wide Web Consortium (W3C) το 1996. Εδραιώθηκε από τον John Bosak της Sun Microsystems και αποτελεί μια μεταφερτή, ευρέως υποστηριζόμενη, ανοιχτή τεχνολογία για την περιγραφή δεδομένων. Χάρη στα ακόλουθα χαρακτηριστικά, όλο και περισσότερες εφαρμογές συμπεριλαμβανομένων των web services χρησιμοποιούν την συγκεκριμένη τεχνολογία καθιστώντας την πρότυπο για την αποθήκευση δεδομένων που ανταλλάσσονται μεταξύ εφαρμογών.

- Είναι επεκτάσιμη και ανεξάρτητη από πλατφόρμες, γεγονός που την καθιστά απρόσβλητη σε τεχνολογικές αλλαγές.
- Εξαιτίας της κειμενικής φύσης τους τα XML έγγραφα δεν εξαρτώνται από την πλατφόρμα του συστήματος.
- Το «ωφέλιμο φορτίο» δεδομένων είναι εμφωλευμένο σε ετικέτες και επομένως αναγνώσιμο από τους χρήστες.
- Η δομή είναι καθορισμένη με σαφήνεια και μπορεί να περάσει μεταξύ διαφορετικών υπολογιστικών συστημάτων, που ειδάλλως θα ήταν ανίκανα να επικοινωνήσουν.

Συμπερασματικά λοιπόν η XML σχεδιάστηκε δίνοντας έμφαση στην απλότητα, την γενικότητα και την χρησιμότητα στο Διαδίκτυο. Είναι μια μορφοποίηση δεδομένων κειμένου, με ισχυρή υποστήριξη του προτύπου Unicode, το οποίο επιτρέπει την αναπαράσταση περιεχομένου από πολλές φυσικές γλώσσες. Αν και η σχεδίαση της XML εστιάζει στα κείμενα, χρησιμοποιείται ευρέως για την αναπαράσταση αυθαίρετων δομών δεδομένων που προκύπτουν για παράδειγμα στις Υπηρεσίες Διαδικτύου (Web Services).

3.2 ΒΑΣΙΚΗ ΟΡΟΛΟΓΙΑ ΤΗΣ XML

Στο παρόν κεφάλαιο παραθέτουμε τους βασικούς όρους που συναντάμε στην απλή καθημερινή χρήση της γλώσσας XML. Δεν αποτελεί μια πλήρη λίστα όλων των στοιχείων που υπάρχουν στην γλώσσα, αλλά ενδεικτική.

- **Χαρακτήρας Unicode :**

Όπως προαναφέρθηκε η γλώσσα XML βασίζεται στο πρότυπο Unicode. Επομένως κάθε κείμενο XML είναι μια ακολουθία χαρακτήρων Unicode. Ελάχιστοι χαρακτήρες Unicode δεν εμφανίζονται στα XML έγγραφα.

- **Σήμανση και Περιεχόμενο:**

Σε κάθε XML έγγραφο που συναντάμε, κάποιοι χαρακτήρες αποτελούν τη σήμανση και οι άλλοι το περιεχόμενο του εγγράφου. Πιο συγκεκριμένα όλα τα αλφαριθμητικά που συνιστούν τη σήμανση είτε ξεκινούν με τον χαρακτήρα " < " ή τον " & " και καταλήγουν στο χαρακτήρα " > " ή αντίστοιχα στο " ; ". Ακολουθίες χαρακτήρων που δεν αποτελούν την σήμανση του εγγράφου έχουν την ιδιότητα του Περιεχομένου.

- **Ετικέτα(Tag):**

Ένα στοιχείο σήμανσης που ξεκινά με τον χαρακτήρα " < " και καταλήγει στον " > ", είναι ετικέτα. Υπάρχουν τρεις κατηγορίες ετικέτας. Η ετικέτα ανοίγματος (π.χ <number>), η ετικέτα κλεισίματος (π.χ < /number>) και ετικέτες χωρίς περιεχόμενο (π.χ < line-break/>). Όλες οι ετικέτες είναι case sensitive δηλαδή υπάρχει διαχωριστική γραμμή μεταξύ κεφαλαίων και πεζών και τα ονόματα τους υπακούουν σε κανόνες ονοματολογίας. Σε αυτό το σημείο πρέπει να σημειωθεί ότι η XML επιτρέπει στο συγγραφέα ενός XML εγγράφου να ορίζει τα δικά του tags. Για παράδειγμα τα tags <name>, <address> που μπορεί να συναντήσει κανείς σε ένα

XML που περιγράφει μια διεύθυνση δεν είναι ορισμένα σε κανένα XML Standard. Αυτά τα tags επινοήθηκαν από το συγγραφέα του συγκεκριμένου XML εγγράφου.

- **Στοιχεία(Elements) :**

Κάθε λογικό απόσπασμα ενός κειμένου είναι το στοιχείο, το οποίο είτε ξεκινά από μια ετικέτα ανοίγματος και καταλήγει σε μια ετικέτα κλεισίματος είτε αποτελείται από μια ετικέτα χωρίς περιεχόμενο. Όλοι οι χαρακτήρες που υπάρχουν ανάμεσα σε μια ετικέτα ανοίγματος και σε μια ετικέτα κλεισίματος αποτελούν το περιεχόμενο του στοιχείου το οποίο μπορεί να περιέχει και σήμανση. Στην συνέχεια παρουσιάζεται ένα παράδειγμα στοιχείου όπου η πρώτη γραμμή περιέχει μια ετικέτα ανοίγματος και η δεύτερη γραμμή την ετικέτα κλεισίματος.

```
<Item optional="1">
```

```
</Item>
```

Όλα τα στοιχεία δομούνται σε μορφή δέντρου δηλαδή οργανώνονται σε ένα ιεραρχικό τρόπο με ένα στοιχείο-πατέρα και στοιχεία-παιδιά. Τα στοιχεία -παιδιά περιέχουν και αυτά άλλα στοιχεία-παιδιά κλπ. Συγκεκριμένα κάθε έγγραφο περιλαμβάνει το στοιχείο ρίζας που περιλαμβάνει κάθε άλλο στοιχείο και είναι το πρώτο μετά την δήλωση XML. Τα στοιχεία δηλαδή εντίθενται το ένα μέσα στο άλλο σχηματίζοντας ιεραρχίες. Κάθε στοιχείο μπορεί να περιέχει δεδομένα, ιδιότητες είτε ταυτόχρονα δεδομένα και ιδιότητες μαζί, ενώ δεν νοείται στοιχείο χωρίς ετικέτα κλεισίματος. Τέλος πρέπει να σημειωθεί ότι κάθε στοιχείο μπορεί να περιέχει ακολουθίες χαρακτήρων Unicode. Έτσι η XML μπορεί να αποθηκεύει χαρακτήρες από σχεδόν οποιαδήποτε γλώσσα

- **Ιδιότητες (Attributes) :**

Σε κάθε έγγραφο είναι πολύ βοηθητικό τα δεδομένα να είναι αυτοπεριγραφικά. Οι ιδιότητες μας βοηθούν να περιγράψουμε τα στοιχεία μας πιο αποτελεσματικά και με σαφήνεια. Τα ονόματα των ιδιοτήτων ακολουθούν τους ίδιους κανόνες με αυτούς των ετικετών ενώ οι τιμές τους εσωκλείονται σε " " ή σε ' '. Κάθε ιδιότητα αποτυπώνεται ως ζεύγος ονόματος – τιμής στις ετικέτες ανοίγματος. Στο προηγούμενο παράδειγμα η optional αποτελεί το όνομα ιδιότητας ενώ το "1" την τιμή.

- **Δήλωση XML:**

Τα έγγραφα μπορούν να αρχίζουν, με την δήλωση κάποιων πληροφοριών σχετικά με αυτά.

Ακολουθεί ένα συνοπτικό παράδειγμα ενός XML εγγράφου που υπακούει στα προαναφερθέντα χαρακτηριστικά.

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<painting> 
```

```
<caption>This is Raphael's "Foligno" Madonna, painted in
```

```
<date>1511</date>-<date>1512</date>.</caption>
```

```
</painting>
```

Στο συγκεκριμένο παράδειγμα η πρώτη γραμμή αποτελεί την δήλωση XML, όπου μας δίνεται πληροφορία για την έκδοση της XML και την κωδικοποίηση που χρησιμοποιείται. Υπάρχουν πέντε στοιχεία: τα painting, img, caption, και δυο date. Τα στοιχεία date είναι παιδιά του στοιχείου caption, το οποίο είναι παιδί του στοιχείου painting. Είναι εμφανές ότι το στοιχείο painting αποτελεί το στοιχείο ρίζα.

Συνοπτικά λοιπόν κάθε XML έγγραφο πρέπει να διαθέτει ένα στοιχείο-ρίζα, όλα τα στοιχεία που περιέχονται πρέπει να εσωκλείονται σωστά, όλα τα tags πρέπει να είναι case sensitive, ενώ επιβάλεται η ύπαρξη μιας ετικέτας ανοίγματος και μιας ετικέτας κλεισίματος. Σε αυτό το σημείο πρέπει να σημειωθεί ότι τα προαναφερθέντα χαρακτηριστικά αποτελούν κάποια από τα κύρια χαρακτηριστικά ενός καλοσχηματισμένου XML εγγράφου. Παράλληλα ωστόσο, κάθε XML έγγραφο θα πρέπει να υπακούει στους κανόνες του DTD ή του XML Schema για να θεωρηθεί καλά δομημένο και έγκυρο. Στην συνέχεια υπάρχει μια σχετική αναφορά στην προδιαγραφή DTD καθώς και στο XML Schema, το οποίο αποτελεί μια προηγούμενη έκδοση του DTD.

3.3 DTD-DOCUMENT TYPE DEFINITION

Το Document Type Definition αποτελεί την βασική προδιαγραφή, η οποία πρέπει να ακολουθείται κατά την δημιουργία ενός XML εγγράφου. Ουσιαστικά το DTD παρέχει ένα αυστηρό πλαίσιο και κανόνες καθορίζοντας έτσι την επιτρεπτή δομή ενός έγκυρου XML εγγράφου. Επιπρόσθετα χρησιμοποιείται για τον έλεγχο της εγκυρότητας και της ακεραιότητας των δεδομένων που περιέχονται σε κείμενα XML(δες 3.3.1).

Πιο συγκεκριμένα το DTD ορίζει κανόνες που αφορούν στα tags της γλώσσας XML. Καθορίζει τα επιτρεπτά ονόματα στοιχείων και τα επιτρεπτά υποστοιχεία και χαρακτηριστικά για κάθε συγκεκριμένο στοιχείο. Τα στοιχεία που χρησιμοποιεί το DTD για όλες τις παραπάνω λειτουργίες είναι τα εξής :

- **DTD Element:** Καθορίζει τι είδους δεδομένα θα έχει το στοιχείο, τον αριθμό των περιστατικών κάθε στοιχείου, τις σχέσεις μεταξύ των στοιχείων κλπ.
- **DTD Attributes:** Καθορίζει διάφορους ορισμούς που σχετίζονται με τα δεδομένα.
- **DTD Entities:** Χρησιμοποιείται για την αναφορά ενός εξωτερικού αρχείου ή για να παρέχει συντομεύσεις σε κοινό κείμενο.

ΠΑΡΑΔΕΙΓΜΑ

<!ELEMENT employee(shift+,home-address,hobbies*)>

Στο συγκεκριμένο παράδειγμα δίνονται οι εξής πληροφορίες: Ένα στοιχείο employee μπορεί να περιέχει ένα ή περισσότερα στοιχεία shift και πρέπει να έχει ένα στοιχείο home-address και μηδέν ή και περισσότερα στοιχεία hobbies.

Πιο συγκεκριμένα κάποιες βασικές παρατηρήσεις σχετικά με τη σύνταξη των εγγράφων DTD είναι:

- Όλα τα ονόματα των αντικειμένων ξεκινούν με γράμμα και δεν επιτρέπεται ο χαρακτήρας κενού διαστήματος, αλλά στην θέση του τοποθετούμε το ενωτικό.
- Η XML υποστηρίζει το πρότυπο κωδικοποίησης χαρακτήρων Unicode. Συνεπώς μπορούμε να χρησιμοποιούμε ελληνικούς χαρακτήρες ως έγκυρους.
- Οι χαρακτήρες *, ?, + ονομάζονται χαρακτήρες ύπαρξης και κάποιους από αυτούς συναντήσαμε στο παράδειγμα μας. Στοιχείο με τον χαρακτήρα + εμφανίζεται μια τουλάχιστον φορά. Με τον χαρακτήρα * καμία ή περισσότερες ενώ με τον ? το στοιχείο εμφανίζεται το πολύ μια φορά. Η μη ύπαρξη δείκτη ύπαρξης σε ένα στοιχείο υποδεικνύει την εμφάνιση του μια και μόνο φορά.
- Οι χαρακτήρες , και | ονομάζονται συζευκτήρες. Το κόμμα υποδεικνύει ότι τα δυο στοιχεία αριστερά και δεξιά του κόμματος εμφανίζονται με την ίδια σειρά στο έγγραφο ενώ ο χαρακτήρας | σημαίνει ότι μόνο ένα από τα στοιχεία εμφανίζεται στο έγγραφο XML.

Όπως έχουμε ήδη αναφέρει κάθε XML έγγραφο για να θεωρείται έγκυρο πρέπει να έχει ένα δικό του DTD που να το συνοδεύει. Αυτό το DTD είναι υπεύθυνο για τον καθορισμό της δομής του συγκεκριμένου εγγράφου ενώ ταυτόχρονα διευκρινίζει τις λεπτομέρειες σχετικά με όλα τα στοιχεία και τις ιδιότητες που χρησιμοποιούνται στο έγγραφο. Ωστόσο υπάρχουν και XML έγγραφα τα οποία ενώ υποχρεούνται να υπακούουν σε όλους τους κανόνες σύνταξης της γλώσσας XML, δεν διαθέτουν δικό τους συνοδευτικό DTD, είναι τα λεγόμενα Well-formed έγγραφα.

3.3.1 XML Parsers

Αν και όλα τα έγγραφα XML αποτελούν αρχεία κειμένου, η ανάκτηση δεδομένων μέσω των τεχνικών σειριακής προσπέλασης αρχείων δεν είναι αποτελεσματική ούτε αρκετά αποτελεσματική, ειδικά σε περιπτώσεις όπου τα δεδομένα πρέπει να προστεθούν ή να αφαιρεθούν δυναμικά. Για την επίλυση αυτού του προβλήματος υπάρχει ο Λεκτικός Αναλυτής-XML PARSER. Βασικό σκοπό έχει την αντιστοίχιση της δενδρικής δομής του κειμένου XML με την αντίστοιχη δομή στην χρησιμοποιούμενη γλώσσα. Ουσιαστικά αναλύει και ελέγχει κάθε έγγραφο XML, περιγράφοντας το κείμενο στα επιμέρους στοιχεία του όπως αρχή ετικέτας, τέλος ετικέτας, κείμενο κλπ.

3.4 XML SCHEMAS

Το DTD αποτέλεσε μια σημαντική προδιαγραφή δίχως την οποία δεν θα μπορούσαμε να μιλάμε για έγκυρα XML έγγραφα. Ωστόσο αποτελεί μια μη επεκτάσιμη προδιαγραφή, η οποία δεν υποστηρίζει ισχυρούς τύπους δεδομένων, ενώ διαθέτει σύνταξη διαφορετική από την XML. Το XML SCHEMA παρουσιάστηκε για να υπερνικήσει όλα τα παραπάνω μειονεκτήματα. Τα XML Schemas αποτελούν μια προηγμένη έκδοση του DTD και ουσιαστικά είναι έγγραφα XML με επέκταση Παπαργύρη Τριανταφυλλιά

αρχείου .xsd, τα οποία καθορίζουν πως πρέπει να δομούνται έγγραφα XML. Είναι επεκτάσιμα και μπορούν να υποστούν χειρισμούς, παραδείγματος χάρη να προστεθούν ή να διαγραφούν στοιχεία, όπως σε οποιοδήποτε άλλο έγγραφο XML.

Τα βασικότερα χαρακτηριστικά του XML Schema είναι τα εξής:

- Η σύνταξη είναι όμοια με αυτή της XML. Συμπερασματικά οποιοσδήποτε επεξεργαστής XML μπορεί να επεξεργαστεί το schema.
- Μας δίνεται η δυνατότητα πέρα των βασικών τύπων (ακέραιοι, πραγματικοί κλπ.) να καθορίζουμε δικούς μας τύπους δεδομένων, απλούς ή και σύνθετους οι οποίοι περιέχουν και άλλα στοιχεία.
- Ορίζει την σειρά με την οποία τα στοιχεία-παιδιά εμφανίζονται. Στο ακόλουθο παράδειγμα ορίζουμε έναν απλό τύπο "year" με τιμές μεταξύ 1900 και 2000 :

```
<xsd: simpleType name="year">  
  <xsd: restriction base="xsd: integer">  
    <xsd: minInclusive value="1900"/>  
    <xsd: maxInclusive value="2000"/>  
  </xsd: restriction>  
</xsd: simpleType>
```

- Το xml schema έχει την ιδιότητα της <<κληρονομικότητας>>. Δηλαδή μπορούμε να παράγουμε νέους τύπους δεδομένων βάσει παλαιών.
- Τέλος παρέχεται υποστήριξη για Namespaces (χρησιμοποιώντας URI). Παρέχεται δηλαδή σε κάθε στοιχείο ένα προσδιοριστικό, με το οποίο αποφεύγεται σύγκρουση ονομάτων μεταξύ των στοιχείων. Ένα παράδειγμα είναι στην περίπτωση συγχώνευσης δυο εγγράφων XML τα οποία περιέχουν και τα δυο από ένα στοιχείο με το όνομα "name", τα οποία έχουν διαφορετικό νόημα σε κάθε έγγραφο

ΚΕΦΑΛΑΙΟ 4

4 SOAP–SIMPLE OBJECT ACCESS PROTOCOL

Στα τέλη της δεκαετίας του 90, μεγάλες εταιρείες πληροφορικής όπως η Microsoft, άρχισαν να διερευνούν κατά πόσο ο κατακευματισμένος υπολογισμός μπορεί να βασιστεί στην XML. Ο σκοπός της έρευνας ήταν να γίνει εφικτή η επικοινωνία μεταξύ των εφαρμογών μέσω κλήσεων απομακρυσμένων διαδικασιών (Remote Procedure Calls-RPCs), χρησιμοποιώντας απλά πρωτόκολλα δικτύου, όπως το HTTP. Το 1999 έκανε την εμφάνιση του το Πρωτόκολλο Απλής Προσπέλασης Αντικειμένου (Simple Object Access Protocol, SOAP), ένας RPC μηχανισμός βασισμένος σε XML.

Το SOAP αποτελεί μια προδιαγραφή πρωτοκόλλου, που ορίζει με σαφήνεια την ανταλλαγή δομημένων πληροφοριών σε ένα κατακευματισμένο υπολογιστικό περιβάλλον ενώ αποτελείται από διάφορα μέρη. Το πρώτο μέρος είναι ο φάκελος, που χρησιμοποιείται για την περιγραφή του περιεχομένου ενός μηνύματος και μερικών ενδείξεων σχετικά με το πώς να το επεξεργαστεί. Το δεύτερο μέρος αποτελείται από τους κανόνες κωδικοποίησης των δομημένων δεδομένων. Το τελευταίο μέρος είναι μια σύμβαση για την σωστή αναπαράσταση κλήσεων διαδικασίας και απαντήσεων σε αυτές. Ακολουθεί μια λεπτομερής ανάλυση του SOAP.

4.1 ΕΙΣΑΓΩΓΗ ΣΤΙΣ ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ

Η εμφάνιση του SOAP το 1999 τράβηξε την προσοχή του οργανισμού W3C και ύστερα από μια διαδικασία αλλαγών, βελτιώσεων και απαραίτητων τροποποιήσεων, το 2003 το πρωτόκολλο SOAP με την έκδοση 1.2 έγινε η προτεινόμενη προδιαγραφή-πρωτόκολλο για τις Υπηρεσίες Web. Σύμφωνα με μια σύσταση του W3C, ένας ορισμός που αντιπροσωπεύει την προδιαγραφή του SOAP είναι ο εξής :

"Το SOAP είναι ένα ελαφρύ πρωτόκολλο προορισμένο για την ανταλλαγή δομημένων πληροφοριών σε ένα αποκεντρωμένο, διανεμημένο περιβάλλον. Χρησιμοποιεί τεχνολογίες XML για να καθορίσει ένα επεκτάσιμο πλαίσιο παρέχοντας μια δομή μηνυμάτων η οποία μπορεί να ανταλλαχτεί πάνω από ποικίλα δικτυακά πρωτόκολλα. Το πλαίσιο έχει σχεδιαστεί να είναι ανεξάρτητο από οποιοδήποτε προγραμματιστικό μοντέλο και σημασιολογία υλοποίησης.

Στον πυρήνα του το SOAP χρησιμοποιείται κατά την ανταλλαγή μηνυμάτων μεταξύ των web services και είναι ένα ελαφρύ πλαίσιο ανταλλαγής μηνυμάτων και δομημένων δεδομένων σύμφωνα με τη γλώσσα XML, με την οποία περιγράφεται το περιεχόμενο των μηνυμάτων που μεταφέρει. Ακριβώς λόγω του ότι το SOAP

Παπαργύρη Τριανταφυλλιά

αποτελεί ένα XML σχήμα, υποστηρίζεται από την πλειοψηφία των υφιστάμενων και νέων τεχνικών περιβαλλόντων και πλατφόρμων και μπορεί να μεταφερθεί από ποίκιλλα πρωτόκολλα μεταφοράς. Ο πιο διαδεδομένος τρόπος ανταλλαγής SOAP μηνυμάτων ωστόσο είναι μέσω του HTTP, το οποίο χρησιμοποιείται ευρέως στο Διαδίκτυο καθώς χρησιμοποιείται από τους browsers για την πλοήγηση στον παγκόσμιο ιστό. Τα δεδομένα μεταφέρονται ακριβώς όπως οι ιστοσελίδες και δεν εμποδίζονται από τείχη ασφαλείας (firewalls) που ενδέχεται να διαθέτουν οι υπολογιστές.

Το SOAP αποτελεί ένα σύνολο συμβάσεων που όχι μόνο καθορίζουν και περιγράφουν ένα μήνυμα αλλά ορίζουν και τις αλληλεπιδράσεις των κόμβων SOAP, οι οποίοι επεξεργάζονται τα μηνύματα κατά μήκος της διαδρομής τους. Ο κύριος στόχος του είναι η παροχή ενός ενιαίου τρόπου για το πακετάρισμα των δεδομένων των μηνυμάτων και ο καθορισμός κανόνων για τη κωδικοποίηση και αποκωδικοποίηση των δεδομένων κατά την μεταφορά. Τα βασικά χαρακτηριστικά του πρωτόκολλου SOAP είναι τα εξής :

1. **ΑΠΛΟΤΗΤΑ:** Εξαιτίας αυτού του χαρακτηριστικού του, το κόστος αλλά και η πολυπλοκότητα υλοποίησης του πρωτόκολλου SOAP μειώνονται αισθητά.
2. **ΕΥΕΛΙΞΙΑ:** Το συγκεκριμένο πρωτόκολλο κάνει χρήση προτύπων πρωτοκόλλων όπως το HTTP ως μέσα μεταφοράς. Συνεπώς είναι αρκετά ευέλικτο καθώς μπορεί να χρησιμοποιηθεί στο Διαδίκτυο χωρίς συμβιβασμούς στην ασφάλεια της υποδομής μιας επιχείρησης.
3. **ΑΝΕΞΑΡΤΗΣΙΑ:** Είναι ανεξάρτητο από πλατφόρμα και γλώσσα προγραμματισμού οπότε μπορεί να χρησιμοποιηθεί για την επικοινωνία μεταξύ εφαρμογών γραμμένων σε διαφορετικές γλώσσες προγραμματισμού και για διαφορετικές πλατφόρμες.
4. **ΕΠΕΚΤΑΣΙΜΟ:** Αποτελεί το πιο σημαντικό του χαρακτηριστικό καθώς έτσι μπορούν να βασιστούν επάνω του πολλές αναπτυσσόμενες τεχνολογίες υπηρεσιών παγκοσμίου ιστού προσφέροντας αξιοπιστία και ασφάλεια.

Τα παραπάνω χαρακτηριστικά είναι μόλις κάποια από τα βασικά πλεονεκτήματα του πρωτοκόλλου SOAP, το οποίο όπως έχει προαναφερθεί καθορίζει σαφώς το πλαίσιο ανταλλαγής μηνυμάτων μεταξύ των web services. Για να γίνουν πιο σαφείς οι λειτουργίες καθώς και η αρχιτεκτονική του SOAP μπορούμε να το διαχωρίσουμε στα εξής μοντέλα:

Η κατασκευή του μηνύματος SOAP, η οποία καθορίζει τη δομή του μηνύματος SOAP

Μοντέλο επεξεργασίας SOAP το οποίο καθορίζει τους κανόνες επεξεργασίας ενός μηνύματος SOAP για κάθε οντότητα που επεξεργάζεται SOAP μηνύματα

Υποκείμενη πρωτοκόλλων σύνδεση της μεταφοράς

4.2 ΜΗΝΥΜΑ SOAP

Το πρωτόκολλο SOAP αποτελεί ένα συνηθισμένο έγγραφο XML, το οποίο χρησιμοποιείται για την ανταλλαγή πληροφορίας μέσω δομής μηνυμάτων. Η δομή ενός μηνύματος SOAP, το οποίο αποτελεί και αυτό ένα έγγραφο XML, είναι πολύ οικεία καθώς θα μπορούσε να συμβολιστεί με την απλή δομή των γραμμάτων που ταχυδρομούνται στην πραγματική ζωή και έχει ως εξής:

- **ENVELOPE (ΦΑΚΕΛΟΣ):**

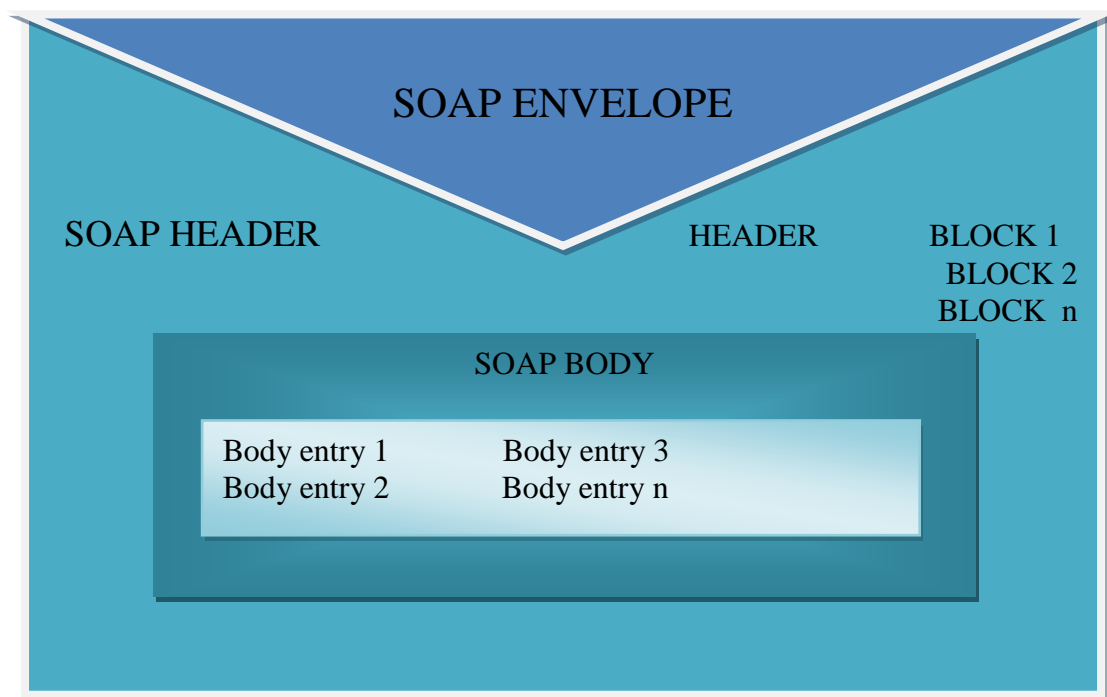
Αποτελεί ουσιαστικά το XML στοιχείο-ρίζα ενός μηνύματος SOAP μέσω του οποίου αναγνωρίζεται το XML έγγραφο ως μήνυμα SOAP. Σε αυτό το στοιχείο περιέχονται όλα τα άλλα στοιχεία XML του μηνύματος SOAP.

- **HEADER (ΕΠΙΚΕΦΑΛΙΔΑ):**

Το συγκεκριμένο στοιχείο είναι πολύ σημαντικό για λόγους που θα παρουσιάσουμε στην συνέχεια και περικλείει ένα επιπρόσθετο και προαιρετικό είδος πληροφορίας όπως η ασφάλεια και οι συναλλαγές.

- **BODY (ΣΩΜΑ):**

Αποτελεί υποχρεωτικό στοιχείο του μηνύματος και περιέχει τα πραγματικά ανταλλασσόμενα δεδομένα. Στην ουσία αυτό το στοιχείο αποτελεί τον σκοπό του μηνύματος



Σχ.2 Δομή ενός μηνύματος SOAP

Όλα τα παραπάνω στοιχεία που περιέχονται σε κάθε μήνυμα SOAP δηλώνονται στο προεπιλεγμένο namespace για το SOAP envelope:

<http://www.w3.org/2003/05/soap-envelope>

Το πρωτόκολλο SOAP περιλαμβάνει επίσης ένα σύνολο κανόνων για το πώς θα γίνεται η κωδικοποίηση κάθε διαφορετικού τύπου δεδομένων που αποστέλλεται σε μία εφαρμογή. Το προεπιλεγμένο namespace για την κωδικοποίηση του SOAP και τους τύπους δεδομένων είναι το :

<http://www.w3.org/2003/05/soap-envelope>

4.2.1 Envelope (Φάκελος)

Το συγκεκριμένο στοιχείο είναι υποχρεωτικό στοιχείο του μηνύματος καθώς είναι αυτό που προσδιορίζει ότι το συγκεκριμένο XML έγγραφο είναι μήνυμα SOAP. Ο προσδιορισμός του ως στοιχείο-ρίζα του μηνύματος οφείλεται στο ότι το συγκεκριμένο στοιχείο περικλείει όλα τα άλλα στοιχεία του μηνύματος.

Απαραίτητα χαρακτηριστικά αυτού του στοιχείου είναι τα εξής:

- Το όνομα του στοιχείου πρέπει να είναι Envelope
- Το όνομα του namespace του να είναι:
<http://www.w3.org/2003/05/soap-envelope>
- Να περιέχει μηδέν ή περισσότερες ιδιότητες ορισμένες με namespace.
- Να περιέχει ένα ή δυο παιδιά-στοιχεία με την εξής σειρά
 1. Προαιρετικό στοιχείο Header
 2. Υποχρεωτικό στοιχείο Body

4.2.2 Header (Επικεφαλίδα)

Το συγκεκριμένο στοιχείο αποτελεί τον σημαντικότερο λόγο που χρησιμοποιούμε την συγκεκριμένη δομή μηνύματος στην θέση ενός απλού XML εγγράφου. Αποτελεί προαιρετικό στοιχείο ,το οποίο έχει την ιδιότητα να αναπαριστά ένα επιπρόσθετο και επεκτάσιμο είδος πληροφορίας το οποίο μεταφέρεται μαζί με το υπόλοιπο μήνυμα δίχως να τροποποιείται ο πυρήνας του μηνύματος.

Στην ουσία το Soap Header παρέχει ένα μηχανισμό για την επέκταση ενός μηνύματος με ένα αποκεντρωμένο και δομημένο τρόπο. Κάθε στοιχείο Envelope μπορεί να περιέχει περισσότερα από ένα στοιχεία Headers. Οι επικεφαλίδες περιέχουν κομμάτια από δεδομένα με δυνατότητα επέκτασης, τα οποία ταξιδεύουν

μαζί με το μήνυμα και μπορούν να γίνουν στόχος για επέκταση σε συγκεκριμένους κόμβους του δικτύου. Έχουμε δυο μορφές επεκτασιμότητας την κατακόρυφη και την οριζόντια.

Κατακόρυφη Επεκτασιμότητα

Έχει ήδη αναφερθεί ότι το στοιχείο Headers αποτελεί προαιρετικό κομμάτι σε κάθε μήνυμα. Για λόγους ασφαλείας σε περιπτώσεις επεκτασιμότητας, η προδιαγραφή SOAP ορίζει την μεταβλητή must Understand. Η συγκεκριμένη μεταβλητή στην ουσία χρησιμοποιείται για να δείξει αν μια επικεφαλίδα που υπάρχει ήδη στο μήνυμα είναι απαραίτητη ώστε ο παραλήπτης να μπορέσει να επεξεργαστεί το μήνυμα. Όταν η τιμή της είναι αληθής (1) τότε ο παραλήπτης αυτού του κομματιού του μηνύματος πρέπει να την κατανοήσει πλήρως και να αποδεχτεί όλους τους όρους της για να συνεχιστεί η επεξεργασία. Αντίστοιχα όταν η επικεφαλίδα έχει τιμή ψευδή (0), σημαίνει ότι είναι προαιρετική και το μήνυμα είναι επεξεργάσιμο και από παραλήπτες που αγνοούν πλήρως τις προδιαγραφές που ορίζει η επικεφαλίδα.

Οριζόντια Επεκτασιμότητα

Ένα SOAP μήνυμα κατά μήκος της διαδρομής του από τον αποστολέα στον παραλήπτη μπορεί να περάσει από ένα σύνολο ενδιάμεσων επεξεργαστών οι οποίοι παραλαμβάνουν το μήνυμα και το προωθούν, ενδεχομένως τροποποιημένο. Ωστόσο το ότι το κυρίως μήνυμα προορίζεται για ένα συγκεκριμένο παραλήπτη δεν σημαίνει απαραίτητα ότι όλα τα μέρη του μηνύματος προορίζονται γι' αυτόν. Αντιθέτως κάποια μέρη μπορεί να απευθύνονται σε ενδιάμεσους παραλήπτες τους λεγόμενους actors. Σε αυτή την περίπτωση ο ενδιάμεσος παραλήπτης δεν προωθεί την συγκεκριμένη ιδιότητα της επικεφαλίδας αλλά μπορεί να προωθήσει μια παρόμοια.

Η τιμή που προσδιορίζει τον παραλήπτη μιας ιδιότητας της επικεφαλίδας είναι ένα URI. Για παράδειγμα το URI <http://schemas.xmlsoap.org/soap/actor/next> ορίζει ως παραλήπτη της επικεφαλίδας την πρώτη SOAP εφαρμογή που θα επεξεργαστεί το μήνυμα. Στην περίπτωση απουσίας της ιδιότητας actor ο τελικός παραλήπτης είναι ο μοναδικός παραλήπτης όλου του μηνύματος.

4.2.3 Header (Σώμα)

Το συγκεκριμένο στοιχείο είναι υποχρεωτικό στοιχείο του μηνύματος SOAP και εμπεριέχει όλη την πληροφορία που επιθυμεί να στείλει ο αποστολέας στον παραλήπτη. Απαραίτητα χαρακτηριστικά αυτού του στοιχείου είναι τα εξής:

- Το όνομα του στοιχείου πρέπει να είναι Body.
- Το όνομα του namespace του να είναι:
<http://www.w3.org/2003/05/soap-envelope>

- Να περιέχει μηδέν ή περισσότερες ιδιότητες ορισμένες με namespace.
- Μπορεί να περιέχει μηδέν ή περισσότερα στοιχεία- παιδιά ορισμένα με namespace (Soap Body Child Element).
- Μπορεί να περιέχει μηδέν ή περισσότερους κόμβους-παιδιά χαρακτήρων.

Κατά την μετάδοση του μηνύματος αν κάποιο σημείο δεν είναι κατανοητό τότε πρέπει να συνταχθεί ένα μήνυμα λάθους το λεγόμενο SOAP FAULT. Τα μηνύματα λάθους είναι απλά μηνύματα SOAP τα οποία περιέχουν ένα απλό στοιχείο Fault μέσα στο στοιχείο Body. Για να αναγνωριστεί επομένως ότι ένα μήνυμα SOAP κουβαλάει δεδομένα σφάλματος πρέπει να περιέχει ένα στοιχείο Fault σαν το μοναδικό στοιχείο-παιδί του στοιχείου Body.

Το μήνυμα λάθους έχει συγκεκριμένη δομή παρέχοντας έτσι πληροφορία σχετικά με το λάθος που συνέβη. Τέτοια λάθη είναι τα εξής:

- Λανθασμένη ή ελλιπής πληροφορία από τον αποστολέα
- Λάθος του παραλήπτη κατά την επεξεργασία
- Λάθος κάποιου μεσολαβητή με την τιμή της μεταβλητής must Understand
- Μη συμβατές εκδόσεις πρωτοκόλλου από κόμβο σε κόμβο

Όλα τα παραπάνω περιέχονται σε μορφή κωδικοποιημένης πληροφορίας στο στοιχείο Code.

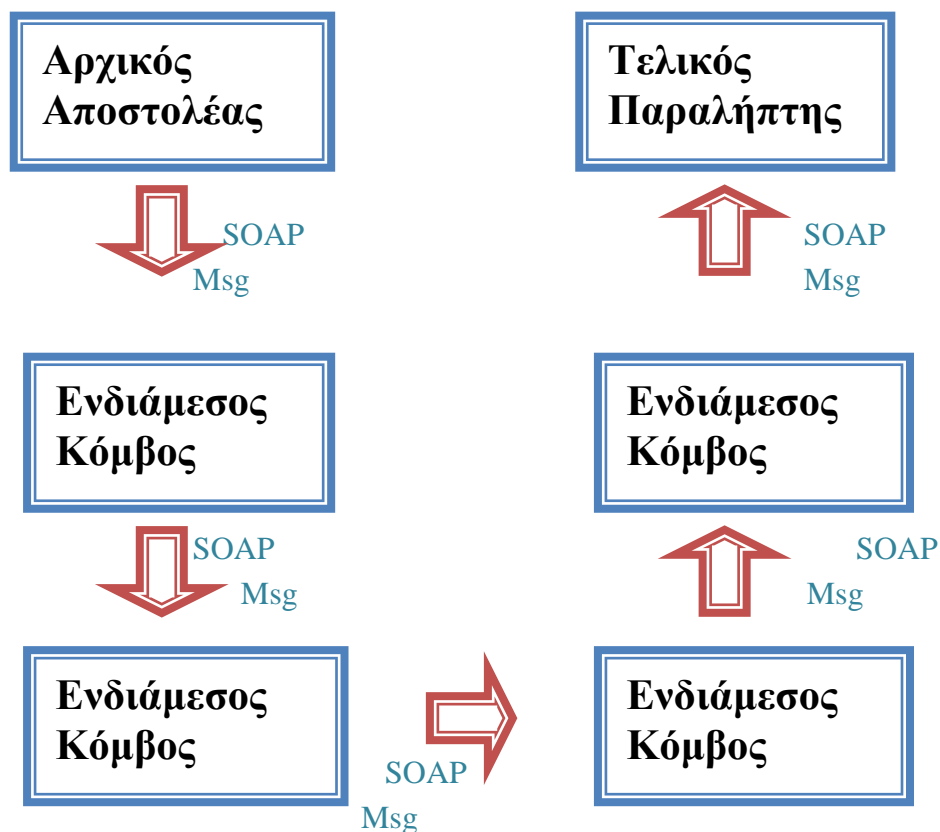
4.3 ΜΟΝΤΕΛΟ ΕΠΕΞΕΡΓΑΣΙΑΣ SOAP

Το SOAP είναι βασικά ένα μοντέλο μονόδρομης επικοινωνίας, το οποίο εγγυάται ότι ένα μήνυμα μεταφέρεται από τον αποστολέα στον παραλήπτη, ενδεχομένως περιλαμβάνοντας ενδιάμεσους σταθμούς (κόμβους) που μπορούν να επεξεργαστούν μέρος του μηνύματος ή και να το μεταβάλουν.

Το μοντέλο επεξεργασίας SOAP περιγράφει αυτό το καταναεμημένο μοντέλο επεξεργασίας, τους συμμετέχοντες, τους κόμβους SOAP και πως ένας δέκτης SOAP επεξεργάζεται ένα μήνυμα SOAP. Πιο συγκεκριμένα ένας κόμβος αναπαριστά τη λογική που είναι υπεύθυνη για τη μεταβίβαση, λήψη, επεξεργασία ή και αναμετάδοση ενός μηνύματος SOAP.

Σε γενικές γραμμές οι κόμβοι SOAP έχουν τη δυνατότητα να στέλνουν και να λαμβάνουν μηνύματα SOAP. Κάθε κόμβος ανάλογα με την λειτουργία που εκτελεί σε σχέση με το μήνυμα SOAP έχει και την ανάλογη ιδιότητα. Αναλυτικότερα ένας κόμβος λέγεται αποστολέας SOAP όταν μεταδίδει το μήνυμα, παραλήπτης SOAP όταν λαμβάνει το μήνυμα, ενώ ενδιάμεσος SOAP όταν ταυτόχρονα λαμβάνει και στην συνέχεια μεταδίδει το μήνυμα SOAP. Ο αρχικός κατασκευαστής του μηνύματος ονομάζεται Αρχικός Αποστολέας ενώ ο τελικός αποδέκτης λέγεται

Τελικός Παραλήπτης. Στο σχήμα που ακολουθεί παρουσιάζεται μια αναπαράσταση της διαδρομής που ακολουθεί κάθε μήνυμα SOAP ώστε να καταλήξει από τον αποστολέα στο παραλήπτη.



Σχ.3: Διαδρομή Μηνύματος SOAP

4.4 ΜΟΝΤΕΛΟ ΑΝΤΑΛΛΑΓΗΣ ΜΗΝΥΜΑΤΩΝ

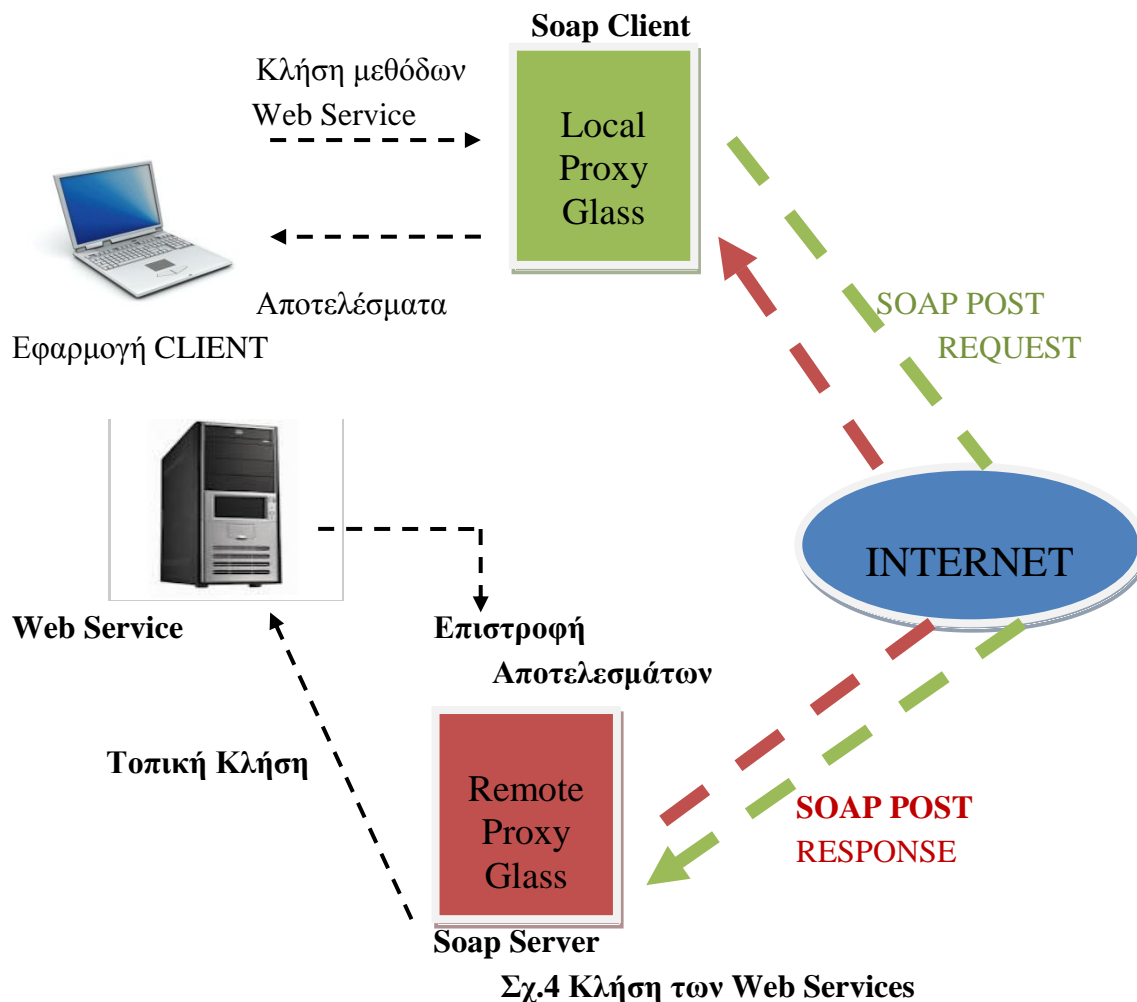
Το SOAP είναι ένα απλό πλαίσιο μηνυμάτων για ανταλλαγή μηνυμάτων σε μορφή XML μεταξύ ενός αρχικού αποστολέα και ενός τελικού αποδέκτη. Τα πιο ενδιαφέροντα σενάρια περιλαμβάνουν πολλαπλές ανταλλαγές μηνυμάτων μεταξύ των δυο αυτών κόμβων. Η πιο απλή μορφή είναι αυτής της αίτησης-απάντησης.

4.4.1 Απομακρυσμένες Κλήσεις Διαδικασιών.

Κάθε μήνυμα SOAP ταυτόχρονα με όλες τις παραπάνω λειτουργίες του, υλοποιεί κλήσεις απομακρυσμένης διαδικασίας RPC(Remote Procedure Call), κάτι το οποίο αποτελούσε και ένα από τους στόχους κατά τον σχεδιασμό του. Πιο συγκεκριμένα για την κλήση μιας μεθόδου ενός απομακρυσμένου αντικειμένου μιας Web Service έχει καθοριστεί μια συγκεκριμένη αναπαράσταση μηνύματος SOAP, η οποία περιλαμβάνει στο σώμα του το όνομα της μεθόδου, τις επιτρεπόμενες

παραμέτρους και το URL της. Το μοντέλο που ακολουθείται είναι κυρίως της μορφής αίτησης – απάντησης ενώ η όλη διαδικασία πραγματοποιείται μεταξύ του πελάτη και του διακομιστή.

Ακολουθεί ένα σχήμα το οποίο θα μας βοηθήσει να κατανοήσουμε αυτή την αμφίδρομη επικοινωνία.



Στο συγκεκριμένο παράδειγμα ο client εκτελεί κλήση με το μοντέλο request – response. Παρατηρούμε ότι εδώ ο client εκτελεί αίτηση για συγκεκριμένη web service στον διακομιστή μεσολάβησης και αυτός με τη σειρά του αναλαμβάνει να προσκομίσει σε αυτόν τις σελίδες που ζητάει.

4.4.2 HTTP Binding

Λόγω της ευρείας χρήσης του στο Internet αλλά και της απλότητας του, το HTTP έχει χρησιμοποιηθεί σε μεγάλο βαθμό για την μεταφορά μηνυμάτων SOAP. Ο συσχετισμός του SOAP με το HTTP πρωτόκολλο έχει τυποποιηθεί από τον W3C και 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 εφόσον ο Αρχικός Αποστολέας αναγνωρίσει τον εξυπηρετητή μέσω ενός URI, συνδέεται σε αυτόν

χρησιμοποιώντας το TCP/IP. Στην συνέχεια στέλνει το αρχικό μήνυμα SOAP από τον κόμβο του σε μία HTTP POST ΑΙΤΗΣΗ (request) ενώ το απαντητικό μήνυμα αποστέλλεται από τον Τελικό Παραλήπτη στην HTTP POST ΑΠΑΝΤΗΣΗ (response). Το HTTP συνδέει το μήνυμα αίτησης με το μήνυμα απάντησης.

4.5 ΠΑΡΑΔΕΙΓΜΑ ΕΝΟΣ SOAP ΜΗΝΥΜΑΤΟΣ

Στο παρόν υποκεφάλαιο παρουσιάζεται ένα πλήρες παράδειγμα ενός SOAP μηνύματος όπου εφαρμόζουμε όλα όσα μάθαμε για την προδιαγραφή SOAP.

```
<? 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: 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>
```

ΕΠΕΞΗΓΗΣΗ ΠΑΡΑΔΕΙΓΜΑΤΟΣ

Το παρόν μήνυμα αποστέλλεται σε μια Web Service την stockinfo.example.com, η οποία για ένα συγκεκριμένο κωδικό χρηματιστηρίου (b:stockexchange) και κωδικό μετοχής (b:stock) επιστρέφει την τρέχουσα αξία.

Στην πρώτη γραμμή του μηνύματος σημειώνεται η δήλωση XML που μας δίνει την έκδοση της XML που χρησιμοποιείται (μην ξεχνάμε ότι κάθε μήνυμα SOAP αποτελεί ένα XML έγγραφο).

Στην συνέχεια ακολουθεί το στοιχείο-Επικεφαλίδα (env:Header). Η πληροφορία στο συγκεκριμένο στοιχείο είναι ένα συγκεκριμένο νούμερο αναφοράς (reference number) το οποίο πιστοποιεί τον πελάτη ή χρήστη που καλεί την υπηρεσία ιστού. Παρατηρούμε ότι η μεταβλητή mustUnderstand δηλώνεται " true ", υποδεικνύοντας ότι η κατανόηση και επεξεργασία αυτού του SOAP Header είναι υποχρεωτική, ώστε να μπορέσει στη συνέχεια ο παραλήπτης να επεξεργαστεί το μήνυμα. Η δεύτερη γραμμή καθώς και τα <env:Header>, <env:Body> αποτελούν τις δηλώσεις όσον αφορά τον φάκελο, τις επικεφαλίδες και το σώμα του μηνύματος SOAP.

Στην περίπτωση που το μήνυμα ανταλλάσσεται με τη μέθοδο HTTP POST, HTTP POST στο request του μηνύματος στην αρχική γραμμή θα υπήρχε μια δήλωση

της μορφής π.χ POST /Reservations HTTP/1.1 ενώ στο αντίστοιχο HTTP POST response μήνυμα θα υπήρχε αντίστοιχα η δήλωση HTTP/1.1 200 OK.

ΚΕΦΑΛΑΙΟ 5

5 WSDL - WEB SERVICES DESCRIPTION LANGUAGE

Στο προηγούμενο κεφάλαιο υπήρξε μια πλήρης αναφορά στο πρωτόκολλο SOAP. Είδαμε ότι το συγκεκριμένο πρωτόκολλο είναι βασισμένο στην XML και ότι μέσω αυτού του πρωτοκόλλου μπορούν να γίνουν αιτήσεις για απομακρυσμένες κλήσεις διαδικασιών από κάποιον client σε κάποιον server. Από την μεριά του πελάτη όμως υπήρχαν δυσκολίες στο καθορισμό της σύνταξης του μηνύματος SOAP, καθώς ο πελάτης δε μπορούσε να ξέρει τι ακριβώς μήνυμα να στείλει. Αν και το πρωτόκολλο SOAP καθορίζει κάποιους κανόνες και ένα συγκεκριμένο μορφότυπο για τα μηνύματα, ωστόσο δεν διέθετε κανέναν μηχανισμό στον πελάτη ώστε να είναι σε θέση να γνωρίζει αρκετές λεπτομέρειες προτού στείλει το μήνυμα, όπως που ακριβώς να το στείλει, ποιες μεθόδους πρέπει να καλέσει από την υπηρεσία ή ποια πρωτόκολλα επικοινωνίας υποστηρίζονται από τον παροχέα της υπηρεσίας.

Είναι εμφανές λοιπόν ότι πελάτης θα πρέπει να είναι σε θέση να γνωρίζει όλες τις λεπτομέρειες για την κλήση μιας υπηρεσίας διαδικτύου, συνεπώς να μπορεί να έχει στην κατοχή του μια πλήρη περιγραφή αυτής. Γι' αυτό το λόγο δημιουργήθηκε ένας καθορισμένος μηχανισμός, η γλώσσα περιγραφής Υπηρεσιών Παγκοσμίου Ιστού WSDL (Web Services Description Language), με σκοπό την παροχή πληροφοριών σχετικά με όλα αυτά που χρειάζεται να γνωρίζει ο πελάτης ώστε να καλέσει μια υπηρεσία. Ακολουθεί μια πλήρης αναφορά

5.1 ΕΙΣΑΓΩΓΗ ΣΤΙΣ ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ

Σύμφωνα με μια σύσταση του W3C, ένας ορισμός που αντιπροσωπεύει την προδιαγραφή του SOAP είναι ο εξής :

"Η WSDL είναι ένα σχήμα XML για την περιγραφή δικτυακών υπηρεσιών σαν ένα σύνολο από τελικά σημεία που λειτουργούν σε μηνύματα τα οποία περιέχουν πληροφορία είτε προσανατολισμένη στα έγγραφα είτε προσανατολισμένη στις διαδικασίες.

Οι λειτουργίες και τα μηνύματα περιγράφονται περιληπτικά, και τότε δένονται σε ένα συγκεκριμένο πρωτόκολλο δικτύων και μορφή μηνυμάτων για να καθορίζουν ένα τελικό σημείο. Πολλά σχετικά τελικά σημεία συνδυάζονται σε υπηρεσίες.

Η WSDL είναι επεκτάσιμη στο να επιτρέπει την περιγραφή τελικών σημείων και των μηνυμάτων τους άσχετα από τη μορφή των μηνυμάτων και των πρωτοκόλλων δικτύων που χρησιμοποιούνται για την επικοινωνία. Παρόλα αυτά, αυτή τη στιγμή στην προδιαγραφή της WSDL οι μόνες συνδέσεις που περιλαμβάνονται περιγράφουν πώς μπορούμε να χρησιμοποιήσουμε την WSDL σε συνδυασμό με το SOAP 1.1, το HTTP GET/POST και το MIME"

Με απλά λόγια η WSDL μας βοηθάει να περιγράψουμε ένα σύνολο από μηνύματα και τον τρόπο με τον οποίο αυτά τα μηνύματα ανταλλάσσονται. Η WSDL βασίζεται στην XML για την περιγραφή των δημόσιων διεπαφών (public interfaces) μιας υπηρεσίας παγκοσμίου ιστού. Η δημόσια διεπαφή μπορεί να περιλαμβάνει πληροφορία σχετική με:

- Όλες τις διαθέσιμες δημόσιες δραστηριότητες της υπηρεσίας παγκοσμίου ιστού.
- Τα πρωτόκολλα XML μηνυμάτων που υποστηρίζουν Web Service.
- Το είδος δεδομένων
- Τη δέσμευση για το συγκεκριμένο πρωτόκολλο μεταφοράς που πρόκειται να χρησιμοποιηθεί.
- Τη διεύθυνση για τον εντοπισμό της συγκεκριμένης υπηρεσίας παγκοσμίου ιστού.

Η WSDL είναι ανεξάρτητη ως προς την πλατφόρμα και την γλώσσα προγραμματισμού. Αυτό το γεγονός την κάνει κατάλληλη για να περιγράψει διεπαφές web services, οι οποίες είναι προσβάσιμες από μια μεγάλη ποικιλία πλατφόρμων και γλωσσών προγραμματισμού. Επιπρόσθετα εκτός του ότι περιγράφει τα περιεχόμενα των μηνυμάτων, ορίζει που είναι διαθέσιμη μια υπηρεσία και ποια πρωτόκολλα επικοινωνίας χρησιμοποιούνται για να επικοινωνήσουμε με αυτή την υπηρεσία. Επομένως κάθε αρχείο WSDL ορίζει όλα όσα χρειάζονται για να γράψουμε ένα πρόγραμμα το οποίο να λειτουργεί με ένα Web Service.

Συνοπτικά η WSDL περιγράφει τις ακόλουθες τρεις σημαντικές ιδιότητες κάθε υπηρεσίας web.

- Τι κάνει η Υπηρεσία
 - Μέθοδοι της
 - Παράμετροι της
 - Αποτελέσματα
- Πώς γίνεται η πρόσβαση στην Υπηρεσία
 - Πρωτόκολλο
 - Είδος κωδικοποίησης
- Που βρίσκεται η Υπηρεσία
 - Δικτυακή Διεύθυνση

5.2 ΤΕΧΝΙΚΗ ΠΕΡΙΓΡΑΦΗ ΤΗΣ WSDL

Η γραμματική και η σύνταξη που περιγράφουν την υπηρεσία παγκοσμίου ιστού, καθορίζονται από την WSDL σαν μια συλλογή από επικοινωνιακά άκρα. Ένα

έγγραφο WSDL καθορίζει τις υπηρεσίες Διαδικτύου (web services) ως ένα σύνολο από τελικά σημεία ή θύρες Υπηρεσίας(ports). Η περιγραφή της κάθε θύρας Υπηρεσίας αποτελείται από τρία 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.
- **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.

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:

< TYPES >

Το συγκεκριμένο στοιχείο ορίζει τους τύπους δεδομένων που ανταλλάσσονται μέσω των μηνυμάτων όταν χρησιμοποιούνται οι διάφορες λειτουργίες που προσφέρει μία υπηρεσία ιστού μέσω των θυρών υπηρεσιών που παρουσιάζει.

< MESSAGE >

Το συγκεκριμένο στοιχείο ορίζει τη μορφή των μηνυμάτων που ανταλλάσσονται κατά την χρήση των λειτουργιών των θυρών υπηρεσίας.

< PORTTYPE >

Το συγκεκριμένο στοιχείο ορίζει όλες τις λειτουργίες που υποστηρίζονται από μια ή περισσότερες θύρες Υπηρεσίας. Ουσιαστικά μέσω αυτού του στοιχείου η κάθε

Υπηρεσία Web ορίζεται σαν το σύνολο των λειτουργιών που υποστηρίζονται και των μηνυμάτων που ανταλλάσσονται.

< OPERATION >

Το συγκεκριμένο στοιχείο ορίζει όλες τις λειτουργίες που υποστηρίζει μια συγκεκριμένη Υπηρεσία.

< BINDING >

Κάθε τέτοιο στοιχείο ορίζει το πως κάθε ανταλλασσόμενο μήνυμα στα πλαίσια κάθε λειτουργίας μπορεί να συσχετιστεί με ένα πρωτόκολλο ανταλλαγής μηνυμάτων για ένα συγκεκριμένο τύπο θυρών Υπηρεσίας.

< SERVICE >

Το συγκεκριμένο στοιχείο αποτελεί μια συλλογή από σχετικά τελικά σημεία ή θύρες Υπηρεσίας (ports). Στην ουσία το συγκεκριμένο στοιχείο χρησιμοποιείται για να περικλείσει μέσα του το σύνολο των πληροφοριών που χρησιμοποιούνται για να περιγράψουν κάθε υπηρεσία παγκοσμίου Ιστού.

ΚΕΦΑΛΑΙΟ 6

6 UDDI - UNIVERSAL DESCRIPTION, DISCOVERY AND INTEGRATION

Στο προηγούμενο κεφάλαιο αναφερθήκαμε στην WSDL, την γλώσσα μέσω της οποίας ο κάθε πάροχος μιας Υπηρεσίας Διαδικτύου περιγράφει την υπηρεσία του έτσι ώστε να γνωρίζει ο κάθε πιθανός πελάτης τι κάνει αυτή η Υπηρεσία, πώς να εγκαθιδρύσει επικοινωνιακή σύνδεση με τον πάροχο της καθώς και τον τρόπο με τον οποίο πρέπει να την καλέσει. Το ερώτημα είναι που δημοσιεύει ο κάθε πάροχος την περιγραφή της Υπηρεσίας του έτσι ώστε να είναι προσβάσιμη σε όλους τους πιθανούς πελάτες.

Όπως έχει ήδη αναφερθεί σε προηγούμενο κεφάλαιο στην SOA αρχιτεκτονική υπάρχει ο ρόλος της Υπηρεσίας Καταλόγου. Ο πελάτης προκειμένου να ανακαλύψει την Υπηρεσία που τον ενδιαφέρει, ψάχνει πρώτα στον κατάλογο αυτό και είναι εδώ το σημείο όπου ο κάθε πάροχος μιας Υπηρεσίας δημοσιεύει την περιγραφή της Υπηρεσίας του. Η επικρατούσα τεχνολογία καταλόγου στον κόσμο των web services είναι το UDDI (Universal Description, Discovery and Integration), για το οποίο ακολουθεί μια σχετικά λεπτομερής περιγραφή.

6.1 ΕΙΣΑΓΩΓΗ ΣΤΙΣ ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ

Η αρχιτεκτονική SOA υποστηρίζει τόσο την δυναμική σύνδεση με Υπηρεσίες Web όσο και την δυνατότητα επαναχρησιμοποίησης τους μέσω της σύνθεσης υπάρχοντων υπηρεσιών σε νέες. Συνεπώς η δημιουργία του Προτύπου Καταλόγου UDDI το 2000 αποσκοπούσε στην διευκόλυνση ανακάλυψης των Υπηρεσιών Διαδικτύου τόσο κατά τον χρόνο σχεδίασης όσο και δυναμικά κατά τον χρόνο εκτέλεσης τους.

Το UDDI παρέχει ένα παγκόσμιο αρχείο καταγραφής Υπηρεσιών Διαδικτύου για να διευκολύνει τις επιχειρήσεις στη γρήγορη, εύκολη και δυναμική εύρεση και συναλλαγή με άλλες επιχειρήσεις. Ουσιαστικά για πρώτη φορά λειτούργησε στις 2 Μαΐου 2001 και αποτελείται στην πραγματικότητα από 2 πανομοιότυπους καταλόγους, που διατηρούνται αυτή τη στιγμή από δύο εταιρείες (IBM και Microsoft), οι οποίες αποτελούν και τους διαχειριστές του UDDI.

Σύμφωνα με μια σύσταση του OASIS, ο σκοπός για το UDDI(Universal Description, Discovery and Integration) είναι ο εξής:

"Τα web services έχουν νόημα μόνο όταν δυνητικοί χρήστες μπορούν να βρουν πληροφορίες ικανές ώστε να επιτρέψουν την εκτέλεση τους. Το UDDI (Universal Description, Discovery and Integration) εστιάζει στον καθορισμό ενός συνόλου από υπηρεσίες που θα υποστηρίζουν την περιγραφή και την ανακάλυψη :

- Των εταιρειών, των οργανισμών και άλλων παρόχων web services.
- Των web services που είναι διαθέσιμες
- Των τεχνικών διεπαφών οι οποίες μπορούν να χρησιμοποιηθούν ώστε να έχει κάποιος χρήστης πρόσβαση σε αυτές τις υπηρεσίες.

Βασισμένο σε ένα κοινό σύνολο από βιομηχανικά πρότυπα συμπεριλαμβανομένων των HTTP, XML, XML Schema και SOAP παρέχει μια διαλειτουργική, θεμελιώδη υποδομή για ένα περιβάλλον λογισμικού προσανατολισμένο στις υπηρεσίες τόσο για δημόσια διαθέσιμες υπηρεσίες όσο και για υπηρεσίες που εκτίθενται μόνο εσωτερικά ενός οργανισμού. "

Με πολύ απλά λόγια το UDDI αποτελεί τον Χρυσό Οδηγό των web services. Ορίζει τον τρόπο καταχώρησης των Web Services στο μητρώο και είναι και αυτό βασισμένο στην γλώσσα XML. Στόχος του είναι η παροχή αναγκαίας υποδομής για την περιγραφή και αναζήτηση Web Services. 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.

Συγκεκριμένα το UDDI απευθύνεται σε δυο ειδών πελάτες. Τις επιχειρήσεις(Πάροχοι Υπηρεσίας) που μπορούν να δημοσιεύσουν μια πλήρη περιγραφή της υπηρεσίας που παρέχουν και τους πελάτες (Αιτουμένους Υπηρεσίας) που επιθυμούν να χρησιμοποιήσουν συγκεκριμένες Υπηρεσίες συνδεδεμένοι προγραμματιστικά με αυτές. Εξαιτίας του Καταλόγου UDDI όλες οι Υπηρεσίες έχουν το δικαίωμα να καταχωρούνται πλέον σε ένα Μητρώο Υπηρεσιών (UDDI Service Register) και από εκεί είναι προσπελάσιμες τόσο από τον Πάροχο Υπηρεσίας (Service Provider) όσο και από τον Πελάτη (Service Requestor). Είναι σαφές λοιπόν ότι υπάρχει μια αναγκαία αλληλεπίδραση μεταξύ του Παρόχου Υπηρεσίας, του Αιτούμενου και του Μητρώου Υπηρεσιών. Αυτή η αλληλεπίδραση πραγματοποιείται μέσω των εξής τριών βασικών λειτουργιών:

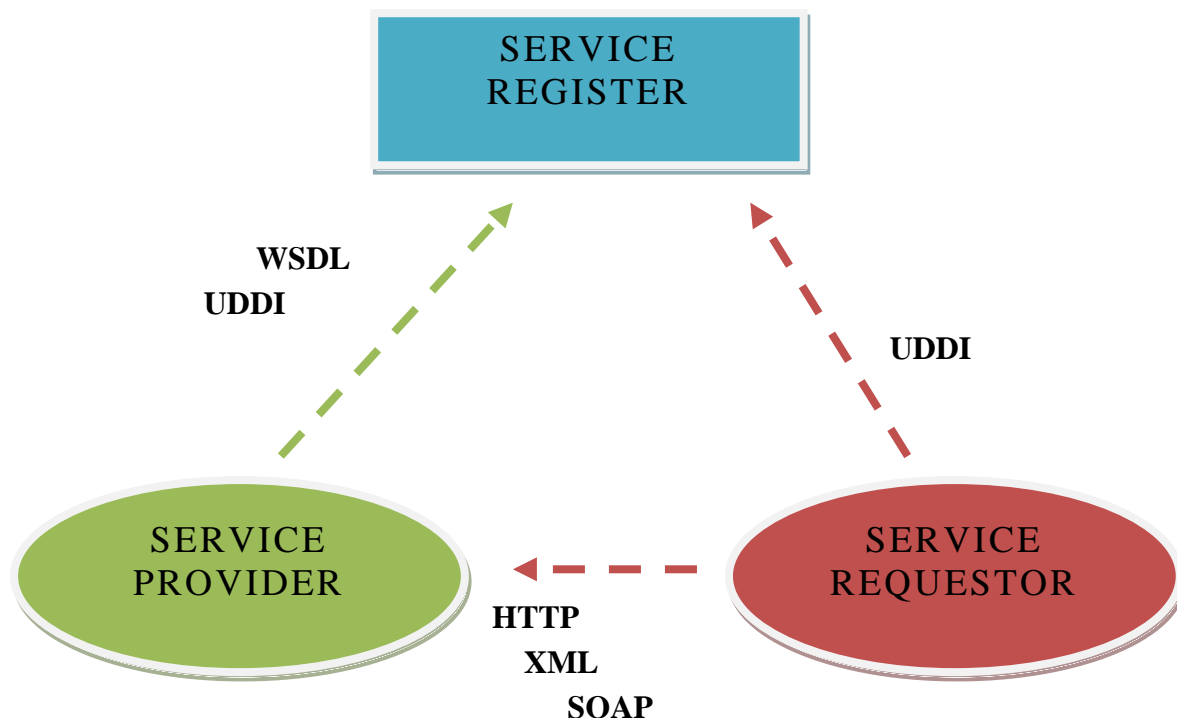
- **PUBLISH**: Δημοσίευση περιγραφής Υπηρεσιών
- **FIND**: Εύρεση περιγραφών Υπηρεσιών
- **BIND**: Επίκληση Υπηρεσιών

Ακολουθεί μια εικονική αναπαράσταση του τρόπου αλληλεπίδρασης των τριών υπηρεσιών.

Ο πάροχος Υπηρεσίας μέσω του WSDL και του UDDI δημοσιοποιεί (publish) την περιγραφή της υπηρεσίας του στον UDDI κατάλογο Υπηρεσίας.

Ο Αιτούμενος αναζητάει (find) μια Υπηρεσία μέσω του καταλόγου UDDI.

Ο Αιτούμενος Υπηρεσίας εφόσον ανακαλύψει μέσω του καταλόγου την Υπηρεσία που τον ενδιαφέρει συνδέεται-αλληλεπιδρά (bind) με τον αντίστοιχο Πάροχο μέσω των πρωτοκόλλων HTTP, SOAP και XML.



Σχ.5 Αλληλεπίδραση Υπηρεσιών

6.2 ΟΡΙΣΜΟΣ ΔΟΜΩΝ ΔΕΔΟΜΕΝΩΝ

Όλες οι εταιρείες και οι οργανισμοί που επιθυμούν να δημιουργήσουν μια Υπηρεσία Διαδικτύου, έχουν το δικαίωμα εφόσον το επιθυμούν, να δημοσιοποιούν δημόσιες πληροφορίες γι' αυτούς και πληροφορίες για τις Υπηρεσίες Διαδικτύου που παρέχουν μέσω του προτύπου Καταλόγου UDDI. Η πληροφορία στο UDDI ταξινομείται σύμφωνα με τον τύπο των εταιρειών (πάροχοι Υπηρεσιών Διαδικτύου) και την γεωγραφική τους θέση. Όταν κάποια επιχείρηση ψάξει να βρει κάποια άλλη επιχείρηση βάση των παραπάνω κριτηρίων, το UDDI μέσω ενός δείκτη στο WSDL αρχείο περιγράφει τις Υπηρεσίες Διαδικτύου που παρέχει η συγκεκριμένη επιχείρηση-Πάροχος.

Μπορούμε να ταξινομήσουμε τις παρεχόμενες Υπηρεσίες του UDDI, με βάση την δομή ενός τηλεφωνικού καταλόγου οπότε θα είχαμε

- **Λευκές Σελίδες** που περιέχουν γενικές πληροφορίες επικοινωνίας με τον πάροχο.
- **Κίτρινες Σελίδες** που περιέχουν πληροφορίες ταξινόμησης επιχειρήσεων και οργανισμών ανάλογα με την λειτουργία τους για τους τύπους και την τοποθεσία των υπηρεσιών.

- *Πράσινες Σελίδες* που περιέχουν τεχνικές πληροφορίες πρόσβασης στις προσφερόμενες υπηρεσίες του συγκεκριμένου παρόχου.

Κάθε καταχώρηση ενός παρόχου σε ένα Μητρώο Υπηρεσιών (UDDI Service Register), ουσιαστικά είναι ένα XML έγγραφο που αποτελείται κυρίως από τους εξής τέσσερις τύπους δεδομένων:

businessEntity

Ο συγκεκριμένος τύπος περιγράφει οποιαδήποτε πάροχο προσφέρει Υπηρεσίες Ιστού και έχει επιλέξει να τις καταχωρήσει στον Μητρώο Υπηρεσιών (UDDI Service Register) και παρέχει λεπτομέρειες για το πως μπορεί να επικοινωνήσει κάποιος μαζί του, κλπ.

businessService

Ο συγκεκριμένος τύπος δεδομένων περιγράφει το σύνολο των Υπηρεσιών Διαδικτύου που παρέχονται από έναν συγκεκριμένο οργανισμό ή επιχείρηση(Πάροχος Υπηρεσίας), ο οποίος περιγράφεται με το τύπο businessEntity.

bindingTemplate

Ο συγκεκριμένος τύπος δεδομένων περιγράφει όλες τις τεχνικές πληροφορίες υλοποίησης μιας Υπηρεσίας Διαδικτύου.

tModel

Ο συγκεκριμένος τύπος εμπεριέχει επαναχρησιμοποιήσιμες έννοιες που προσφέρουν το τεχνικό μοντέλο (technical model) περιγραφής προδιαγραφών των Υπηρεσιών όπως ένας τύπος υπηρεσίας ιστού, ένα πρωτόκολλο για επικοινωνία με υπηρεσίες ιστού. Έτσι, στοιχεία tModel μπορούν να αναπαριστούν πρωτόκολλα μεταφοράς δεδομένων όπως HTTP ή SMTP, φόρμα ταχυδρομικών διευθύνσεων κ.α.

ΚΕΦΑΛΑΙΟ 7

7 ΥΛΟΠΟΙΗΣΗ WEB SERVICE

7.1 ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΑ ΕΡΓΑΛΕΙΑ ΣΥΣΤΗΜΑΤΟΣ

Μέχρι αυτό το σημείο έχει υπάρξει μια πλήρης αναφορά στις βασικές τεχνολογίες των Web Services, χωρίς την παρουσία των οποίων δεν θα μπορούσαμε να μιλάμε σήμερα για Υπηρεσίες Διαδικτύου. Στο παρόν κεφάλαιο θα ασχοληθούμε με την υλοποίηση των Web Services. Για να μπορέσουμε να αναπτύξουμε ένα αξιόπιστο και εφαρμόσιμο web service, υπάρχει μια πληθώρα εργαλείων πάνω στα οποία μπορούμε να στηρίξουμε την σχεδίαση μας.

Πολλές ήδη υπάρχουσες γλώσσες προγραμματισμού, εξαιτίας της ραγδαίας εξάπλωσης και απήχησης των Υπηρεσιών Ιστού στους χρήστες Διαδικτύου, ενσωμάτωσαν στις δυνατότητες τους τη δυνατότητα για αυτόματη δημιουργία SOAP Servers και δημοσίευση και ανάκληση web services. Από την μεριά της Microsoft, οι έτοιμες λύσεις που δίνει το περιβάλλον Visual Studio.NET έχουν προσελκύσει πολλούς προγραμματιστές για να δημιουργούν τέτοιες Υπηρεσίες ενώ ταυτόχρονα άλλες μεγάλες εταιρείες όπως η IBM και η ORACLE διαθέτουν τα δικά τους προγραμματιστικά εργαλεία.

Στην συνέχεια ακολουθεί μια αναφορά στα σημαντικότερα και πιο χρησιμοποιούμενα εργαλεία που συναντάμε κατά την σχεδίαση των Υπηρεσιών Διαδικτύου

7.1.1 Περιβάλλον Πλατφόρμας

VISUAL STUDIO

Το Visual Studio αποτελεί ένα ολοκληρωμένο περιβάλλον ανάπτυξης που χρησιμοποιείται για την ανάπτυξη εφαρμογών Διαδικτύου και desktop εφαρμογών καθώς και για Υπηρεσίες Διαδικτύου. Αποτελεί ένα ενοποιημένο περιβάλλον, στο οποίο οι προγραμματιστές μπορούν να χρησιμοποιούν υπάρχουσες δεξιότητες για να δημιουργούν μοντέλα ενός ολοένα μεγαλύτερου αριθμού εφαρμογών, να συγγράφουν κώδικα γι' αυτές, να εντοπίζουν τα σφάλματά τους, να τις δοκιμάζουν και να τις αναπτύσσουν.

Παρέχει ισχυρά εργαλεία τεκμηρίωσης και εγγραφής κώδικα, ανίχνευσης και διόρθωσης σφαλμάτων, σχεδίασης κλάσεων και σχήματος βάσης δεδομένων. Οι

ενσωματωμένες γλώσσες προγραμματισμού που συναντάμε σε αυτό το περιβάλλον είναι οι Visual Basic, Visual C++ και η Visual C#.

TWISTER

Το Twister είναι μια πλατφόρμα που μπορούμε να χρησιμοποιήσουμε για τον σχεδιασμό και την λειτουργία των web services. Υποστηρίζει πλήρως την ανάπτυξη και εκτέλεση των ροών εργασίας και είναι γραμμένος σε γλώσσα JAVA. Πιο αναλυτικά η συγκεκριμένη πλατφόρμα τρέχει μέσα σε έναν servlet υποδοχέα τον Tomcat, ο οποίος αναλύεται παρακάτω και ο οποίος χάρη στον Axis μπορεί να καλεί τις web services του Twister και επιπλέον μπορεί να εμφανίζει τις web σελίδες του user interface του Twister.

Είναι σημαντικό να αναφέρουμε ότι για να μπορέσει να λειτουργήσει η συγκεκριμένη πλατφόρμα πρέπει να συνδεθεί με μια οποιαδήποτε σχεσιακή βάση δεδομένων, κάποιες από τις οποίες περιγράφονται παρακάτω. Επίσης διαθέτει ενσωματωμένη μια ακόμα XML βάση δεδομένων την XINDICE.

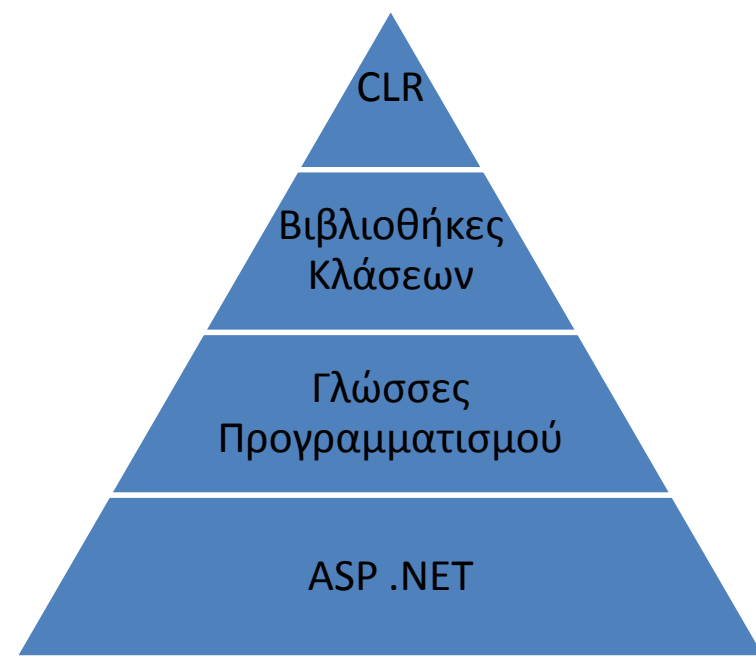
7.1.2 NET Framework

Το NET Framework είναι μια βιβλιοθήκη για υπολογιστές με λειτουργικό σύστημα Windows και υποστηρίζει την ανάπτυξη και λειτουργία εφαρμογών και Υπηρεσιών Διαδικτύου. Αποτελεί ουσιαστικά μια πλατφόρμα πάνω στην οποία μπορούμε να δημιουργήσουμε κάθε είδους εφαρμογές. Παρέχει τη δυνατότητα στους προγραμματιστές να χρησιμοποιούν διάφορες εξελιγμένες λειτουργίες και η εφαρμογή του να είναι συμβατή με όλα τα συστήματα που υποστηρίζουν και έχουν εγκατεστημένο το NET Framework.

Ουσιαστικά το NET Framework σχεδιάστηκε για να εκπληρώσει τους ακόλουθους στόχους.

- Να παρέχει ένα σταθερό και πιο απλουστευμένο περιβάλλον ανάπτυξης για όλα τα είδη εφαρμογών (Windows-based Web-based εφαρμογές) καθώς και Υπηρεσιών Web.
- Βελτιώνει το βαθμό ασφάλειας των εφαρμογών των Windows, καθιστώντας τις έτσι πιο αξιόπιστες.
- Παρέχει ένα σύνολο από βιβλιοθήκες, οι οποίες μπορούν να λειτουργούν με πολλές γλώσσες.

Στο ακόλουθο σχήμα απεικονίζεται η βασική του δομή. Ουσιαστικά αποτελείται από τέσσερα (4) στοιχεία. Το Common Language Runtime (CLR), μια βιβλιοθήκη κλάσεων, ένα σύνολο από γλώσσες προγραμματισμού και από την ASP NET.



Σχ.6 Δομή ενός NET Framework

Common Language Runtime (CLR)

Οι γλώσσες προγραμματισμού συνήθως αποτελούνται από έναν compiler-μεταγλωττιστή και ένα runtime περιβάλλον. Κατά την διάρκεια της μεταγλώττισης, ο μεταγλωττιστής μετατρέπει τον εν λόγω κώδικα σε ενδιάμεσο κώδικα (byte code) που ονομάζεται Common Intermediate Language(CIL), ο οποίος ουσιαστικά ορίζει οδηγίες για το (CLR), που λειτουργεί ως εικονική μηχανή. Στην συνέχεια κατά την διάρκεια της εκτέλεσης ο μεταγλωττιστής του (CLR) μετατρέπει τον (CIL) κώδικα σε κώδικα μηχανής.

Το περιβάλλον παρέχει ένα σύνολο υπηρεσιών του λειτουργικού συστήματος στον εκτελέσιμο κώδικα. Οι υπηρεσίες αυτές είναι ενσωματωμένες σε ένα επίπεδο που επιτρέπει στον κώδικα να μην ασχολείται με λεπτομέρειες χαμηλού επιπέδου του λειτουργικού συστήματος. Τέτοιες λειτουργίες μπορεί να είναι η διαχείριση μνήμης, εγγραφή και ανάγνωση αρχείων κλπ.

Πριν το NET Framework κάθε γλώσσα διέθετε το δικό της runtime περιβάλλον, το καθένα από τα οποία όμως υποστηριζόταν από μια συγκεκριμένη γλώσσα, με αποτέλεσμα να μην μπορούν να χρησιμοποιηθούν λειτουργίες από το περιβάλλον μιας γλώσσας σε άλλη. Με την δημιουργία του NET Framework, ενοποιήθηκαν όλα τα runtime περιβάλλοντα, ώστε οι προγραμματιστές να μπορούν να χρησιμοποιούν μόνο ένα περιβάλλον. Πιο συγκεκριμένα η λύση δόθηκε από το (CLR), το οποίο παρέχει δυνατότητες όπως διαχείριση μνήμης, ασφάλεια κλπ. για κάθε γλώσσα που δουλεύει με το NET Framework .

Βιβλιοθήκες Κλάσεων

Είναι μια ολοκληρωμένη συλλογή από επαναχρησιμοποιούμενους τύπους που μπορούν να χρησιμοποιηθούν για την ανάπτυξη εφαρμογών. Στην ουσία το NET Framework χρησιμοποιεί πολλές κλάσεις για να βοηθήσει τους προγραμματιστές στην επαναχρησιμοποίηση κώδικα. Οι βιβλιοθήκες NET Class Libraries περιέχουν κώδικα για προγραμματιστικά θέματα όπως εγγραφή αρχείων, μετατροπή σε XML,

δομές δεδομένων κλπ., ενώ είναι διαθέσιμες σε κάθε γλώσσα που λειτουργεί με NET Framework.

Γλώσσες προγραμματισμού

Η Microsoft παρέχει ένα σύνολο γλωσσών που είναι ήδη συμβατές με το NET. Η C είναι μια από αυτές, ενώ δημιουργήθηκαν νέες εκδόσεις των Visual Basic, Visual C ++ , JscriptNET. Ωστόσο η Microsoft προέβη στην δημοσίευση μιας πλήρους τεκμηρίωσης του πως οι κατασκευαστές γλωσσών μπορούν να κάνουν τις γλώσσες τους συμβατές με το NET. Έτσι σήμερα υπάρχουν πάνω από 20 γλώσσες τρίτων κατασκευαστών που είναι συμβατές και αυτές με την NET.

ASP-NET

Το ASP δημιουργήθηκε από την Microsoft και αποτελεί ένα σύνολο τεχνολογιών για την δημιουργία εφαρμογών διαδικτύου και Υπηρεσιών Διαδικτύου. Ουσιαστικά αποσκοπεί στην πρόσθεση δυναμικών ιδιοτήτων στις web σελίδες. Αυτό το πέτυχε με προγραμματισμό πίσω από την web σελίδα, κυρίως σε VB Script. Όταν οι χρήστες επισκέπτονταν ένα site, τους ζητούνταν να επιβεβαιώσουν κάποιες πληροφορίες και στην συνέχεια ένα Script δημιουργούσε τη σελίδα ανάλογα με τα στοιχεία που εισήγαγε ο χρήστης.

7.1.3 Web server

Ένας από τους πιο διαδεδομένους Web Server που χρησιμοποιείται κατά την υλοποίηση Web Services είναι ο Tomcat, ο οποίος αποτελεί έναν από τους πιο διαδεδομένους διακομιστές εφαρμογών που κυκλοφορούν στην αγορά λογισμικού. Ουσιαστικά χρησιμοποιείται ως μια πλατφόρμα που περιέχει κλάσεις για υποστήριξη των servlets. Υπερτερεί έναντι άλλων Web Server ως προς τα εξής:

- Διανέμεται δωρεάν μέσω του World Wide Web(WWW), ο οποίος παρέχει δωρεάν και μια πλήρη τεκμηρίωση όσον αφορά το Tomcat.
- Αποτελεί μια ανοιχτού κώδικα υλοποίηση της τεχνολογίας των σελίδων διακομιστών JAVA (Servlet και JSP).
- Αποτελεί έναν αρκετά σταθερό Web Server.

Για την εγκατάσταση του συγκεκριμένου προγράμματος κατεβάζουμε το αρχείο εγκατάστασης από το επίσημο site του Tomcat <http://tomcat.apache.org/>. Αφού τρέξουμε το αρχείο και εγκαταστήσουμε το πρόγραμμα, στην συνέχεια εισχωρούμε στην αρχική σελίδα του Tomcat πληκτρολογώντας την διεύθυνση <http://localhost:8080/index.html>.

7.1.4 Apache Axis

Το Apache Axis αποτελεί μια υλοποίηση ανοιχτού κώδικα της μηχανής SOAP. Δημιουργήθηκε από τον οργανισμό Apache και αρχικά πήρε την ονομασία Apache SOAP. Στην συνέχεια μετεξελέχθηκε σε περιβάλλον ανάπτυξης Java παίρνοντας την τελική του ονομασία Apache Axis. Το συγκεκριμένο εργαλείο

προσφέρει ένα πλαίσιο για την κατασκευή επεξεργαστών SOAP, όπως πελάτες και διακομιστές. Συγκεκριμένα, το Axis είναι μια υλοποίηση ενός διακομιστή που τοποθετεί στις μηχανές servlet έτσι ώστε οι υπηρεσίες Ιστού να μπορούν να φιλοξενοούνται χρησιμοποιώντας τους παραδοσιακούς διακομιστές Ιστού. Επίσης, το Axis υποστηρίζει το πρότυπο WSDL και δημιουργεί ένα αρχείο WSDL για κάθε υπηρεσία Ιστού που αναπτύσσεται.

Για να εκτελεστεί το Axis πρέπει στον υπολογιστή μας να έχουμε εγκαταστήσει την βασική έκδοση της J2SE (Java 2 platform Standard Edition) με την εργαλειοθήκη ανάπτυξης λογισμικού SDK (Software Development Kit). Επίσης, πρέπει στον υπολογιστή να έχουμε εγκαταστήσει ένα διακομιστή εφαρμογών κατά προτίμηση τον TOMCAT καθώς επίσης και έναν σαρωτή XML (MML Parser). Πιο συγκεκριμένα για την εγκατάσταση του Apache Axis ανατρέχουμε στη σελίδα <http://www.apache.org/axis>.

7.1.5 Συστήματα Διαχείρισης Βάσης Δεδομένων

Με τον όρο σχεσιακή βάση δεδομένων εννοείται μία συλλογή δεδομένων οργανωμένη σε συσχετισμένους πίνακες που παρέχει ταυτόχρονα ένα μηχανισμό για ανάγνωση, εγγραφή, τροποποίηση ή και πιο πολύπλοκες διαδικασίες πάνω στα δεδομένα. Ο σκοπός μιας βάσης δεδομένων είναι η οργανωμένη αποθήκευση πληροφορίας και η δυνατότητα εξαγωγής της πληροφορίας αυτής, ιδίως σε πιο οργανωμένη μορφή, σύμφωνα με ερωτήματα που τίθενται στη σχεσιακή βάση δεδομένων. Τα δεδομένα είναι δυνατόν να αναδιοργανώνονται με πολλούς διαφορετικούς τρόπους, σε νοητούς πίνακες, χωρίς να είναι απαραίτητη η αναδιοργάνωση των φυσικών πινάκων που τα αποθηκεύουν.

Οι ερωτήσεις, είτε από το χρήστη είτε από λογισμικό, προς τη βάση δεδομένων, γίνονται συνήθως μέσω της διαδεδωμένης διαλογικής γλώσσας SQL (Structured Query Language). Εκτελώντας ερωτήματα ο χρήστης (ή το λογισμικό που εκπροσωπεί το χρήστη) είναι δυνατόν, ανάλογα με τα δικαιώματά του, να δημιουργήσει, να μεταβάλλει και να διαγράψει δεδομένα στη βάση, ή να ανασύρει πληροφορίες με σύνθετα κριτήρια αναζήτησης

Μέχρι στιγμής έχει αναπτυχθεί μια πληθώρα συστημάτων διαχείρισης βάσης δεδομένων όσον αφορά τις Υπηρεσίες Web. Στην παρούσα διπλωματική θα αναφερθούμε στην MicrosoftSQL και στην PostgreSQL, οι οποίες αποτελούν σήμερα τις πιο διαδεδωμένες γλώσσες διαχείρισης σχεσιακών βάσεων δεδομένων.

MicrosoftSQL

Το SQL Server έχει αναπτυχθεί και προωθείται από την Microsoft. και αποτελεί ίσως ένα από τα πιο εμπορικά ΣΔΒΔ του σήμερα. Υποστηρίζει πλήρως την περιγραφή, ανεξαρτησία και προστασία των δεδομένων ενώ διαθέτει μηχανισμούς ταυτόχρονης προσπέλασης και επεκτασιμότητας. Πιο συγκεκριμένα παρέχει δυνατότητες για:

- τον ορισμό, τη διαγραφή και τη μεταβολή πινάκων και κλειδίων,
- τη σύνταξη ερωτήσεων (*queries*),
- την εισαγωγή, διαγραφή και μεταβολή στοιχείων,

- τον ορισμό όψεων (*views*) πάνω στα δεδομένα,
- τον ορισμό δικαιωμάτων πρόσβασης,
- τον έλεγχο της ακεραιότητας των στοιχείων,
- τον έλεγχο συναλλαγών (*transaction*)

Ο SQL Server πέρα των βασικών λειτουργιών παρέχει και κάποιες πρόσθετες υπηρεσίες οι οποίες αν και δεν είναι απαραίτητες , αποτελούν υπηρεσίες προστιθέμενης αξίας πάνω από το κεντρικό σύστημα διαχείρισης βάσης δεδομένων. Κάποιες από αυτές τις λειτουργίες είναι οι εξής:

- Replication Services: Χρησιμοποιούνται για την αντιγραφή και τον συγχρονισμό αντικειμένων της βάσης δεδομένων.
- Analysis Services: Προσφέρει την on line επεξεργασία δεδομένων και υποστηρίζει την λειτουργία εξόρυξης δεδομένων για εφαρμογές business intelligence.
- Reporting Services: Επιβλέπει δεδομένα που έχουν συγκεντρωθεί από μια SQL Server βάση δεδομένων.
- Integration Services: Χρησιμοποιείται για την ενσωμάτωση δεδομένων που προέρχονται από δεδομένα διαφορετικών προελεύσεων.

Η κεντρική βάση δεδομένων του SQL υποστηρίζει διαφορετικούς τύπους, συμπεριλαμβανομένων των ακεραίων αριθμών, αριθμών κινητής υποδιαστολής, δεκαδικών, αλφαριθμητικών, Varchar (σειρές χαρακτήρων μεταβλητού μήκους), δυαδικών αριθμών (για τα μη δομημένα δεδομένα), κειμένων (για κείμενα). Επιτρέπει επίσης καθορισμένους από το χρήστη σύνθετους τύπους δεδομένων (UDTs), δηλαδή τύπους που βασίζονται στους βασικούς τύπους αλλά μπορούν να τροποποιηθούν.

Τα στοιχεία στη βάση δεδομένων αποθηκεύονται σε ένα (ή περισσότερα) αρχεία με επέκταση .mdf. Τα δευτεροβάθμια στοιχεία αποθηκεύονται στο αρχείο με επέκταση .ndf. Το αρχείο καταγραφής το οποίο περιέχει όλες τις πρόσφατες αλλαγές στη βάση δεδομένων αποθηκεύεται σε αρχείο με επέκταση .ldf. Ο χώρος αποθήκευσης που διατίθεται σε μια βάση δεδομένων διαιρείται σε διαδοχικά αριθμημένες σελίδες, κάθε μία από τις οποίες έχει μέγεθος 8 KB.

PostgreSQL

Η PostgreSQL είναι μια σχεσιακή βάση δεδομένων ανοικτού κώδικα με πολλές δυνατότητες. Η ανάπτυξη της διαρκεί ήδη πάνω από δύο δεκαετίες και βασίζεται σε μια αποδεδειγμένα καλή αρχιτεκτονική η οποία έχει δημιουργήσει μια ισχυρή αντίληψη των χρηστών της γύρω από την αξιοπιστία, την ακεραιότητα δεδομένων και την ορθή λειτουργία.

Η PostgreSQL τρέχει σε όλα τα βασικά λειτουργικά συστήματα, στα οποία περιλαμβάνονται το Linux, το UNIX (AIX, BSD, HP-UX, SGI, IRIX) και τα WINDOWS. Συμπεριλαμβάνει τους περισσότερους SQL92 και SQL99 τύπους δεδομένων συμπεριλαμβανομένων INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL και TIMESTAMP. Επίσης έχει την δυνατότητα αποθήκευσης μεγάλων δυαδικών αντικειμένων (binary), όπως εικόνες, ήχοι ή βίντεο. Διαθέτει επίσης περιβάλλοντα προγραμματισμού για τις γλώσσες προγραμματισμού

C, C++, JAVA, Perl, Python, Ruby, και υποστήριξη για την πλατφόρμα NET και το πρότυπο ODBC.

Ένα από τα σημαντικότερα της πλεονεκτήματα είναι η δυνατότητα αποθήκευσης πολλής μεγάλης ποσότητας δεδομένων ενώ υποστηρίζει τη χρήση της από πολλούς χρήστες ταυτόχρονα. Εκτός όμως από λειτουργικής άποψης η συγκεκριμένη σχεσιακή βάση δεδομένων προτιμάται και λόγω των κανόνων που τη διέπουν σαν λογισμικό, καθώς αποτελεί ένα open source λογισμικό με άδεια BSD. Αυτό σημαίνει ότι κάθε χρήστης έχει το δικαίωμα να χρησιμοποιήσει, τροποποιήσει και διανέμει το λογισμικό όπως επιθυμεί. Στην ιστοσελίδα <http://www.postgresql.org> διανέμονται δωρεάν οι πιο πρόσφατες εκδόσεις της συγκεκριμένης σχεσιακής βάσης δεδομένων.

7.1.6 Τρόποι επικοινωνίας με την Βάση Δεδομένων

Για την επικοινωνία με το Σύστημα Διαχείρισης Βάσεων Δεδομένων, ένα ευρέως χρησιμοποιούμενο πρωτόκολλο είναι το JDBC (Java DataBase Connectivity). Το JDBC είναι ιδιαίτερα απλό, ευέλικτο και αποτελείται από κλάσεις και διαπροσωπίες (interfaces) που επιτρέπουν την επεξεργασία των δεδομένων μιας οποιασδήποτε Βάσης Δεδομένων με τη χρήση ενσωματωμένης SQL, καθώς επίσης και στην επισκόπηση των μετά-δεδομένων της.

Για να μπορέσει μια συγκεκριμένη Βάση Δεδομένων να επικοινωνήσει μέσω JDBC θα πρέπει να υλοποιηθούν οι κατάλληλοι οδηγοί (drivers), κλάσεις δηλαδή, που υλοποιούν τα interfaces και τις αφηρημένες κλάσεις που δηλώνονται από το JDBC API.

Μια άλλη εξίσου σημαντική εφαρμογή είναι το Hibernate, μια υψηλής απόδοσης αντικειμενοστραφής οντότητα. Η συγκεκριμένη εφαρμογή επιτρέπει την ερώτηση της βάσης με την δικιά της γλώσσα (HQL), μια εξίσου αντικειμενοστραφής γλώσσα που παρουσιάζει πολλές ομοιότητες με την SQL.

Το Hibernate παρέχει την δυνατότητα απευθείας ερώτησης στην βάση δεδομένων καθώς και επιστροφή δεδομένων στο χρήστη, απαλλάσσοντας τον έτσι από την ταλαιπωρία της σύνδεσης κάθε εφαρμογής με τη βάση δεδομένων και την επικοινωνία μαζί της. Πιο αναλυτικά όταν το Hibernate επιθυμεί να επιτελέσει μια λειτουργία εγγραφής ή ανάγνωσης με τη βάση δεδομένων, πρέπει να είναι σε θέση να γνωρίζει πού να κάνει την καταχώρηση ή από πού να διαβάσει. Για αυτή τη λειτουργία υπάρχει το Hibernate mapping αρχείο, το οποίο αναζητά και βρίσκει τον κατάλληλο πίνακα της βάσης που πρέπει να χρησιμοποιηθεί καθώς και την κατάλληλη στήλη αυτού του πίνακα.

Σημαντική σημείωση είναι ότι η συγκεκριμένη εφαρμογή είναι μεταφέρσιμη σε οποιαδήποτε βάση δεδομένων χρησιμοποιεί SQL ενώ μπορούμε να την βρούμε στην διεύθυνση <http://www.hibernate.org>

7.2 ΤΕΧΝΟΛΟΓΙΕΣ

7.2.1 Java

Η Java είναι μία γλώσσα προγραμματισμού υψηλού επιπέδου. Αναπτύχθηκε από την εταιρεία Sun microsystems και γρήγορα έλαβε μεγάλη δημοσιότητα λόγω της ανεπανάληπτης χρησιμότητάς της στην ανάπτυξη (κυρίως Web) εφαρμογών. Ένα από τα βασικότερα πλεονεκτήματα της έναντι των περισσότερων άλλων γλωσσών είναι η ανεξαρτησία του λειτουργικού συστήματος και πλατφόρμας. Ουσιαστικά όλα τα προγράμματα τα οποία είναι γραμμένα σε Java μπορούν να είναι κατανοητά από κάθε υπολογιστή ανεξάρτητα του είδους επεξεργαστή και του λειτουργικού του συστήματος εξαιτίας της Java Virtual Machine.

Κάθε πρόγραμμα που γράφεται σε Java, μεταγλωττίζεται μέσω του μεταγλωττιστή Javac σε μια ενδιάμεση μορφή κώδικα την bytecode. Στην συνέχεια όταν ξεκινά η εκτέλεση του συγκεκριμένου προγράμματος, η Java Virtual Machine, η οποία βρίσκεται εγκατεστημένη σε αυτό, αναλαμβάνει να διαβάσει τα αρχεία bytecode και να τα μετατρέψει σε γλώσσα μηχανής, η οποία να υποστηρίζεται από τον συγκεκριμένο επεξεργαστή και το λειτουργικό σύστημα. Με αυτόν τον τρόπο λοιπόν κάθε εφαρμογή αυτής της γλώσσας μπορεί να τρέξει σε οποιαδήποτε πλατφόρμα χωρίς καμία αλλαγή.

Κάποια από τα βασικότερα χαρακτηριστικά της, τα οποία την καθιστούν την πλέον χρησιμοποιούμενη γλώσσα κατά την υλοποίηση των Web Services είναι:

- Η συγκεκριμένη γλώσσα είναι μια καθαρά Αντικειμενοστραφής γλώσσα, το οποίο την διευκολύνει στην ανάπτυξη επαναχρησιμοποιούμενου κώδικα.
- Παράγει σταθερό και αξιόπιστο κώδικα ,διαθέτοντας μηχανισμούς ελέγχου λαθών, φροντίζει για την αποδέσμευση της μνήμης κλπ.
- Έχει δυναμικό χαρακτήρα καθώς μπορεί να φορτώνει δυναμικά μια κλάση στον μεταφραστή Java οποιαδήποτε χρονική στιγμή κατά την εκτέλεση.
- Συνοδεύεται από πλήρη και δωρεάν τεκμηρίωση μέσω του World Wide Web

7.2.2 Πακέτο Δημιουργίας Web Service

Ως πακέτο για τη δημιουργία Web Services το πιο διαδεδομένο και ευρέως χρησιμοποιούμενο είναι το Java Web Services Developer Pack (JWSDP) της εταιρείας Sun Microsystems. Περιλαμβάνει ένα πλήθος τεχνολογιών, οι οποίες κάνουν πολύ πιο εύκολη και γρήγορη την κατασκευή Web Services, χρησιμοποιώντας την Java.

Το βασικό API του JWSDP είναι το JAX-RPC, το οποίο είναι μια συντομογραφία του Java API for XML-based RPC. Το API (Application Programming Interface) αποτελεί έναν πηγαίο κώδικα και περιλαμβάνει προδιαγραφές για ρουτίνες, δομές δεδομένων κλάσεις αντικειμένων και μεταβλητών. Χρησιμοποιείται για την κατασκευή τόσο των Web Services όσο και των clients που τις καλούν χρησιμοποιώντας remote procedure calls (RPC) και XML.

Το JAX-RPC χρησιμοποιεί το SOAP σαν το πρωτόκολλο μεταφοράς μηνυμάτων. Παρόλο που το JAX-RPC στηρίζεται σε αρκετά πολύπλοκα πρωτόκολλα, το API του κρύβει αυτή την πολυπλοκότητα από τον σχεδιαστή μιας εφαρμογής. Από τη μεριά του server, ο σχεδιαστής της εφαρμογής αρκεί να ορίσει τις remote procedures με το να ορίσει τις μεθόδους σε ένα interface γραμμένο σε Java. Ο σχεδιαστής επίσης γράφει τον κώδικα μίας ή και περισσότερων κλάσεων που υλοποιούν αυτές τις μεθόδους. Αλλά και η σχεδίαση του client είναι αρκετά εύκολη. Ένας client δημιουργεί ένα proxy, ένα τοπικό object που παριστάνει την service, και μετά απλά καλεί τις μεθόδους στον proxy.

Με το JAX-RPC οι clients και οι Web Services έχουν ένα μεγάλο πλεονέκτημα – η Java είναι μια ανεξάρτητη πλατφόρμας προγραμματιστική γλώσσα. Επιπλέον το JAX-RPC δεν είναι περιοριστικό. Ένας JAX-RPC client μπορεί να έχει πρόσβαση σε μια Web Service η οποία δεν τρέχει σε πλατφόρμα που χρησιμοποιεί Java και αντίστροφα. Αυτό οφείλεται στο ότι το JAX-RPC χρησιμοποιεί τεχνολογίες ορισμένες από το World Wide Consortium (W3C), όπως του HTTP, του SOAP και του WSDL.

7.2.3 Servlet

Κάθε servlet αποτελεί μια εφαρμογή που εκτελείται σε ένα server ο οποίος βρίσκεται συνδεδεμένος στο World Wide Web. Ουσιαστικά δηλαδή κάθε servlet λειτουργεί σαν client για το web service, στέλνοντας μηνύματα SOAP και λαμβάνοντας ως απάντηση μηνύματα SOAP. Ένα από τα σημαντικότερα πλεονεκτήματα αυτής της εφαρμογής είναι ότι κάνει χρήση της χωρίς να απαιτείται η λειτουργία περιορισμών ασφαλείας που ισχύουν στις περισσότερες μικροεφαρμογές. Κάποιες από τις σημαντικότερες λειτουργίες των servlet είναι οι εξής:

- Άμεση επικοινωνία με server μέσω χρήσης πρότυπης διασύνδεσης.
- Συλλογή δεδομένων από μια φόρμα σε μια ιστοσελίδα.
- Αποστολή πληροφοριών σε χρήστη του web υπό μορφή HTML εγγράφου είτε κάποια άλλη γνωστή μορφοποίηση.
- Δυνατότητα παρακολούθησης της κίνησης κάθε χρήστη μέσα σε μια ιστοσελίδα, μέσω υποστήριξης για συνόδους-sessions.
- Λήψη πληροφοριών από πεδία σε ένα URL.
- Αποδοτική διαχείριση μνήμης. Αν πολλοί χρήστες κάνουν διαχείριση του servlet, μόνο ένα αντίγραφο του είναι φορτωμένο και αυτό το servlet δημιουργεί νήματα, ώστε να το χειρίζεται κάθε χρήστης.

7.2.4 Δημιουργία Δυναμικών Σελίδων

Για την δημιουργία του client για την επίδειξη της εφαρμογής η πιο αποδοτική και ευρέως διαδεδομένη εφαρμογή είναι αυτή των JSPs (Java Server Pages), η οποία επιτρέπει στους προγραμματιστές την δημιουργία δυναμικών

Παπαργύρη Τριανταφυλλιά

σελίδων, οι οποίες μεταφράζονται κατά την διάρκεια της εκτέλεσης τους. Αποτελεί πολλή σημαντική εφαρμογή στον κόσμο των web services και μόλις μερικά από τα προτερήματα της έναντι άλλων είναι τα εξής:

- Είναι ανεξάρτητες της αρχιτεκτονικής του συστήματος στο οποίο εκτελούνται.
- Έχουν τη δυνατότητα να διαβιβάζουν τις αιτήσεις των χρηστών σε άλλους servers ή JSPs, ενώ ταυτόχρονα υποστηρίζονται από την μεγαλύτερη πλειονότητα των servers που υπάρχουν στην αγορά.
- Παρέχουν όλες τις δυνατότητες των φαινομενικά ισχυρότερων servlets, αφού αυτόματα μεταφράζονται σε servlets που αναλαμβάνουν να φέρουν σε πέρας την απαίτηση.
- Μπορούν να εξυπηρετούν ταυτόχρονα πολλαπλές αιτήσεις διαφορετικών χρηστών και επιπλέον έχουν τη δυνατότητα να τις συγχρονίζουν.
- Δίνουν τη δυνατότητα στον προγραμματιστή να δημιουργήσει πολύ εύκολα τόσο στατικές όσο και δυναμικές οντότητες, με μια πιο φυσική προσέγγιση από άλλες τεχνολογίες.

Συνοπτικά λοιπόν από την μεριά του χρήστη, όταν ο ίδιος επιθυμεί την προσπέλαση μιας web service, αρχικά θα πρέπει να πληκτρολογήσει την κατάλληλη διεύθυνση. Στην συνέχεια θα του παρουσιαστεί το κεντρικό μενού της εφαρμογής σε μια HTML σελίδα που υπάρχει στον συγκεκριμένο server (π.χ Tomcat). Επιλέγοντας ο χρήστης μια συγκεκριμένη λειτουργία της εφαρμογής, καλεί στην συνέχεια το κατάλληλο servlet, το οποίο παίρνει σαν είσοδο τις πιθανές παραμέτρους που ορίζει ο χρήστης, εν συνεχεία επικοινωνεί με την web service και αφού λάβει την απάντηση την επιστρέφει στον χρήστη μέσα σε μια σελίδα JSP.

7.3 ΠΑΡΑΔΕΙΓΜΑΤΑ ΥΛΟΠΟΙΗΣΗΣ WEB SERVICE

ΠΑΡΑΔΕΙΓΜΑ 1

Στο ακόλουθο παράδειγμα παρουσιάζεται η υλοποίηση μιας απλής Υπηρεσίας Web, η οποία μετατρέπει την θερμοκρασία από βαθμούς Κελσίου σε Φαρενάιτ και αντίστροφα, με χρήση ASP.NET. Το συγκεκριμένο παράδειγμα μπορώ να το τρέξω με χρήση NET Server.

```
<% @ WebService Language="VBScript" Class="TempConvert"%>
```

```
Imports System
```

```
Imports System.Web.Services
```

```
Public Class TempConvert :Inherits WebService
```

```
<WebMethod(> Public Function FahrenheitToCelsius
```

```
(ByVal Fahrenheit As String) As String
    Dim fahr
    Fahr=trim(replace(Fahrenheit,"","."))
    If fahr="" or IsNumeric(fahr)=false then return "Error"
    Return (((fahr) - 32) / 9) * 5)
end function

<WebMethod(> Public Function CelsiusToFahrenheit
(ByVal Celsius As String) As String
    dim cel
    cel=trim(replace(Celsius,"","."))
    if cel="" or IsNumeric(cel)=false then return "Error"
    return (((cel) * 9) / 5) + 32)
end function

end class
```

ΕΠΕΞΗΓΗΣΗ

Η Πρώτη γραμμή του παραδείγματος περιγράφει ότι πρόκειται για μια Web Service, γραμμένη σε VBScript και λεχει το όνομα κλάσης TempConvert

Στην επόμενη γραμμή εισάγετε ο ονοματοχώρος System.Web.Services από το NET Framework.

Στην συνέχεια η δήλωση Public Class TempConvert :Inherits WebService, δηλώνει ότι η τάξη TempConvert αποτελεί είδος τάξης Web Service.

Τα επόμενα βήματα αποτελούν την βάση του προγραμματισμού VB.Η συγκεκριμένη εφαρμογή εκτελεί δυο βασικές λειτουργίες. Η πρώτη μετατρέπει τα Φαρενάιτ σε Κελσίου και η δεύτερη το αντίστροφο. Η λειτουργία αυτή ορίζεται ως Web Method, μέσω της οποίας οι λειτουργίες της εφαρμογής μας σε Υπηρεσίες Web.

ΜΕΤΑΤΡΟΠΗ ΑΠΟ ΦΑΡΕΝΑΙΤ ΣΕ ΚΕΛΣΙΟΥ

```
<WebMethod(> Public Function FahrenheitToCelsius
(ByVal Fahrenheit As String) As String
    Dim fahr
    Fahr=trim(replace(Fahrenheit,"","."))
    If fahr="" or IsNumeric(fahr)=false then return "Error"
    Return (((fahr) - 32) / 9) * 5)
```

end function

ΜΕΤΑΤΡΟΠΗ ΑΠΟ ΚΕΛΣΙΟΥ ΣΕ ΦΑΡΕΝΑΙΤ

```
<WebMethod(> Public Function CelsiusToFahrenheit  
(ByVal Celsius As String) As String  
    dim cel  
    cel=trim(replace(Celsius,",","."))  
    if cel="" or IsNumeric(cel)=false then return "Error"  
    return (((cel) * 9) / 5) + 32)  
end function
```

Η εφαρμογή μας λήγει με τη δήλωση end class.

Τέλος δημοσιεύοντας το αρχείο .asmx σε ένα server με NET Υποστήριξη ,έχουμε μια Υπηρεσία Web.

ΠΑΡΑΔΕΙΓΜΑ 2

Στο παράδειγμα που ακολουθεί αναπτύσσεται μια Υπηρεσία Διαδικτύου κατά την οποία οποιοσδήποτε χρήστης επιθυμεί ενημερώνεται για την διαθεσιμότητα κάποιου βιβλίου που επιθυμεί ώστε στην συνέχεια να μπορεί να προβεί σε ηλεκτρονική παραγγελία.

```
public class KataxwrisiImp(  
    Public void kataxwrise(String ISBN) (  
    )  
)  
Public class InStockImpl {  
    Public String isInStock ( String ISBN ) {  
        String ans="no"  
        String[]ISBNs=new String[20]  
        ISBNs[0]= "0-125-533-545";  
        Int i="0";  
        While( ISBNs[i]!=null ){  
            if (ISBNs[i].equals( ISBN))          ans="yes"  
            if( ans.equals ("yes")) break;  
  
            if( i<ISBNs.length)i++; else return ans;  
  
        }  
  
        return ans  
    }  
}
```


ΕΠΕΞΗΓΗΣΗ

Στην ουσία έχουν δημιουργηθεί δυο web services, οι οποίες συνεργάζονται για την παροχή μιας κοινής λειτουργίας. Κάθε βιβλίο βρίσκεται καταχωρημένο στο ηλεκτρονικό κατάστημα βάση ενός ISBN κωδικού.

Αρχικά λοιπόν δημιουργείται η πρώτη web service με τις εντολές `public class KataxwrisiImp(`

`Public void kataxwrise(String ISBN)`

με τις οποίες καταχωρείται ο συγκεκριμένος κωδικός ISBN του βιβλίου που δίνει ο χρήστης. Στην συνέχεια η δεύτερη web service που ονομάζεται In Stock, παίρνει ως είσοδο το ISBN και αφού ελέγξει στην βάση της την διαθεσιμότητα του βιβλίου στέλνει στον χρήστη την αντίστοιχη απάντηση.

Ο έλεγχος της βάσης δεδομένων πραγματοποιείται με την χρήση της εντολής επαναληπτικού βρόχου while.

```
While( ISBNs[i]!=null ){  
    if (ISBNs[i].equals( ISBN))      ans="yes"  
    if( ans.equals ("yes")) break;  
  
    if( i<ISBNs.length)i++; else return ans;  
  
}
```

Ελέγχονται έτσι όλα τα ISBN της βάσης δεδομένων του συγκεκριμένου ηλεκτρονικού καταστήματος και αν βρεθεί ο συγκεκριμένος κωδικός, στέλνεται στον χρήστη η απάντηση <yes>(δηλ. διαθέσιμο), αλλιώς <no>.

ΚΕΦΑΛΑΙΟ 8

8 ΠΡΟΣΘΕΤΕΣ ΤΕΧΝΟΛΟΓΙΕΣ

Μέχρι αυτό το σημείο έχουμε αναφερθεί στις τρεις σημαντικές τεχνολογίες SOAP, WSDL και UDDI καθώς και στην περιγραφική γλώσσα XML, οι οποίες αποτελούν τον πυρήνα των Web Services. Ωστόσο για να μπορούμε να μιλάμε για μια ολοκληρωμένη κατανεμημένη τεχνολογία πρέπει να μπορέσει να εξασφαλιστεί η ακεραιότητα, η αξιοπιστία και η ασφάλεια στο περιβάλλον των Web Services καθώς οι συναλλαγές. Ακολουθεί μια αναφορά στις βασικές απειλές που δέχονται οι Web Services, οι τρόποι αντιμετώπισης των και η εξασφάλιση της ασφάλειας τους καθώς και τεχνολογία των συναλλαγών.

8.1 ΑΠΕΙΛΕΣ ΣΤΙΣ ΔΙΑΔΙΚΤΥΑΚΕΣ ΥΠΗΡΕΣΙΕΣ

Οι Υπηρεσίες Διαδικτύου αποτελούν μη ελεγχόμενες εφαρμογές, οι οποίες ουσιαστικά ελέγχονται από τους ίδιους τους χρήστες. Ταυτόχρονα αποτελούν εφαρμογές οι οποίες κάνουν χρήση εγγράφων XML μέσω SOAP, τα οποία έγγραφα είναι σε απλό κείμενο και όχι σε δυαδική μορφή. Γι' αυτό το λόγο εναπόκειται καθημερινά σε ένα ευρύ φάσμα επιθέσεων. Κατά την ανταλλαγή πληροφοριών μεταξύ πελάτη-διακομιστή πολλοί είναι αυτοί που προσπαθούν να παρέμβουν συγκεντρώνοντας πληροφορίες αρχικά για τον πάροχο της web service. Μέσω του UDDI Registry κάθε "εισβολέας" συλλέγει όλες τις αρχικές λεπτομέρειες που χρειάζεται ενώ στην συνέχεια αποκτώντας ένα αντίγραφο του WSDL αποκτά μια πλήρη εικόνα για τον τρόπο με τον οποίο μπορεί να αλληλεπιδράσει με την συγκεκριμένη Υπηρεσία. Από αυτό το σημείο και έπειτα ο "εισβολέας" επιλέγει με ποιόν τρόπο θα επέμβει στην συγκεκριμένη εφαρμογή αλλοιώνοντας την. Κάποιοι από τους τρόπους είναι οι εξής:

8.1.1 Parameter Tampering - SQL Injection

Ένας από τους βασικότερους στόχους ενός "εισβολέα" είναι να μπορέσει να ρίξει την συγκεκριμένη Υπηρεσία Διαδικτύου και ακόμα περισσότερο να αποκτήσει πρόσβαση σε πληροφορίες που δεν επιτρέπεται. Οι περισσότερες web services διατηρούν κάποιου είδους input validation ελέγχοντας με αυτόν τον τρόπο τυχόν άκυρες εισόδους από ενδεχόμενους clients, εισόδους δηλαδή που δεν πληρούν τις Παπαργύρη Τριανταφυλλιά

προδιαγραφές της συγκεκριμένης Υπηρεσίας. Οι web services που υστερούν στην συγκεκριμένη λειτουργία είναι επιρρεπείς σε Parameter Tampering(Αλλοίωση Παραμέτρων), όπου ο "εισβολέας" ρίχνει το σύστημα υποβάλλοντας μη πρότυπα μοτίβα παραμέτρων. Εκτός αυτού μπορεί αν επιθυμεί να αποκτήσει παράνομη πρόσβαση στα δεδομένα της εφαρμογής. Από την στιγμή που μια web service δεν κάνει σωστό validation στους clients, υποβάλλοντας ειδικούς χαρακτήρες ο εισβολέας (π.χ ερωτηματικό) μπορεί να εισχωρήσει χωρίς άδεια στην βάση δεδομένων της συγκεκριμένης εφαρμογής. Η συγκεκριμένη επίθεση έχει το όνομα SQL Injection.

8.1.2 Soap messages flooding

Ένας άλλος τρόπος παρέμβασης σε μια Υπηρεσία Web κατά την εφαρμογή της, είναι η υπερφόρτωση του συστήματος. Γνωρίζουμε ήδη ότι η αλληλεπίδραση του client με την web service πραγματοποιείται μέσω μηνυμάτων SOAP, τα οποία ελέγχονται στην συνέχεια από τον XML-Parser. Ο "εισβολέας" επιθυμώντας να εμποδίσει την Υπηρεσία Web να συνεχίσει να λαμβάνει ορθά μηνύματα SOAP, στέλνει έγκυρα μεν αλλά συνεχόμενα και πολλαπλά μηνύματα-αιτήσεις SOAP. Αν και τα ίδια τα μηνύματα είναι έγκυρα ωστόσο ο XML-Parser κατά πάσα πιθανότητα αδυνατεί να επεξεργαστεί ένα τόσο μεγάλο όγκο μηνυμάτων με αποτέλεσμα την πιθανή διακοπή λήψης των πραγματικών requests που προέρχονται από τον client της εφαρμογής

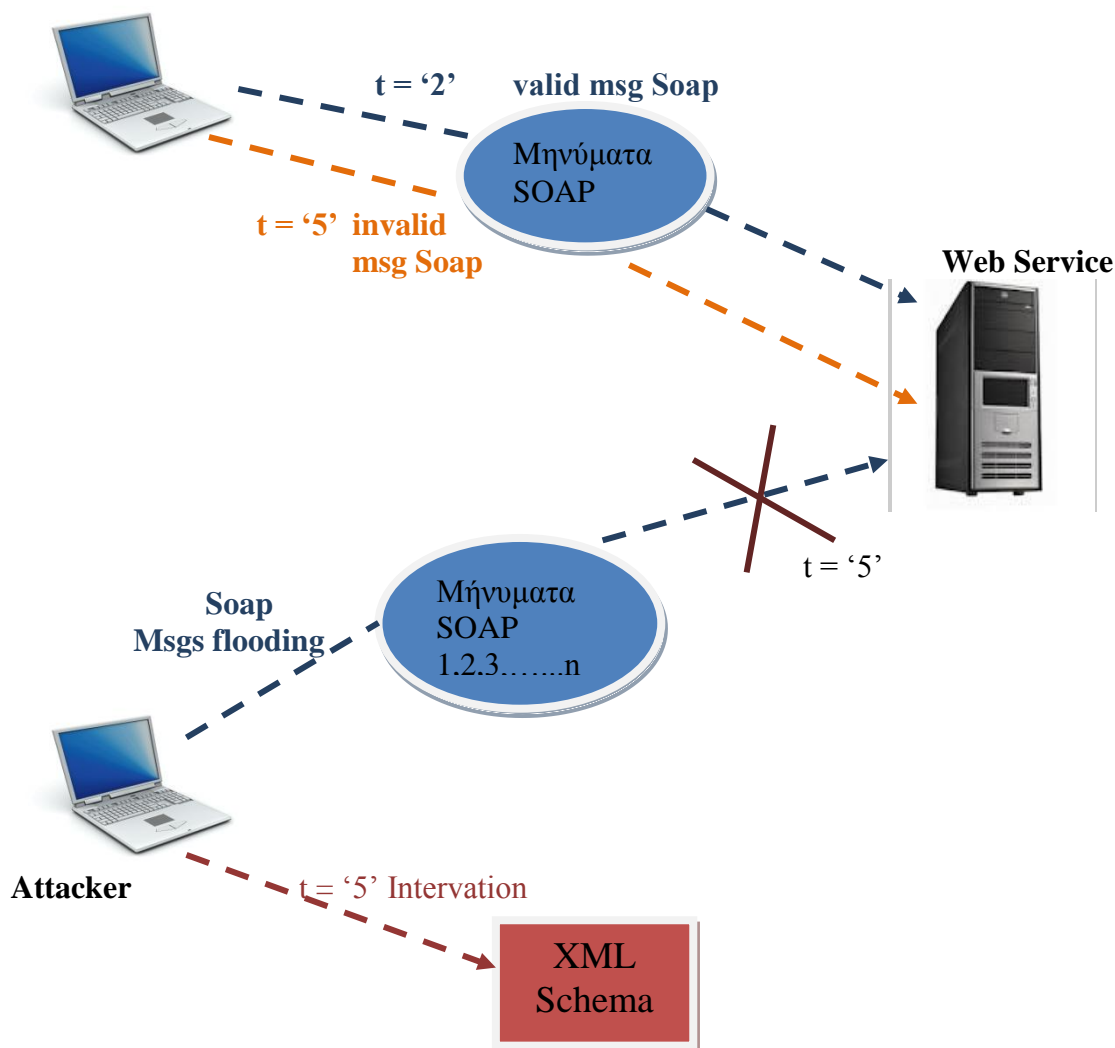
8.1.3 Schema Poisoning

Με τον προηγούμενο τρόπο ο εισβολέας παρεμβαίνει άμεσα στον XML Parser, καθιστώντας τον ανήμπορο να ανταπεξέλθει στα συνεχόμενα και πολλαπλά μηνύματα SOAP. Σε αυτήν την περίπτωση ωστόσο όλα τα απεσταλμένα μηνύματα θεωρούνται έγκυρα. Σε πολλές περιπτώσεις λοιπόν υπάρχει η πιθανότητα ο XML Parser να είναι σε θέση να επεξεργαστεί όλα αυτά τα μηνύματα σε σχετικά σύντομο χρονικό διάστημα με αποτέλεσμα η εφαρμογή να συνεχίσει να λειτουργεί ομαλά.

Ένας λοιπόν πιο αποδοτικός τρόπος απειλής είναι να μπορέσει ο εισβολέας να μετατρέψει όλα τα μηνύματα του client σε μη έγκυρα. Κάθε μήνυμα SOAP, όπως έχει ήδη αναφερθεί, περιέχει την πληροφορία σε μορφή XML εγγράφου. Το XML Schema καθορίζει την απαιτούμενη δομή του κάθε εγγράφου XML ώστε να θεωρείτε έγκυρο. Ο εισβολέας λοιπόν αποσκοπεί στο να επιτύχει Schema poisoning, παρεμβαίνοντας στον κόμβο που βρίσκεται το schema και τροποποιώντας το. Ως εκ τούτου όλα τα εισερχόμενα μηνύματα SOAP παρουσιάζονται ως άκυρα εφόσον δεν υπακούουν στο τροποποιημένο πλέον schema

Ακολουθεί εικονική αναπαράσταση των δυο αναφερόμενων τελευταίων μεθόδων.

client



Σχ.7 Soap messages flooding-Schema Poisoning

ΕΠΕΞΗΓΗΣΗ ΣΧΗΜΑΤΟΣ

Κατά την χρονική στιγμή $t=2$ ο client στέλνει κάποια μηνύματα SOAP στην web service. Ταυτόχρονα έχουμε και κατακλυσμό μηνυμάτων από τον attacker προς την web service εξίσου. Όλα τα μηνύματα είναι έγκυρα, υπακούοντας στο XML Schema. Ωστόσο υπάρχει ο κίνδυνος μη αποδοτικής συμπεριφοράς του XML Parser, εξαιτίας του κατακλυσμού μηνυμάτων. Κατά την χρονική στιγμή $t=5$, ο attacker παρεμβαίνει στο XML Schema, τροποποιώντας το. Τα μηνύματα που στέλνονται από τον client θεωρούνται πλέον άκυρα, ενώ ο εισβολέας σταματά την μέθοδο Soap messages flooding καθώς πλέον επιτίθεται μέσω του Schema Poisoning.

8.1.4 Παρέμβαση κατά την Δρομολόγηση

Σε προηγούμενο κεφάλαιο αναφερθήκαμε στο μοντέλο επεξεργασίας SOAP μηνυμάτων σύμφωνα με το οποίο κάθε μήνυμα το οποίο δρομολογείται από τον

Αρχικό Αποστολέα με σκοπό να φτάσει στον Τελικό Παραλήπτη, μπορεί να περάσει μέσα από ένα σύνολο ενδιάμεσων κόμβων. Αυτό το μοντέλο προσπαθούν να εκμεταλλευτούν οι διάφοροι επικείμενοι εισβολείς των web services. Κάθε εισβολέας μιας web service προσπαθεί να αποκτήσει υπό τον έλεγχο του έναν ή περισσότερους κόμβους της συγκεκριμένης εφαρμογής επιτυγχάνοντας έτσι είτε την δρομολόγηση των μηνυμάτων σε ανύπαρκτους παραλήπτες είτε ακόμα χειρότερα σε διευθύνσεις ελεγχόμενες από τον ίδιο. Με αυτόν τον τρόπο ο εισβολέας γίνεται παράνομος βέβαια παραλήπτης πληροφοριών της συγκεκριμένης Υπηρεσίας Διαδικτύου.

8.1.5 Malicious Contents

Η απειλή αυτή σχετίζεται με δυαδικά συνημμένα των SOAP μηνυμάτων. Κάποιος εισβολέας μπορεί να στοχεύσει στην τροποποίηση κάποιων από τα δυαδικά συνημμένα, όπως εκτελέσιμα αρχεία, δημιουργώντας έτσι προβλήματα μέσα στα web services. Συνημμένα κακόβουλα προγράμματα όπως ιοί, worms, ή Trojan horses μπορούν να μεταδοθούν μέσω μηνυμάτων SOAP σε όλη την αρχιτεκτονική web services

8.2 ΕΞΑΣΦΑΛΙΣΗ ΑΣΦΑΛΕΙΑΣ ΣΤΙΣ WEB SERVICES

Στο προηγούμενο υποκεφάλαιο παρουσιάστηκαν μόλις κάποιοι από τους τρόπους με τους οποίους κάποιος πιθανός εισβολέας μπορεί να παρέμβει και να αλλοιώσει τα αποτελέσματα εφαρμογής μιας Υπηρεσίας Ιστού. Οι απειλές λοιπόν είναι πολλές ενώ ταυτόχρονα είναι πολύ σημαντική η εξασφάλιση μιας όσο το δυνατόν ασφαλούς ανταλλαγής δεδομένων μεταξύ πελάτη-διακομιστή ώστε να μπορούμε να αναφερόμαστε σε ένα αξιόπιστο και ασφαλές περιβάλλον εφαρμογής των Υπηρεσιών Διαδικτύου. Έχουν αναπτυχθεί αρκετοί τρόποι εξασφάλισης αξιοπιστίας άλλοι πιο αποδοτικοί και άλλοι λιγότερο. Επιγραμματικά είναι οι εξής:

- Σύνδεση με HTTP
- Σύνδεση με HTTPS
- Σύνδεση μέσω χρήσης SOAP Security Extensions
 - XML Signature
 - XML Encryption
 - SAML-XACML
 - XKMS

Ακολουθεί μια πιο εκτενής αναφορά στους παραπάνω τρόπους διαφύλαξης ή και ενίσχυσης ασφάλειας στις Υπηρεσίες Διαδικτύου.

8.2.1 Πιστοποίηση για Σύνδεση πάνω από HTTP

Στην συγκεκριμένη περίπτωση απαιτείται όνομα και κωδικός χρήστη ώστε να είναι δυνατή η προσπέλαση του σε Υπηρεσίες Web. Τα συγκεκριμένα στοιχεία πριν τη μεταφορά τους κωδικοποιούνται με χρήση πχ κάποιου αλγόριθμου για να υπάρχει προστασία σε περίπτωση υποκλοπής. Ωστόσο με αυτή τη μέθοδο υπάρχει η αδυναμία κρυπτογράφησης του απεσταλμένου μηνύματος SOAP, καθιστώντας αυτόν τον τρόπο αρκετά επισφαλής.

8.2.2 Πιστοποίηση για σύνδεση πάνω από HTTPS

Στην περίπτωση αυτή γίνεται χρήση ψηφιακών πιστοποιητικών για ασφαλή μετάδοση, ενώ η σύνδεση γίνεται στη θύρα 443, αντί για την 80 στην οποία γίνονται οι συνδέσεις HTTP. Ωστόσο ο συγκεκριμένος τρόπος υστερεί όταν πρόκειται για δρομολόγηση μηνυμάτων SOAP μέσω ενδιάμεσων κόμβων. Σε αυτή τη περίπτωση το HTTPS δεν εξασφαλίζει ότι το μήνυμα δεν αλλοιώθηκε κατά την μετάβαση του σε κάποιον ενδιάμεσο κόμβο.

8.2.3 Ψηφιακή Υπογραφή-XML Signature

Όταν μιλάμε για Υπηρεσίες Διαδικτύου ουσιαστικά αναφερόμαστε σε ανταλλαγή SOAP μηνυμάτων που περιέχουν ευαίσθητα δεδομένα όπως πληροφορίες για πελάτες και συναλλαγές. Γι' αυτόν τον λόγο η εξασφάλιση της απαραίτητης μυστικότητας-εμπιστευτικότητας και ακεραιότητας των μηνυμάτων είναι επιτακτική. Για την επίτευξη των δυο παραπάνω τομέων ασφαλείας δημιουργήθηκε η προδιαγραφή της Ψηφιακής Υπογραφής. Μέσω αυτής της προδιαγραφής έγινε δυνατή η παροχή ασφαλών Υπηρεσιών με τη μορφή ακεραιότητας δεδομένων και πιστοποίησης μηνυμάτων.

Το πρότυπο XML Signature ορίστηκε από το World Web Consortium μαζί με την επιτροπή Internet Engineering Task Force. Πρόκειται ουσιαστικά για ένα έγγραφο XML, με την ιδιότητα της ψηφιακής υπογραφής για κάποιο περιεχόμενο, το οποίο μπορεί να βρίσκεται εντός ή και εκτός του εγγράφου της ψηφιακής υπογραφής. Οι XML ψηφιακές υπογραφές (XML Digital Signatures) είναι ένα πρότυπο για την ασφαλή επικύρωση της προέλευσης των μηνυμάτων (αν δηλαδή τα μηνύματα έχουν δημιουργηθεί από τον κατάλληλο αποστολέα και δεν έχουν τροποποιηθεί κατά την μεταφορά). Η προδιαγραφή της XML υπογραφής επιτρέπει στα έγγραφα XML να υπογραφούν με ένα τυποποιημένο τρόπο, χρησιμοποιώντας διαφορετικούς αλγόριθμους ψηφιακής υπογραφής. Οι ψηφιακές υπογραφές μπορούν να χρησιμοποιηθούν για επικύρωση των μηνυμάτων και για μη-αποποίηση.

Τα στοιχεία μιας XML υπογραφής, όπως φαίνεται είναι τα εξής:

```
<element name="Signature" type="ds:SignatureType"/>
<complexType name="SignatureType">
  <sequence>
```

```

        <element ref="ds:SignedInfo"/>
        <element ref="ds:SignatureValue"/>
        <element ref="ds:KeyInfo" minOccurs="0"/>
        <element ref="ds:Object" minOccurs="0"
            maxOccurs="unbounded"/>
    </sequence>
<attribute name="Id" type="ID" use="optional"/>
</complexType>

```

Το στοιχείο SignedInfo περιλαμβάνει τον αλγόριθμο canonicalization, έναν αλγόριθμο υπογραφής και μια ή περισσότερες αναφορές.

Το στοιχείο SignatureValue περιέχει την πραγματική τιμή της ψηφιακής υπογραφής. Κωδικοποιείται πάντα χρησιμοποιώντας την base64.

Το στοιχείο KeyInfo είναι ένα προαιρετικό στοιχείο που επιτρέπει τον παραλήπτη να λάβει το κλειδί που απαιτείται για να επικυρώσει την υπογραφή. Το KeyInfo μπορεί να περιέχει κλειδιά, ονόματα, πιστοποιητικά και άλλες πληροφορίες διαχείρισης δημόσιου κλειδιού.

Το στοιχείο Object είναι ένα προαιρετικό στοιχείο που μπορεί να εμφανιστεί μια ή περισσότερες φορές. Όταν εμφανίζεται, αυτό το στοιχείο μπορεί να περιέχει οποιαδήποτε δεδομένα. Το στοιχείο Object μπορεί να περιλαμβάνει τον τύπο MIME, την ταυτότητα και τα χαρακτηριστικά κωδικοποίησης.

8.2.4 Ψηφιακή Κρυπτογράφηση-XML Encryption

Η συγκεκριμένη προδιαγραφή αποσκοπεί εξίσου με την Ψηφιακή Υπογραφή, στην εξασφάλιση της εμπιστευτικότητας και της ακεραιότητας των ανταλλασσόμενων δεδομένων. Σε αντίθεση ωστόσο με την ψηφιακή υπογραφή εδώ κρυπτογραφείτε ολόκληρο το μήνυμα. Στην συνέχεια αποκρυπτογραφείται στον πρώτο κόμβο και είναι ανοικτό για επισκόπηση προτού κρυπτογραφηθεί πάλι συνολικά για την συνέχιση της διαδρομής.

Ουσιαστικά αυτό το πρότυπο μπορεί να κρυπτογραφεί ανεξάρτητα τα διάφορα στοιχεία ενός XML εγγράφου και να εξασφαλίζει ότι τα κρυπτογραφημένα δεδομένα, κατά τη μεταφορά ή την αποθήκευση, δεν θα μπορούν να προσπελασθούν από μη εξουσιοδοτημένα πρόσωπα. Επίσης πολύ σημαντικό είναι ότι με χρήση ψηφιακής κρυπτογράφησης διατηρείται η ασφάλεια των δεδομένων όχι μόνο όταν τα δεδομένα μεταφέρονται, αλλά και όταν είναι σε στάση σε έναν συγκεκριμένο κόμβο.

8.2.5 SAML-XACML

Η SAML ορίστηκε από την OASIS XML-Based Security Services Technical Committee (SSTC). Αποτελεί τη βασική γλώσσα υλοποίησης του προτύπου PMI, το οποίο είναι υπεύθυνο για την ταυτοποίηση χρηστών σε ένα υπολογιστικό περιβάλλον. Ουσιαστικά μέσω της SAML εφαρμόζονται συγκεκριμένες πολιτικές

ασφαλείας μέσω των οποίων αποφασίζεται σε τι μπορεί να έχει πρόσβαση κάθε χρήστης ενός δικτύου. Η πιο συνήθης περιγραφική γλώσσα των πολιτικών ασφαλείας και της πιστοποίησης χρηστών βάση κριτηρίων είναι η XACML.

8.2.6 XKMS

Όλες οι παραπάνω προδιαγραφές αποτελούν σημαντικούς τρόπους εξασφάλισης της ασφάλειας κατά την εφαρμογή Υπηρεσιών Διαδικτύου. Ωστόσο είναι σαφές ότι επιβάλεται η διαφύλαξη όλων αυτών των κλειδιών ταυτοποίησης, ψηφιακών υπογραφών και πιστοποιητικών σε περιβάλλον αξιόπιστο και οργανωμένο και ασφαλή.

Η XKMS ορίστηκε από τον W3C, με σκοπό να παρέχει ένα τυποποιημένο σύνολο XML ορισμών για τη διαχείριση των υπηρεσιών πιστοποίησης, κρυπτογράφησης και ψηφιακών υπογραφών. Αυτό επιτρέπει την ύπαρξη μιας έμπιστης ουδέτερης οντότητας που βρίσκει και παρέχει τα κατάλληλα κλειδιά και πιστοποιητικά. Αυτή η έμπιστη τρίτη οντότητα ενεργεί σαν μεσάζοντας, ο οποίος απελευθερώνει τον προγραμματιστή της υπηρεσίας ιστού από την υποχρέωση να ελέγχει τη διαθεσιμότητα των κλειδιών ή των πιστοποιητικών και να εξασφαλίζει την εγκυρότητά τους.

Η XKMS περιλαμβάνει δυο μέρη: την XML Key Information Service Specification (X-KISS) και την XML Key Registration Service Specification (X-KRSS). Και οι δύο αυτές προδιαγραφές βασίζονται στη γλώσσα XML, χρησιμοποιούν το SOAP και οι σχέσεις μεταξύ των μηνυμάτων καθορίζονται από την WSDL.

8.2.7 Υπό εξέλιξη προδιαγραφές

Οι εταιρείες IBM και Microsoft συνεργάστηκαν για να αναπτύξουν ένα σύνολο από προδιαγραφές ασφάλειας, οι οποίες απευθύνονται στο πώς παρέχεται η προστασία στα ανταλλασσόμενα μηνύματα σε ένα περιβάλλον υπηρεσίας ιστού. Αυτές οι προδιαγραφές καλύπτουν τεχνολογίες ασφαλείας συμπεριλαμβανομένων της ακεραιότητας, της εμπιστευτικότητας, της αυθεντικοποίησης, της εξουσιοδότησης, των ασφαλών διαδρομών επικοινωνίας κλπ. Μερικές από αυτές είναι:

- WS-Security, η οποία καθορίζει μια αρχιτεκτονική για ασφαλή επικοινωνία.
- WS-Policy, η οποία καθορίζει κανόνες αλληλεπίδρασης υπηρεσιών.
- WS-Trust, η οποία καθορίζει ένα μηχανισμό, ο οποίος λαμβάνει ένα τεκμήριο ασφάλειας και εγκαθιστά μια έμπιστη σχέση μεταξύ ομάδων.
- WS-Privacy, η οποία καθορίζει την εξασφάλιση ιδιωτικότητας στα δεδομένα.

- WS-Secure Conversation, η οποία καθορίζει μηχανισμούς για την εγκατάσταση και την διαμοίραση ασφαλών περιβάλλοντων και παραγωγής κλειδιών, ώστε να καθιστούν ικανή μια ασφαλή επικοινωνία.
- WS-Federation, η οποία καθορίζει τους κανόνες σχετικά με την ταυτότητα σε κατανεμημένο περιβάλλον.

8.3 WEB SERVICES ΚΑΙ ΣΥΝΑΛΛΑΓΕΣ

Μέχρι αυτό το σημείο της διπλωματικής έχουμε αναφερθεί στις βασικές τεχνολογίες των web services, στις μεθόδους επικοινωνίας των πελατών με τις συγκεκριμένες εφαρμογές καθώς και στον τρόπο υλοποίησης τους και εξασφάλισης της ασφαλείας τους. Ωστόσο δεν θα μπορούσαμε να μην αναφερθούμε σε ένα ακόμα σημαντικό χαρακτηριστικό των Υπηρεσιών Διαδικτύου, τις Συναλλαγές, το οποίο έχει αποδειχθεί εξαιρετικά χρήσιμο στο πέρασμα των χρόνων. Οι Συναλλαγές στο περιβάλλον Web Services μας εξασφαλίζουν ότι μια ομάδα από Web Services μπορούν να επικοινωνήσουν μεταξύ τους, συνθέτοντας επιχειρησιακές διαδικασίες και επισκοπώντας έτσι στην επίτευξη ενός κοινού σκοπού.

Για παράδειγμα έστω ότι ένας χρήστης –πελάτης (client) επιθυμεί να προγραμματίσει ένα επαγγελματικό ταξίδι. Θα καλέσει ουσιαστικά μια επιχειρησιακή διαδικασία η οποία επιβάλλεται με την σειρά της να καλέσει διάφορες πρόσθετες υπηρεσίες. Αρχικά θα ζητήσει το όνομα του υπαλλήλου και αρκετές λεπτομέρειες όσον αφορά το ταξίδι όπως ημερομηνία άφιξης-αναχώρησης, προορισμό κλπ. Κατόπιν η επιχειρησιακή διαδικασία θα καλέσει μια υπηρεσία για να ελέγξει την επαγγελματική θέση που ανήκει ο υπάλληλος κατατάσσοντας τον στην ανάλογη κατηγορία. Τέλος θα επικοινωνήσει με web services διαφόρων αεροπορικών εταιρειών, επιλέγοντας το εισιτήριο με το χαμηλότερο κόστος. Έτσι λοιπόν αν και ο πελάτης ουσιαστικά αισθάνεται ότι επικοινωνεί με μια και μόνο Υπηρεσία Διαδικτύου, η οποία και θα τον ενημερώσει για το βέλτιστο αποτέλεσμα, ωστόσο εσωτερικά υπάρχει μια αλληλεπίδραση και συνεργασία μεταξύ κάποιων Υπηρεσιών Διαδικτύου.

8.3.1 BPEL

Η Συναλλαγή των Υπηρεσιών Διαδικτύου με σκοπό την επίτευξη κοινού στόχου οδηγεί σε επιχειρησιακές διαδικασίες. Ωστόσο για να έχουν ισχύ απαιτείται μια τυποποιημένη βάση και μια εξειδικευμένη γλώσσα. Η γλώσσα αυτή πρέπει να παρέχει τη δυνατότητα να εκφραστούν οι επιχειρησιακές διαδικασίες με έναν τυποποιημένο τρόπο, χρησιμοποιώντας συνήθως μια αποδεκτή γλώσσα. Η Γλώσσα Εκτέλεσης Επιχειρηματικών Διαδικασιών BPEL (Business Process Execution Language) είναι μια τέτοια γλώσσα και έχει επικρατήσει ως το κυρίαρχο πρότυπο.

Ο κύριος σκοπός της BPEL είναι μια προτυποποίηση της διαδικασίας αυτοματοποίησης μεταξύ των Υπηρεσιών Διαδικτύου. Αποτελεί και αυτή μια γλώσσα βασισμένη στην XML ενώ ταυτόχρονα στηρίζεται στα πρότυπα XML-Schema και WSDL. Τα μηνύματα WSDL και οι ορισμοί τύπων στο XML-Schema παρέχουν το μοντέλο δεδομένων που χρησιμοποιείται από την BPEL. Ουσιαστικά η BPEL στοχεύει στο να μειώσει την πολυπλοκότητα της σύνδεσης Υπηρεσιών Διαδικτύου και να διευκολύνει τη ολοκλήρωση των επιχειρησιακών διαδικασιών.

Η σημασία της είναι τεράστια καθώς τυχόν απουσία της θα ανάγκαζε τους οργανισμούς να δημιουργήσουν ιδιόκτητα επιχειρησιακά πρωτόκολλα γεγονός που περικλείει την πραγματική συνεργασία μεταξύ των υπηρεσιών ιστού. Παράλληλα μια τέτοια πρακτική θα προκαλούσε αρκετά προβλήματα διαλειτουργικότητας και επομένως θα δυσκόλευε το έργο της ολοκλήρωσης.

Στην ουσία αποτελεί μια γλώσσα <<ενορχήστρωσης>> που καθορίζει τον ρόλο της κάθε υπηρεσίας στην νέα σύνθεση.

8.3.1.1 Δομή της γλώσσας

Η BPEL αποτελεί μια σχετικά απλή σύνθεση εκφράσεων και ετικετών καθώς έχει στόχο να ενορχηστρώσει υπηρεσίες και βασίζεται στην γλώσσα XML. Η βασική της δομή στηρίζεται στα εξής βασικά κομμάτια.

- Δηλώσεις
- Βασικές Δραστηριότητες
- Δομημένες Δραστηριότητες

ΔΗΛΩΣΕΙΣ: Κάθε πρόγραμμα περιλαμβάνεται ανάμεσα στις ετικέτες-δηλώσεις $\langle \rangle$, ενώ στην αρχή του κώδικα δηλώνονται οι συνεργάτες Υπηρεσιών και οι μεταβλητές. Πιο συγκεκριμένα:

- **< process >:** Κάθε διαδικασία μπορεί να είναι σύγχρονη όπου μπλοκάρει τον πελάτη έως ότου τελειώσει η διαδικασία και επιστραφεί το αποτέλεσμα ή ασύγχρονη όπου δεν μπλοκάρει ο πελάτης.
- **<partnerLink>:** Περιγράφει συνεργάτες Υπηρεσιών, τους ρόλους τους και τις λειτουργίες τους σε γλώσσα WSDL.
- **<variable>:** Το συγκεκριμένο στοιχείο περιέχει ένα σύνολο στοιχείων μεταβλητών. Κάθε τέτοιο στοιχείο ορίζει μια μεταβλητή που χρησιμοποιείται από την διαδικασία.

ΒΑΣΙΚΕΣ ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ: Οι βασικές δραστηριότητες της BPEL είναι οι εξής:

- **<receive>:** Συνεχής αναμονή μιας επιχειρησιακής διαδικασίας για την άφιξη ενός αντίστοιχου μηνύματος. Η δραστηριότητα ολοκληρώνεται με την άφιξη του μηνύματος.
- **<reply>:** Επιτρέπει στην επιχειρησιακή διαδικασία να στείλει ένα μήνυμα ως ανταπόκριση σε ένα μήνυμα που ελήφθη μέσω μιας μίας δραστηριότητας <receive>.
- **<wait>:** Δηλώνει την αναμονή για μια δεδομένη χρονική περίοδο.
- **<invoke>:** Επιτρέπει στην επιχειρησιακή διαδικασία να καλέσει μια request-response λειτουργία μέσω ενός porttype που παρέχεται από κάποιον partner.

- **<assign>**: Ανανεώνει τις τιμές των μεταβλητών των partner links με νέα δεδομένα.
- **<throw>**: Χρησιμοποιείται για να προκαλέσουμε ρητά ένα σφάλμα σε απάντηση σε έναν συγκεκριμένο όρο καταδεικνύοντας έτσι την εμφάνιση λάθους κατά την εκτέλεση διαδικασίας.

ΔΟΜΗΜΕΝΕΣ ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ: Οι δομημένες δραστηριότητες καθοδηγούν ουσιαστικά την διαδικασία και είναι οι εξής:

- **<sequence>**: Καθορίζει μια διαταγμένη ακολουθία δραστηριοτήτων έτσι ώστε να εκτελούνται στη σειρά με την οποία παρατίθενται
- **<switch>**: Επιλέγει έναν κλάδο από το σύνολο των δυνατών επιλογών.
- **<flow>**: Επιτρέπει την παράλληλη εκτέλεση μιας συλλογής ακολουθιών.
- **<while>**: Χρησιμοποιείται για τον καθορισμό βρόχων.
- **<if>-<elseif>-<else>**: Επιτρέπουν να επιλέξουμε έναν κλάδο εκτέλεσης βασισμένο στη λογική.

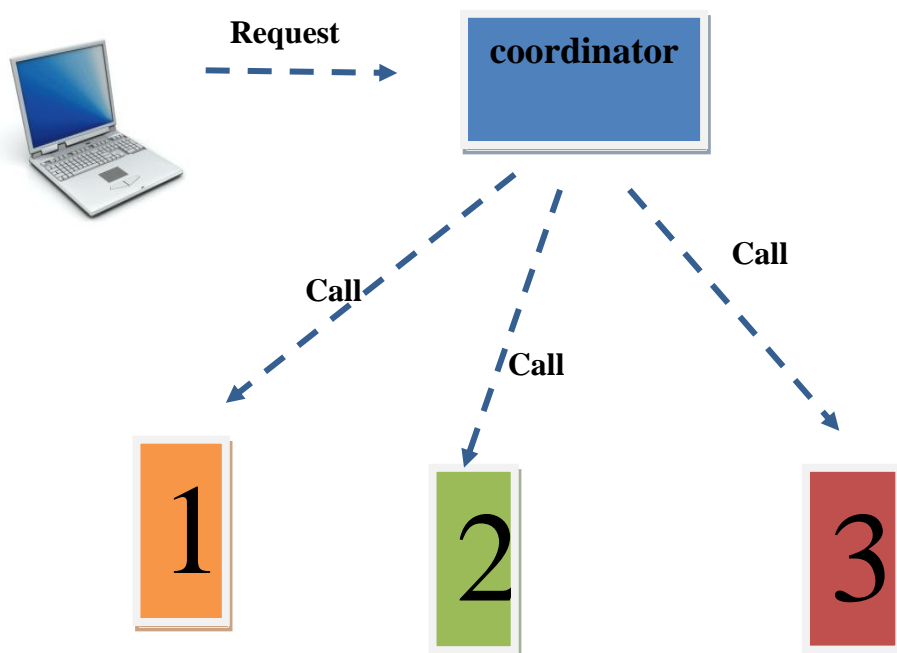
8.3.2 Τεχνικές Ενορχήστρωσης και Χορογραφίας

Ουσιαστικά υπάρχουν δυο βασικοί τρόποι αλληλεπίδρασης των Υπηρεσιών Διαδικτύου, με την βοήθεια των οποίων αυτοματοποιούνται και ολοκληρώνονται οι επιχειρησιακές διαδικασίες

ΕΝΟΡΧΗΣΤΡΩΣΗ

Η συγκεκριμένη τεχνική είναι και η τεχνική στην οποία βασίζεται η BPEL. Εδώ ορίζεται ένα στρώμα ελέγχου, ο ενορχηστρωτής, ο οποίος είναι υπεύθυνος για την πραγματοποίηση επικοινωνίας μεταξύ των συμβαλλόμενων Υπηρεσιών Διαδικτύου. Ουσιαστικά πρόκειται για μια ροή διαδικασίας ανταλλαγής μηνυμάτων μεταξύ των υπηρεσιών, που ελέγχονται από ένα συμβαλλόμενο μέρος-ενορχηστρωτή.

client

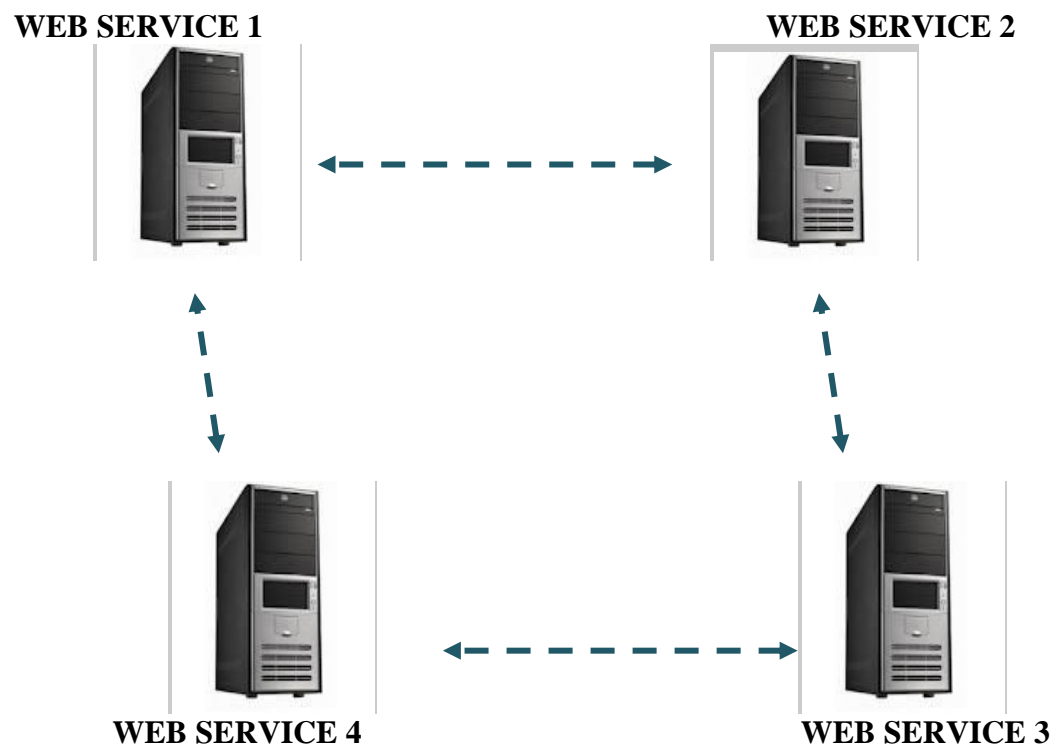


Σχ.8 Ενορχήστρωση Υπηρεσιών Παγκοσμίου Ιστού

Έστω ο πελάτης επικοινωνεί με την Υπηρεσία Web 1 προκειμένου να του επιστρέψει ένα αποτέλεσμα. Η συγκεκριμένη Υπηρεσία θα πρέπει να αλληλεπιδράσει με άλλες τρεις συγκεκριμένες Υπηρεσίες Διαδικτύου προκειμένου να βρει το βέλτιστο αποτέλεσμα για τον πελάτη της. Με τη μέθοδο της ενορχήστρωσης, ο ενορχηστρωτής παίζει τον ρόλο του μεσάζοντα. Εφόσον δεχτεί αίτηση από την web service1 αυτοματοποιεί τη διαδικασία καλώντας τις αντίστοιχες υπηρεσίες και λαμβάνοντας τις απαντήσεις τους. Στο τέλος της αλληλεπίδρασης αποστέλλει το τελικό αποτέλεσμα στην Υπηρεσία Web 1.

ΧΟΡΟΓΡΑΦΙΑ

Στην Χορογραφία δεν υπάρχει κάποια υπηρεσία η οποία συντονίζει την αυτοματοποίηση και ολοκλήρωση της διαδικασίας. Έτσι κανένα συμβαλλόμενο μέρος δεν έχει την κυριότητα και τον έλεγχο της συνομιλίας και γι' αυτόν τον λόγο επιβάλλεται το κάθε μέρος να περιγράφει ξεχωριστά το ρόλο που παίζει στην αλληλεπίδραση. Ουσιαστικά πρόκειται για ένα κατανεμημένο σύστημα το οποίο λειτουργεί με βάση τους δικούς του κανόνες και δεν διαθέτει κεντρικό έλεγχο όπως η τεχνική της ενορχήστρωσης.



Σχ.9 Χορογραφία Υπηρεσιών Παγκοσμίου Ιστού

ΚΕΦΑΛΑΙΟ 9

9 ΒΙΒΛΙΟΓΡΑΦΙΑ

9.1 ΑΝΑΦΟΡΕΣ

- Leonard Richardson & Sam Ruby, RESTful Web Services
- Chatterjee - Webber, Developing Enterprise Web Services, Prentice Hall PTR , N.Jersey 2004
- "Relationship to the World Wide Web and REST Architectures". Web Services Architecture. W3C. <http://www.w3.org/TR/ws-arch/#relwwwrest>. Retrieved 2011-04-22.
- Newcomer E., Lomow G (2005) “ Understanding SOA with Web Services”: Addison – Wesley Professional
- Mathew Macdonald. Microsoft.NET Distributed Applications: Integrating XML Web Services and .NET Remoting, 2001
- Gustavo Alonso, Fabio Casati, Harumi Kuno and Vijay Machiraju. Web Services Concepts, Architectures and Applications, Springer, 2004

9.2 ΔΙΕΥΘΥΝΣΕΙΣ URL

1. <http://www.w3.org/XML>
2. <http://www.w3.org/TR/ws-arch/>
3. <http://www.w3.org/TR/xmlschema-0/>
4. <http://www.w3.org/TR/soap/>
5. <http://www.w3.org/TR/wsdl/>
6. <http://www.w3.org/TR/wsdl20/>
7. <http://www.oasis-open.org/>
8. <http://www.xml.org/>
9. <http://www.uddi.org/>
10. <http://www.w3schools.com/webservices/default.asp>
11. <http://www.w3schools.com/dtd/default.asp>
12. <http://www.w3schools.com/soap/default.asp>
13. <http://www.w3schools.com/wsdl/default.asp>
14. <http://www.w3schools.com/xml/default.asp>
15. <http://www.roseindia.net/webservices/webservices.shtml>
16. <http://www.it.uom.gr/project/soap/Downloads/WebservicesTheory.pdf>
17. <http://www.hitmill.com/webservices/index.html>

18.http://en.wikipedia.org/wiki/Web_application

19.http://en.wikipedia.org/wiki/Web_services

20.<http://en.wikipedia.org/wiki/XML>

21.<http://en.wikipedia.org/wiki/HTTP>

22.<http://www.w3.org/DesignIssues/WebServices.html>

23.www.digitalnews.gr/2037/net-framework

24.<http://www.w3schools.com/webservices/ws-example.asp>

25.<http://www.onjava.com/lpt/a/1025>