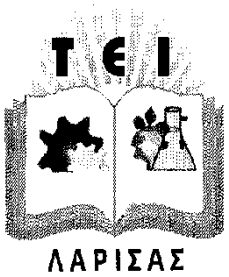


ΑΝΩΤΑΤΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ
ΙΔΡΥΜΑ ΛΑΡΙΣΑΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ

ΤΜΗΜΑ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ
ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

" Η τ ε χ ν ο λ ο γ ί α τ ω ν W e b S e r v i c e s "

ΓΕΡΑΣΙΜΟΣ ΜΗΝΙΑΤΗΣ

ΕΠΙΒΛΕΠΩΝ: ΟΝΟΜΑΤΕΠΩΝΥΜΟ, τίτλος, βαθμίδα

ΛΑΡΙΣΑ 2011

«Δηλώνω υπεύθυνα ότι το παρόν κείμενο αποτελεί προϊόν προσωπικής μελέτης και εργασίας και πως όλες οι πηγές που χρησιμοποιήθηκαν για τη συγγραφή της δηλώνονται σαφώς είτε στις παραπομπές είτε στη βιβλιογραφία. Γνωρίζω πως η λογοκλοπή αποτελεί σοβαρότατο παράπτωμα και είμαι ενήμερος/η για* την *επέλευση των νομίμων συνεπειών»

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή

Τόπος, Ημερομηνία

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1. Ονοματεπώνυμο, Υπογραφή
2. Ονοματεπώνυμο, Υπογραφή
3. Ονοματεπώνυμο, Υπογραφή

Απλός Ορισμός

Μια διαδικτυακή υπηρεσία (Web Service) είναι μια μέθοδος επικοινωνίας μεταξύ δύο ηλεκτρονικών συσκευών σε ένα δίκτυο

Αναλυτικός Ορισμός Προσανατολισμένος στην Ανάπτυξη Εφαρμογών Λογισμικού.

“Web Services ονομάζεται μια ομάδα στενά σχετιζόμενων τεχνολογιών βασισμένη σε ανοικτή και με υποδομή προσανατολισμένη κυρίως Διαδικτυακά. Είναι η σύγχρονη, κινητήρια ψηφιακή δύναμη που κάνει τις διαδικτυακές πύλες των μεγάλων εταιριών να δουλεύουν, παρέχοντας υπηρεσίες και πληροφορίες σε όλους όσους είναι εξουσιοδοτημένοι για κάτι τέτοιο. Υπάλληλοι, πελάτες και προμηθευτές μπορούν μέσω αυτών να λαμβάνουν ευαίσθητες και υψηλής ποιότητας υπηρεσίες όπου κι αν βρίσκονται στον κόσμο. Τα Web Services αποτελούν έναν από τους περισσότερο αναπτυσσόμενους τομείς της Τεχνολογίας Πληροφοριών (IT) και συνιστούν την αιχμή του δόρατος της βιομηχανίας του Ιντερνετ και των Ηλεκτρονικών Υπολογιστών στις μέρες μας.”

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

1. Εισαγωγή

Στις 10 Μάρτη του 2000, ο δείκτης Nasdaq ξεπέρασε τις 5.047[1] μονάδες πραγματοποιώντας ένα ρεκόρ χωρίς προηγούμενο. Η χρυσή εποχή της βιομηχανίας των Πληροφοριών ανθούσε. Εταιρίες σαν την Yahoo και οι μετοχές της ήταν χρυσό όνειρο για κάθε επενδυτή. Τρία χρόνια αργότερα όμως, στις 10 του Μάρτη 2003, ο ίδιος δείκτης έκλεισε στις 1.278 μονάδες. Κάπου μεταξύ αυτών των δύο ημερομηνιών η φούσκα της υψηλής τεχνολογίας είχε σκάσει. Πιθανόν όχι επειδή η ίδια η Τεχνολογία Πληροφοριών (IT) δεν είναι στην πραγματικότητα το μέλλον της ανάπτυξης, αλλά επειδή τέτοιοι παράλογοι ρυθμοί ανάπτυξης των οικονομικών μεγεθών δεν ανταποκρίνονται στην παραγωγή αληθινού έργου. Τότε μια νέα, πιο λιτή εποχή στον τομέα της Τεχνολογίας Πληροφοριών είχε αρχίσει. "Η Νέα Οικονομία" ξεθώριασε μακριά, σαν παράλογη χρηματιστηριακή φούσκα που ήταν, όμως πολλές νέες και πολλά υποσχόμενες τεχνολογίες έδειχναν τότε την πραγματική τους αξία. Οι Τεχνολογίες των Πληροφοριών μέσω του Διαδικτύου και των μεγάλων δικτύων ήταν και είναι απαραίτητες για όλες τις επιχειρήσεις. Αποτελούν έναν σύγχρονο παράγοντα που διαδραματίζει σημαντικό ρόλο σχετικά με την εξέλιξή και την ανάπτυξή τους. Σε όλη την βιομηχανία της τεχνολογίας των Πληροφοριών και την ραγδαία εξάπλωσή της τα επόμενα χρόνια οι νέες τάσεις διακυβεύονταν από την αποδοτικότητα και οι απολαβές σε περικοπές των εξόδων έπαιζαν σημαντικό ρόλο.

Περίπου την ίδια στιγμή, μια νέα τεχνολογία που έγινε γνωστή[2] ως υπηρεσίες Διαδικτύου (Web Services) άρχισε να κινείται στην πρώτη γραμμή του IT περιβάλλοντος. Αντίθετα με πολλές άλλες προσανατολισμένες εφαρμογές της τεχνολογίας προς το Διαδίκτυο, οι οποίες εξαφανίστηκαν σχεδόν όσο γρήγορα εμφανίστηκαν, οι υπηρεσίες Web αψήφησαν τις αρνητικές πιθανότητες των αρχών της δεκαετίας του 2000. Παρέμειναν στο προσκήνιο των εξελίξεων όλη την δεκαετία και έγιναν ένα σημαντικό κεφάλαιο της υψηλής τεχνολογίας του Διαδικτύου. Εξελίχθηκαν ραγδαία και αναπτύχθηκαν τεχνολογικά ενοποιώντας συστήματα και καθιστώντας τα δεδομένα και τις πληροφορίες διαχειρίσιμες από πολλές διαφορετικές εφαρμογές.

Υπηρεσίες διαδικτύου ή Web Services, με λίγα λόγια, είναι εφαρμογές βασισμένες στον Ιστό

1 Nasdaq, ο συγκεντρωτικός χρηματιστηριακός συντελεστής των εταιριών Πληροφορικής για τις Η.Π.Α. Πηγή: Το επίσημο Website http://www.nasdaq.com

2 Ο όρος web service ορίστηκε από την αρμόδια αρχή δηλ το World Wide Web Consortium (W3C). Περισσότερες πληροφορίες στο http://www.w3.org

που δυναμικά αλληλεπιδρούν με άλλες εφαρμογές του Ιστού με χρήση ανοικτών προτύπων. Οι Web Services δεν υπήρξαν επαναστατικές, αλλά βασίζονται σε πρότυπα των οποίων η προσέγγιση λειτουργίας τους τα καθιστά δια λειτουργικά. Το αυξανόμενο bandwidth (χωρητικότητα) των γραμμών του Internet έδωσε αυτό που έλειπε ώστε οι Web Services και η ενοποίηση που προσέφεραν στα μέχρι τότε "κλειστά" συστήματα των εφαρμογών δικτύου να τις αναδείξει ως το επόμενο "μεγάλο θέμα" στο Διαδίκτυο. Λίγο αργότερα η έκρηξη[3] της αυξανόμενης ζήτησης για διείσδυση στο Διαδίκτυο δικαίωσε όλες τις εταιρίες που επένδυσαν στις υπηρεσίες αυτές και τις έκανε να δείχνουν μεγάλα νούμερα ανάπτυξης.

Σε αντίθεση με προηγούμενες εφαρμογές λογισμικού που ήταν υποκειμενικά προσανατολισμένες σε κλειστά πρότυπα, οι υπηρεσίες Web στηρίζονται σε ευρέως αποδεκτά πρότυπα του κλάδου. Χρησιμοποιούν Extensible Markup Language (XML). Αυτή είναι μια γλώσσα ανοιχτού προτύπου για την περιγραφή των δεδομένων και στοιχείων δεδομένων σε μια ιστοσελίδα. Επίσης χρησιμοποιείται και για την περιγραφή μορφοποιημένων εγγράφων. Η Extensible Markup Language και τα συναφή πρότυπα, όπως το Simple Object Access Protocol (SOAP), το Universal Description Discovery and Integration (UDDI), και η Web Services Description Language (WSDL), επιτρέπουν σε εφαρμογές να επικοινωνούν μεταξύ τους μέσω του Internet. Εκτός από αυτά τα απλά πρότυπα, πιο σύνθετες Web Services βασίζονται σε πολυάριθμες άλλες προδιαγραφές για να διευκολυνθεί η συνεργασία μεταξύ εμπορικών εταίρων και εταιρικών πυλών (portals).

3 The Bandwidth Explosion, http://www.dpu.se/ovube_e.html Η επιτυχής διείσδυση του Διαδικτύου σε όλο και περισσότερα νοικοκυριά παγκοσμίως φέρνει την 2η μεγάλη ευκαιρία για τις εταιρίες Πληροφορικής.

2. Web Services με απλά λόγια

Ο πιο απλός τρόπος για να αντιληφθούμε τις Web Services είναι σαν λογισμικό που γνωρίζει με ποιον τρόπο πρέπει να "μιλήσει" σε διαφορετικού τύπου λογισμικά μέσω δικτύου. Γενικά μπορούμε με ασφάλεια να πούμε ότι Web Service είναι σχεδόν κάθε εφαρμογή που έχει την δυνατότητα να προσδιορίσει σε άλλες εφαρμογές τι κάνει, και μπορεί να το κάνει αυτό για εφαρμογές που έχουν εξουσιοδότηση ή άλλες που θεωρούνται "συνεργάτες" ή "εταίροι". Στην πραγματικότητα οι Web Services είναι ένα επίπεδο υποδομών μεταξύ των υφιστάμενων μοντέλων. Συγκεκριμένα μια Web Service πληρεί τα ακόλουθα κριτήρια:

- Είναι σε θέση να εκθέσει και να περιγράψει την παρουσία της σε άλλες εφαρμογές, επιτρέποντας αυτές να κατανοήσουν τι κάνει.
- Μπορεί να γίνει εύκολα αντιληπτή από άλλες εφαρμογές μέσω ενός καταλόγου, αν η υπηρεσία έχει καταχωρηθεί στον κατάλογο αυτό.
- Μπορεί να προβληθεί ή χρησιμοποιηθεί από την εφαρμογή που την χρειάζεται μέσω απλών πρωτοκόλλων.

Οι Web Services επιτρέπουν σε μικρούς και μεγάλους οργανισμούς να χρησιμοποιούν τις υπάρχουσες εφαρμογές τους με νέους καινοτόμους τρόπους. Για παράδειγμα μια αεροπορική εταιρία θα μπορούσε να χρησιμοποιήσει αυτή την τεχνολογία, ενσωματώνοντας στοιχεία από την στεγασμένη σε mainframes βάση δεδομένων της, σε online συστήματα κρατήσεων που στεγάζονται σε Linux ώστε να τα εκμεταλλευτούν συνεργάτες της που ασχολούνται με κρατήσεις ξενοδοχείων και ενοικιάσεις αυτοκινήτων. Για να γίνουν στο παρελθόν τέτοιες συνδέσεις "ξένων" συστημάτων έπρεπε μια κατάλληλη προγραμματιστική ομάδα να αναπτύξει προσαρμοσμένο (custom) λογισμικό. Αυτές οι υλοποιήσεις έχουν αποδειχθεί χρονοβόρες, πολύπλοκες και άρα ακριβές. Με τις υπηρεσίες Web το έργο αυτό ολοκληρώνεται βασικά με την XML, είναι εύκολο και εφαρμόσιμο σε πολλά συστήματα.

Μια απλή web service χαρακτηρίζεται από τρία πρότυπα. Τα SOAP, UDDI και WSDL που στο σύνολό τους υλοποιούν μια βασική λειτουργία τύπου "αίτησης και απόκρισης". Οι απλές υπηρεσίες Web δεν είναι από την φύση τους διαδραστικές δηλαδή δεν απαντούν σε ερωτήσεις. Αντίθετα δημιουργούν μια γενική προσφορά πληροφορίας. Τέτοια παραδείγματα

είναι Web Services που παραδίδουν σε Web Sites πληροφορίες σχετικές με τις πρόσφατες ειδήσεις, τις τιμές των μετοχών και αναφορές καιρικών συνθηκών σε πραγματικό χρόνο.

Από την άλλη μεριά μια σύνθετη υπηρεσία Web ενδέχεται να συνεπάγεται πολλαπλές συνδέσεις και μακρόχρονη ανταλλαγή πληροφοριών με συνεργάτες ή συνεργαζόμενες εφαρμογές με αυτή. Για παράδειγμα μια μεγάλη αλυσίδα λιανικού εμπορίου πρέπει να εφοδιαστεί με στοκ παιγνιδιών για τα Χριστούγεννα. Θα μπορούσε να χρησιμοποιήσει μια ειδική εφαρμογή Web Services που έχει υλοποιηθεί ώστε να δημιουργεί συνδέσεις με μεγάλους προμηθευτές της και αφού συλλέξει πληροφορίες για τα προσφερόμενα προϊόντα από αυτούς να δημιουργεί προτάσεις προς αυτούς σχετικά με τιράζ αγορών. Να ταξινομεί τις απαντήσεις τους με βάση κανόνες προσφοράς / κόστους και να παρουσιάζει αναφορές με τα αποτελέσματα.

Άλλο ένα παράδειγμα μια σύνθετης Web Service θα μπορούσε να αφορά μια μεγάλη αυτοκινητοβιομηχανία η οποία μέσω του Διαδικτύου έχει πρόσβαση σε όλες τις βάσεις δεδομένων των επίσημων αντιπροσώπων της. Στην περίπτωση που τα τμήματα service αυτών, παρουσιάζουν ξαφνικά δραματική πτώση σε συγκεκριμένα κομμάτια εξαρτημάτων, θα ήταν δυνατό να προληφθεί η ζήτηση δημιουργώντας άμεσα αίτημα παραγωγής των συγκεκριμένων ανταλλακτικών. Ακόμα θα ήταν δυνατό μέσω τέτοιων Web Services να υπάρξουν αναφορές τύπου συναγερμού ώστε πολύ σύντομα να εντοπίζονται κατασκευαστικά λάθη στην παραγωγή και να διορθώνονται άμεσα ή να δημιουργούνται διαδικασίες ανακλήσεων.

Παρατηρούμε λοιπόν ότι η ενοποίηση και επικοινωνία των συστημάτων με τέτοιους τρόπους μέσω των Web Services, προσφέρει πολλά πλεονεκτήματα στην σύγχρονη αγορά και οδηγεί τις επιχειρήσεις σε νέους ορίζοντες μέσω ουσιαστικά άπειρων ευκαιριών.

2.1. Ο διχασμός.

Ο τομέας των Web Services είναι χωρισμένος και διχασμένος αν θέλετε σε δύο μεγάλα στρατόπεδα. Από την μια πλευρά η Microsoft προωθεί τις τεχνολογίες .NET που απαιτούν την χρήση λογισμικού Windows. Η Sun Microsystems από την άλλη πλευρά προωθεί μια εκδοχή των Web Services αυστηρά προσανατολισμένη στην γλώσσα προγραμματισμού Java.

Ενώ η Microsoft έχει το πλεονέκτημα της μεγαλύτερης εταιρίας λογισμικού, η προσέγγιση της Sun δείχνει πολύ δελεαστική λόγω της ανοικτής, ευέλικτης, μη περιοριστικής εφαρμογής της. Η εκδοχή Sun's Java 2 Platform, Enterprise Edition (J2EE) που υποστηρίζεται από περισσότερους από τριάντα μεγάλους κατασκευαστές, έχει γίνει ουσιαστικά η εκπλήρωση του οράματος για ένα ενοποιημένο δίκτυο αλληλο-συνεργαζόμενων υπηρεσιών που κάνουν τις εξελιγμένες τεχνολογίες να επικοινωνούν εύκολα μεταξύ τους.

Η δύναμη του J2EE έγκειται στο γεγονός ότι δεν δεσμεύει τους χρήστες στο όραμα μιας ενιαίας εταιρίας. Βασίζεται σε τεχνολογίες που υπάρχουν εκεί για κάποιο καιρό και ταιριάζει καλά και με το κίνημα του ελεύθερου λογισμικού ανοικτού κώδικα. Το .NET είναι ουσιαστικά μια επανατοποθέτηση αρχιτεκτονικής της πλατφόρμας της Microsoft προσανατολισμένης στις σύγχρονες τάσεις ενοποίησης και συνεργασίας των συστημάτων μέσω δικτύων. Για αυτό το λόγω ταιριάζει καλύτερα σε οργανισμούς και εταιρίες που είναι ήδη βαθιά δεσμευμένοι με τεχνολογίες της Microsoft.

Το ιστορικό της Microsoft στην υιοθέτηση ανοικτών προτύπων επίσης συμβάλει στην δημιουργία ενός δισταγμού για την εμπιστοσύνη στο .NET. Ο γίγαντας του λογισμικού πολλές φορές στο παρελθόν έχει χρησιμοποιήσει ανοικτά πρότυπα απλά για να ενσωματώσει λειτουργίες στα δικά του κλειστά συστήματα διακομιστών (servers) και πελατών (clients). Στα χέρια της Microsoft τα ανοικτά πρότυπα XML και SOAP θα απέδιδαν κάτι καλύτερο από την υλοποίηση με Java; Αυτή την στιγμή κανείς δεν ξέρει πραγματικά.

2.2. Η δύναμη των Web Services

Οι Web Services χαρακτηρίζονται από την ανεξάρτητη φύση τους. Η δύναμή τους βρίσκεται ακριβώς σε αυτή την ανεξαρτησία. Δεν έχει σημασία αν οι συνδεόμενοι οργανισμοί και επιχειρήσεις είναι οργανωμένες με εσωτερικής κατασκευής λογισμικό που τρέχει σε mainframes και θέλουν να συνδεθούν με έναν συνεργάτη που τρέχει λογισμικό του εμπορίου σε έναν απλό Windows 2003 Server. Οι Υπηρεσίες Web είναι ουδέτερες και οι επιχειρήσεις που τις υιοθετούν δεν χρειάζεται να ανησυχούν για θέματα όπως η δυαδική συμβατότητα μεταξύ λειτουργικών συστημάτων.

Αυτή η ουδετερότητα είναι κρίσιμη όταν πρόκειται να υλοποιήσει κανείς συνεργασίες μεταξύ εταίρων σε επίπεδο λογισμικού. Η ενσωμάτωση επιχειρησιακών διαδικασιών με τα παραδοσιακά μέσα λογισμικού ήταν δύσκολη και ακριβή επειδή είναι απίθανο οι εταιρίες να χρησιμοποιούν συμβατά λογισμικά υλοποιημένα με παρόμοιες τεχνολογίες. Με λίγα λόγια αν το λογισμικό των εταιριών που θέλουν να συνεργαστούν ήταν "γραμμένο" σε διαφορετική γλώσσα τότε η επικοινωνία των συστημάτων απαιτούσε πολύ κόπο, χρόνο και χρήμα. Όμως η XML μπορεί να ερμηνευτεί από οποιαδήποτε γλώσσα προγραμματισμού. Δεδομένα και πληροφορίες με αυτό τον τρόπο μπορούν να σταλούν από οποιοδήποτε ενδιάμεσο λογισμικό οπότε δεν έχει σημασία αν χρησιμοποιήσατε Java, C++, λειτουργικό Microsoft ή Linux.

Η λειτουργική απλότητα είναι άλλος ένας λόγος για τον οποίο οι επιχειρήσεις ενδιαφέρονται τόσο πολύ για τις Web Services. Με τις ετικέτες καθορισμού αξίας (value-defining tags) η XML επιτρέπει σε μια σε εφαρμογή Web Service να κάνει μια κλήση απομακρυσμένης λειτουργίας (remote procedure call – RPC) σε ένα χώρο αποθήκευσης δεδομένων. Μια τέτοια διαδικασία καθιστά ακόμη και την ανταλλαγή αναλυτικών πληροφοριών απλή και σχεδόν αλάνθαστη.

Τα πρότυπα και η απλότητα οδηγούν σε ταχύτερη ανάπτυξη εφαρμογών με πολύ πιο χαμηλά κόστη. Με τα πρότυπα οι διευθυντές των έργων ανάπτυξης γνωρίζουν με τι έχουν να κάνουν. Δεν υπάρχει μυστήριο, δηλαδή δεν υπάρχουν δυσάρεστες εκπλήξεις ασυμβατοτήτων. Επίσης με τα πρότυπα οι πωλητές δεν μπορούν να ισχυριστούν υψηλά κόστη και να ζητούν υψηλές τελικές τιμές από τον πελάτη που θέλει να αναπτύξει την εφαρμογή, αφού δεν πρόκειται για ιδιόκτητες τεχνολογίες. Επιπλέον οι υπηρεσίες Web μπορούν να βοηθήσουν στην βελτίωση της αποτελεσματικότητας επιτρέποντας στις επιχειρήσεις να ρυθμίζουν τις διαδικασίες μεταξύ τους και των εμπορικών εταίρων τους.

Η υπόσχεση των τεχνολογιών Web Services για εύκολη και σταθερή εργασία, μαζί με την σχετικά γρήγορη καμπύλη εκμάθησης, έχει εμπνεύσει ήδη πολλούς προγραμματιστές και ομάδες ανάπτυξης εφαρμογών, να ξεκινήσουν την απόκτηση δεξιοτήτων ώστε να μπορούν να προσφέρουν τέτοιες υπηρεσίες στους πελάτες στους. Δεδομένου ότι η εκμάθηση ανάπτυξης Web Services δεν είναι τόσο δύσκολη όπως για παράδειγμα η εκμάθηση της C++ ή μια μετάβαση από την C++ στην Java, είναι εύκολο να βρεθούν αρκετά ταλέντα που θα

δραστηριοποιηθούν σε αυτό τον τομέα. Παλαιότερες μεταβάσεις σε νέες τεχνολογίες για την εποχή τους είχαν αποδειχθεί μεγάλος πονοκέφαλος για τις εταιρίες ανάπτυξης ή μεμονωμένους ανεξάρτητους προγραμματιστές.

Οι Web Services έχουν το δυναμικό να προσφέρουν σημαντική εξοικονόμηση κόστους λόγω της προτυποποίησης, της ανοικτής αρχιτεκτονικής και της ευκολίας της επικοινωνίας συμβατών και ασύμβατων συστημάτων μεταξύ τους. Παλαιότερα μεγάλοι οργανισμοί έχουν δαπανήσει ποσά ύψους μερικών εκατομμυρίων δολαρίων απλά για να συνδέσουν τα περιβάλλοντα των εφαρμογών τους. Βέβαια ίσως πολλοί οργανισμοί, μεγάλες ή μικρότερες εταιρίες να χρειαστεί για αρχή να μεταβάλουν, μετατρέψουν τα συστήματα λογισμικών και τα συστήματα των βάσεων δεδομένων που χρησιμοποιούν αν είναι απαρχαιωμένα. Όμως αυτά έτσι κι αλλιώς θα χρειαστούν αργά ή γρήγορα εκσυγχρονισμό. Το κόστος μετατροπής υφιστάμενων υπηρεσιών σε τεχνολογίες με Web Services θα αποδειχθεί μακροπρόθεσμα πολύ μικρό αφού θα αποκτήσουν υποδομή που πολύ εύκολα θα επιτρέπει την συνεργασία με χιλιάδες άλλες Web Services πελατών ή συνεργατών τους. Για παράδειγμα μεταφέροντας την παλαιά βάση δεδομένων σε καινούρια τεχνολογία συμβατή με μηνύματα SOAP δύναται μια επηχείρηση να προσφέρει πανεύκολα τα δεδομένα που επιθυμεί μέσω e-shops σε χιλιάδες νέους πελάτες.

2.3. Η τεχνολογία των Web Services

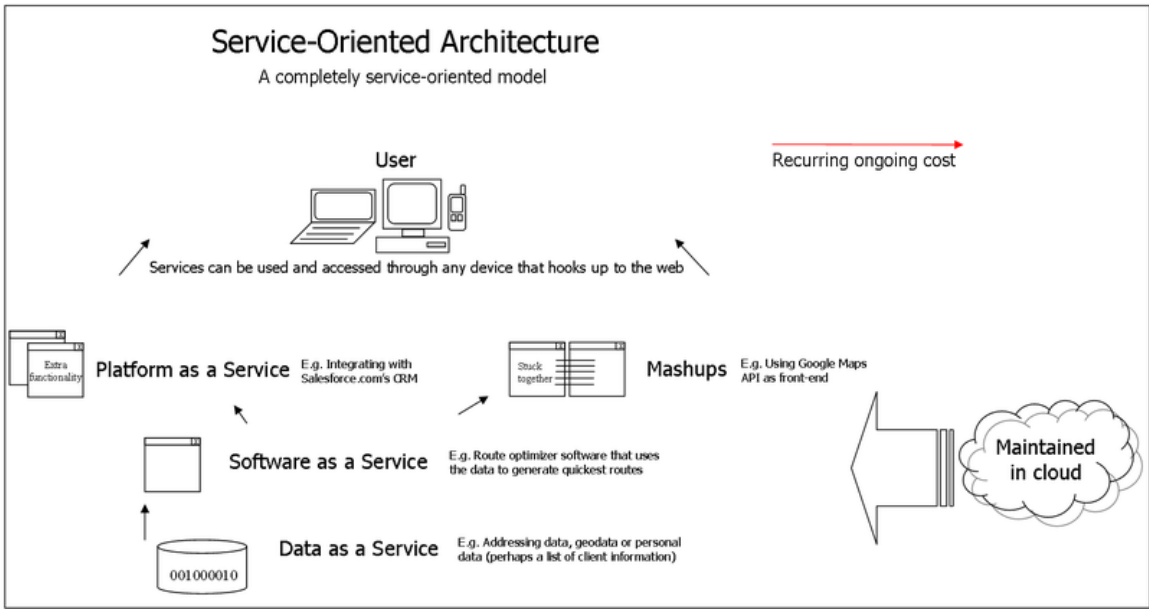
Το W3C (World Wide Web Consortium)[4] ορίζει μια Web Service σαν ένα σύστημα λογισμικού σχεδιασμένο να υποστηρίζει δια λειτουργική αλληλεπίδραση μηχανής-προς-μηχανή μέσω ενός δικτύου. Διαθέτει μια διεπαφή που περιγράφεται σε επεξεργάσιμη μηχανογραφικά μορφή (ειδικά Web Services Description Language WSDL). Τα συστήματα αλληλεπιδρούν με τη Διαδικτυακή Υπηρεσία με τρόπο που προβλέπεται από την περιγραφή της, χρησιμοποιώντας μηνύματα SOAP, που μεταφέρονται χρησιμοποιώντας HTTP με XML σύνταξη ή προτυποποίηση, σε συνδυασμό με διαδικτυακά σχετιζόμενα πρότυπα.

Το W3C αναφέρει επίσης: "Μπορούμε να προσδιορίσουμε δύο σημαντικές κατηγορίες των υπηρεσιών Web. Υπηρεσίες Web συμβατές με REST, στις οποίες ο πρωταρχικός σκοπός της

4 Είναι ο κύριος διεθνής οργανισμός τυποποίησης για το World Wide Web, WWW ή συντομογραφία W3

υπηρεσίας είναι να χειριστεί XML αναπαραστάσεις πόρων που προέρχονται από το Web χρησιμοποιώντας ένα ενιαίο σύνολο λειτουργιών και αυθαίρετες Υπηρεσίες Web, όπου η υπηρεσία μπορεί να εκθέσει ένα αυθαίρετο σύνολο πράξεων."

Οι "μεγάλες" υπηρεσίες Web λοιπόν χρησιμοποιούν μηνύματα Extensible Markup Language (XML) που είναι συμβατά με το πρότυπο SOAP και έχουν γίνει δημοφιλείς στις παραδοσιακές επιχειρήσεις. Σε τέτοια συστήματα συνήθως υπάρχει μια αναγνώσιμη από μηχάνημα περιγραφή των εργασιών που προσφέρονται από την υπηρεσία Web, που έχει γραφτεί με Web Services Description Language (WSDL). Το τελευταίο δεν είναι μια απαίτηση ενός τελικού σημείου SOAP, αλλά είναι απαραίτητη προϋπόθεση για την αυτοματοποιημένη παραγωγή κώδικα υπολογιστή-πελάτη σε πολλά Java και .NET πλαίσια SOAP.



Εικόνα 1: Επεξήγηση της Service Oriented Architecture από wikipedia.org

Το Web API είναι μια εξέλιξη στις υπηρεσίες Web (σε ένα κίνημα που ονομάζεται Web 2.0), όπου έμφαση έχει δοθεί στην απομάκρυνση από τις Web Services που βασίζονται σε υλοποίηση SOAP έναντι εκείνων που βασίζονται σε επικοινωνίες Representational State Transfer (REST). Οι Web Services τύπου REST δεν απαιτούν XML, SOAP, ή WSDL ορισμούς API. Τα Web API επιτρέπουν το συνδυασμό πολλαπλών υπηρεσιών Web σε νέες εφαρμογές γνωστές ως mashups.

Όταν χρησιμοποιείται στο πλαίσιο της ανάπτυξης ιστοσελίδων, το Web API είναι ένα καθορισμένο σύνολο μηνυμάτων αίτησης του Hypertext Transfer Protocol (HTTP) μαζί με ορισμό των μηνυμάτων απόκρισης, τα οποία εκφράζονται συνήθως σε μορφή Extensible Markup Language (XML) ή JavaScript Object Notation (JSON).

Κατά την εκτέλεση σύνθετων υπηρεσιών Web, κάθε υπό-υπηρεσία μπορεί να θεωρείται ανεξάρτητη. Ο χρήστης δεν έχει κανένα έλεγχο επί αυτών των υπηρεσιών. Οι ίδιες οι υπηρεσίες Web δεν παρέχουν εγγύηση αξιοπιστίας. Η παρέχουσα την Web Service εταιρία, μπορεί να αφαιρέσει, να αλλάξει ή να ενημερώσει τις υπηρεσίες χωρίς προειδοποίηση προς τους χρήστες. Η ανοχή και αξιοπιστία στις βλάβες ή στην κακή λειτουργία δεν υποστηρίζεται με συγκεκριμένο τρόπο. Σφάλματα μπορεί να συμβούν κατά τη διάρκεια της εκτέλεσης. Ο χειρισμός εξαιρέσεων στο πλαίσιο των υπηρεσιών Web είναι ακόμη ένα ανοικτό θέμα έρευνας και δεν έχουν συμφωνηθεί τακτικές ή πολιτικές αντιμετώπησης. Ωστόσο, μια τέτοια αποτυχία ή σφάλμα της υπηρεσίας μπορεί να αντιμετωπιστεί με την αποστολή ενός αντικειμένου σφάλματος στον πελάτη σαν ειδοποίηση πως η υπηρεσία δεν λειτούργησε με αποδεκτό τρόπο.

2.4. Μορφές χρήσης των Web Services

Οι υπηρεσίες Web είναι ένα σύνολο εργαλείων που μπορούν να χρησιμοποιηθούν με μια σειρά από τρόπους. Οι τρεις πιο συχνές μορφές χρήσης είναι οι RPC, SOA και REST.

Κλήσεις απομακρυσμένης διαδικασίας ή Remote procedure calls (RPC)

Οι RPC Web Services παρουσιάζουν μια κατανεμημένη λειτουργία (ή μέθοδο) κλήσης διεπαφής που είναι οικεία σε πολλούς προγραμματιστές. Συνήθως, η βασική μονάδα RPC της υπηρεσίας Web είναι η WSDL λειτουργία.

Εικόνα 2: Απλοποιημένο σχήμα request and provide που αφορά την λειτουργία XML-RPC της Web Service με ενδοιάμεσο SOAP. Το σχήμα προέρχεται από wikipedia.org

Τα πρώτα εργαλεία για προγραμματιστές Web Services, επικεντρώθηκαν στο στυλ εφαρμογής/υλοποίησης RPC, και ως αποτέλεσμα αυτή η εφαρμογή είναι ευρέως διαδεδομένη και υποστηρίζεται. Ωστόσο, ορισμένες φορές έχει επικριθεί επειδή δεν είναι χαλαρά συνδεδεμένες οι κλήσεις, αφού τις περισσότερες φορές υλοποιήθηκαν με συγκεκριμένες λειτουργίες και μεθόδους εξειδικευμένες της γλώσσας προγραμματισμού που χρησιμοποιήθηκε. Πολλοί προμηθευτές αισθάνθηκαν ότι αυτή η προσέγγιση στις Web Services είναι αδιέξοδο, και πίεσαν για υλοποιήσεις RPC που θα άρουν τον περιορισμό αυτό προσανατολισμένες προς το WS-I Basic Profile.

Άλλες υλοποιήσεις με σχεδόν ίδια λειτουργικότητα σαν των RPC είναι του Object Management Group's (OMG) που λέγεται Common Object Request Broker Architecture (CORBA), της Microsoft το Distributed Component Object Model (DCOM) και της Sun MicroSystems η Java Remote Method Invocation (RMI).

Αρχιτεκτονική Προσανατολισμένη στην Υπηρεσία ή Service-Oriented Architecture (SOA)

Οι Υπηρεσίες Web μπορούν να χρησιμοποιηθούν επίσης για την εφαρμογή μιας αρχιτεκτονικής σύμφωνης με την έννοια SOA, όπου η βασική μονάδα επικοινωνίας είναι ένα μήνυμα κι όχι μια λειτουργία όπως πριν. Αυτό συνήθως αναφέρεται ως υπηρεσία message-oriented.

Οι SOA Web Services υποστηρίζονται από πολλούς μεγάλους κατασκευαστές λογισμικού και αναλυτές του κλάδου. Σε αντίθεση με τις RPC υπηρεσίες Web, εδώ η χαλαρή σύζευξη είναι πιο πιθανή, διότι η εστίαση είναι στο «συμβόλαιο» που το WSDL παρέχει, παρά τις υποκείμενες λεπτομέρειες της εφαρμογής υλοποίησης ή της γλώσσας προγραμματισμού.

Κατάσταση Παραστατικής Μεταφοράς ή Representational state transfer (REST)

Η χρήση/υλοποίηση των Web Services με REST, επιχειρεί να περιγράψει αρχιτεκτονικές που χρησιμοποιούν HTTP ή παρόμοια πρωτόκολλα, περιορίζοντας την διεπαφή σε ένα σύνολο γνωστών, και τυποποιημένων διεργασιών (όπως GET, POST, PUT, DELETE της HTTP). Εδώ, η εστίαση επιτυγχάνεται αλληλεπιδρώντας με πόρους ή σημεία κατάστασης, παρά με μηνύματα ή ενέργειες.

Μια αρχιτεκτονική που βασίζεται σε REST (που είναι «ήρεμη», RESTful) μπορεί να χρησιμοποιήσει WSDL για να περιγράψει μηνύματα SOAP μέσω HTTP, μπορεί να εφαρμοστεί ως μια αφαίρεση καθαρά πάνω από το SOAP (π.χ., WS-Transfer), ή μπορεί να δημιουργηθεί χωρίς καθόλου χρήση SOAP.

Με την WSDL ver 2.0 παρέχεται υποστήριξη για σύνδεση με όλες τις μεθόδους αίτησης HTTP (όχι μόνο GET και POST όπως και στην έκδοση 1.1), έτσι ώστε πλέον επιτρέπεται καλύτερη εφαρμογή των RESTfull υπηρεσιών Web. Ωστόσο, η υποστήριξη για την προδιαγραφή αυτή εξακολουθεί να είναι φτωχή ή ελλιπής στα εργαλεία ανάπτυξης λογισμικού τα οποία ακόμη προσφέρουν υποστήριξη μόνο για WSDL ver 1.1.

2.5. Αυτοματοποιημένες μεθοδολογίες σχεδιασμού

Αυτοματοποιημένα εργαλεία που έχουν κατασκευαστεί μπορούν να βοηθήσουν στη δημιουργία μιας υπηρεσίας Web. Για υπηρεσίες που χρησιμοποιούν WSDL είναι δυνατό είτε να δημιουργηθούν αυτόματα WSDL για τις υφιστάμενες κλάσεις (classes), ονομάζεται "από κάτω προς τα επάνω στρατηγική" (bottom-up) , ή να δημιουργηθεί ένας σκελετός κλάσης, δεδομένου του υπάρχοντος WSDL και αυτή η προσέγγιση ονομάζεται "στρατηγική από πάνω προς τα κάτω"(top-down).

Ένας προγραμματιστής χρησιμοποιώντας bottom-up στρατηγική γράφει εφαρμοσμένες κλάσεις πρώτα (σε κάποια γλώσσα προγραμματισμού), και στη συνέχεια χρησιμοποιεί ένα εργαλείο δημιουργίας WSDL για να εκθέσει τις μεθόδους από αυτές τις κλάσεις, σε υπηρεσία Web. Αυτή είναι συχνά η πιο απλή προσέγγιση και προτιμάται.

Όταν ένας προγραμματιστής χρησιμοποιήσει την μέθοδο top-down τότε γράφει μόνος του πρώτα το έγγραφο WSDL και στη συνέχεια; χρησιμοποιεί ένα εργαλείο αυτόματης παραγωγής κωδικα για να παράγει τον σκελετό της κλάσσης όπως απαιτείται. Αυτή η μέθοδος γενικά θεωρείται πιο δύσκολη αλλά παράγει πιο καθαρά σχέδια.

2.6. Επικρίσεις

Οι επικριτές των non-RESFfull Web Services παραπονούνται συχνά ότι είναι υπερβολικά πολύπλοκες, και εξαρτώνται από εργαλεία ανάπτυξης μεγάλων προμηθευτών λογισμικού, ενώ η ανάπτυξή τους θα έπρεπε να γίνεται με εφαρμογές ανοιχτού κώδικα. Υπάρχουν εφαρμογές ανοιχτού κώδικα, όπως οι Apache Axis και Apache CXF για web services development.

Ένα βασικό μέλημα για την ανάπτυξη REST Web Services είναι ότι το περιβάλλον εργαλείων SOAP WS καθιστά εύκολο να καθοριστούν νέες διεπαφές για απομακρυσμένη αλληλεπίδραση. Δεδομένου ότι μια μικρή αλλαγή στο διακομιστή (ακόμη και μια αναβάθμιση της SOAP στοίβας) μπορεί να οδηγήσει σε διαφορετικές WSDL και συνεπώς διαφορετική διεπαφή υπηρεσίας, οι προγραμματιστές συχνά στηρίζονται σε διαδικασίες ενδοσκόπησης για σωστή εξαγωγή του WSDL. Οι κλάσεις της πλευράς του πελάτη (client side) που μπορεί να παραχθούν από τις WSDL και XSD περιγραφές των Web Services είναι συχνά συνδεδεμένες με μια συγκεκριμένη έκδοση του τελικού σημείου SOAP. Αυτό μπορεί να δημιουργήσει ασυμβατότητες και να "σπάσει", αν το τελικό σημείο (end-point) αλλάξει ή το κομμάτι της SOAP στοίβας στην πλευρά πελάτη (client side) αναβαθμιστεί. Καλά σχεδιασμένα SOAP τελικά σημεία (με χειρόγραφες XSD και WSDL) δεν υποφέρουν από αυτό το πρόβλημα, αλλά εξακολουθεί να υπάρχει το πρόβλημα ότι μια προσαρμοσμένη διεπαφή για κάθε υπηρεσία απαιτεί μια προσαρμοσμένη client side για κάθε υπηρεσία.

3. Βασικές Τεχνολογίες Υλοποίησης των Web Services

Θα εξετάσουμε αμέσως τώρα τις βασικές τεχνολογίες στις οποίες βασίζονται οι Web Services για την υλοποίησή τους. Εξηγώντας αυτό το περισσότερο τεχνικό κομμάτι θα δώσουμε και κάποια παραδείγματα ενδεικτικά τις λειτουργίας της κάθε τεχνολογίας.

3.1. Extensible Markup Language (XML)

Η XML (αγγλικό αρκτικόλεξο από το Extensible Markup Language) είναι μία γλώσσα σήμανσης, που περιέχει ένα σύνολο κανόνων για την ηλεκτρονική κωδικοποίηση κειμένων. Ορίζεται, κυρίως, στην προδιαγραφή XML 1.0[5], που δημιούργησε ο διεθνής οργανισμός προτύπων W3C (World Wide Web Consortium), αλλά και σε διάφορες άλλες σχετικές προδιαγραφές ανοιχτών προτύπων.

Η XML σχεδιάστηκε δίνοντας έμφαση στην απλότητα, τη γενικότητα και τη χρησιμότητα στο Διαδίκτυο. Είναι μία μορφοποίηση δεδομένων κειμένου, με ισχυρή υποστήριξη Unicode για όλες τις γλώσσες του κόσμου. Αν και η σχεδίαση της XML εστιάζει στα κείμενα, χρησιμοποιείται ευρέως για την αναπαράσταση αυθαίρετων δομών δεδομένων, που προκύπτουν για παράδειγμα στις υπηρεσίες ιστού (Web Services).

Υπάρχει μία ποικιλία διεπαφών προγραμματισμού εφαρμογών, που μπορούν να χρησιμοποιούν οι προγραμματιστές, για να προσπελάσουν δεδομένα XML, αλλά και διάφορα συστήματα σχημάτων XML, τα οποία είναι σχεδιασμένα για να βοηθούν στον ορισμό γλωσσών, που προκύπτουν από την XML.

Έως το 2009, είχαν αναπτυχθεί εκατοντάδες γλώσσες που βασίζονται στην XML, συμπεριλαμβανομένων του RSS, του SOAP και της XHTML. Προεπιλεγμένες κωδικοποιήσεις βασισμένες στην XML, υπάρχουν για τις περισσότερες σουίτες εφαρμογών γραφείου, συμπεριλαμβανομένων του Microsoft Office (Office Open XML), του OpenOffice.org (OpenDocument) και του iWork της Apple.

Αμέσως παραθέτουμε ένα μικρό παράδειγμα της σύνταξης κώδικα σε XML:

5 Οι προδιαγραφές της XML, στην πέμπτη έκδοσή τους πλέον από http://www.w3.org/TR/xml/

```
<?xml version="1.0"?>
    <quiz>
        <question>
        Who was the forty-second
        president of the U.S.A.?
        </question>
        <answer>
        William Jefferson Clinton
        </answer>
        <!-- Note: We need to add
        more questions later.-->
    </quiz>
```

3.1.1 Βασική ορολογία

Εδώ παραθέτουμε τα βασικά στοιχεία που συναντάμε στην απλή καθημερινή χρήση της γλώσσας XML. Δεν είναι η πλήρης λίστα όλων των όρων που υπάρχουν στη γλώσσα XML, αλλά ενδεικτική. Βασίζεται στην προδιαγραφή XML 1.0.

Χαρακτήρας Unicode

Εξ ορισμού, ένα κείμενο XML είναι μία ακολουθία χαρακτήρων. Σχεδόν κάθε χαρακτήρας Unicode μπορεί να εμφανίζεται σε ένα κείμενο XML.

Επεξεργαστής και Εφαρμογή

Είναι το λογισμικό που επεξεργάζεται ένα κείμενο XML. Είναι αναμενόμενο, ότι ένας επεξεργαστής δουλεύει για μία εφαρμογή. Υπάρχουν μερικές πολύ συγκεκριμένες απαιτήσεις, σχετικά με το τι μπορεί και τι δεν μπορεί να κάνει ένας επεξεργαστής XML, αλλά καμία, όσον αφορά στη συμπεριφορά της εφαρμογής. Ο επεξεργαστής (όπως ονοματίζεται από την προδιαγραφή), αναφέρεται συχνά, με τον αγγλικό όρο XML parser.

Σήμανση και Περιεχόμενο

Οι χαρακτήρες που απαρτίζουν ένα κείμενο XML, αποτελούν είτε τη σήμανση είτε το

περιεχόμενό του. Η σήμανση και το περιεχόμενο, μπορούν να επισημανθούν και να διακριθούν, ύστερα από την εφαρμογή κάποιων απλών συντακτικών κανόνων. Όλα τα αλφαριθμητικά που συνιστούν τη σήμανση, είτε ξεκινούν με το χαρακτήρα "<" και καταλήγουν στο χαρακτήρα ">", είτε ξεκινούν με το χαρακτήρα "&" και καταλήγουν στο χαρακτήρα ";". Ακολουθίες χαρακτήρων που δε συνιστούν τη σήμανση, αποτελούν το περιεχόμενο ενός κειμένου XML.

Ετικέτα

Ένα στοιχείο σήμανσης που ξεκινά με το χαρακτήρα "<" και καταλήγει στο χαρακτήρα ">" είναι ετικέτα. Υπάρχουν τρία είδη ετικέτας: ετικέτες-αρχής, για παράδειγμα <section>, ετικέτες-τέλους, για παράδειγμα </section>, και ετικέτες χωρίς περιεχόμενο, για παράδειγμα <line-break/>.

Στοιχείο

Ένα λογικό απόσπασμα ενός κειμένου ε'ιναι το στοιχείο, που είτε ξεκινά με μία ετικέτα-αρχής και καταλήγει σε μία ετικέτα-τέλους, είτε αποτελείται μόνο από μία ετικέτα-χωρίς-περιεχόμενο. Οι χαρακτήρες που υπάρχουν, αν υπάρχουν, μεταξύ μιας ετικέτας-αρχής και μιας ετικέτας-τέλους, συνιστούν το περιεχόμενο του στοιχείου, το οποίο μπορεί να περιέχει σήμανση, συμπεριλαμβανομένων και άλλων στοιχείων, που ονομάζονται στοιχεία-παιδιά. Ένα παράδειγμα ενός στοιχείου είναι το <Greeting>Hello, world.</Greeting>. Ένα άλλο είναι το <line-break/>.

Χαρακτηριστικό

Ένα στοιχείο σήμανσης είναι το χαρακτηριστικό, που αποτελείται από ένα ζευγάρι όνομα/τιμή, το οποίο υπάρχει μέσα σε μία ετικέτα-αρχής ή σε μία ετικέτα-χωρίς-περιεχόμενο. Στο παράδειγμα παρακάτω, το στοιχείο img έχει δύο χαρακτηριστικά, τα src και alt: . Ένα άλλο παράδειγμα θα ήταν το <step number="3">Connect A to B.</step>, όπου το όνομα του χαρακτηριστικού είναι "number" και η τιμή του είναι "3".

Δήλωση XML

Τα κείμενα XML μπορούν να αρχίζουν, με τη δήλωση κάποιων πληροφοριών σχετικών με αυτά, όπως στο ακόλουθο παράδειγμα που ορίζεται η έκδοση της XML και η κωδικοποίηση που χρησιμοποιείται:

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

Το παρακάτω είναι ένα μικρό, αλλά πλήρες κείμενο XML, που κάνει χρήση όλων των παραπάνω εννοιών και στοιχείων.

```
<?xml version="1.0" encoding='UTF-8'?>
<painting>
    <img src="madonna.jpg" alt='Foligno Madonna, by Raphael'/>
    <caption>This is Raphael's "Foligno" Madonna, painted in
    <date>1511</date>-<date>1512</date>.</caption>
</painting>
```

Υπάρχουν πέντε στοιχεία σε αυτό το κείμενο του παραδείγματος: τα painting, img, caption, και δύο date. Τα στοιχεία date, είναι παιδιά του στοιχείου caption, το οποίο είναι παιδί του στοιχείου-ρίζας painting. Το στοιχείο img έχει δύο χαρακτηριστικά, τα src και alt.

3.1.2 Χαρακτήρες και διαφυγή

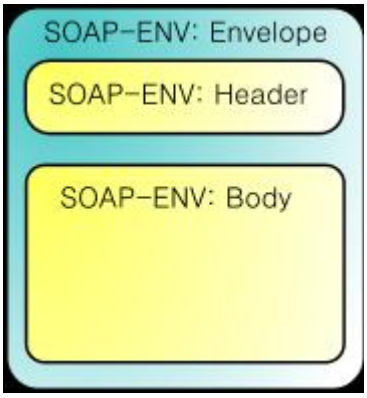
Τα κείμενα XML αποτελούνται εξ ολοκλήρου από χαρακτήρες Unicode. Εκτός από ένα μικρό αριθμό, ειδικά εξαιρούμενων χαρακτήρων ελέγχου, κάθε χαρακτήρας που ορίζεται στο Unicode, μπορεί να εμφανίζεται στο περιεχόμενο ενός κειμένου XML. Το σύνολο των χαρακτήρων που μπορούν να εμφανίζονται στη σήμανση, αν και κάπως περιορισμένο, παραμένει μεγάλο.

Η XML παρέχει κάποιες διευκολύνσεις για την ταυτοποίηση της κωδικοποίησης των χαρακτήρων Unicode που απαρτίζουν ένα κείμενο και για την απεικόνιση χαρακτήρων που, για τον έναν ή τον άλλο λόγο, δεν μπορούν να χρησιμοποιηθούν ευθέως.

3.2. Simple Object Access Protocol (SOAP)

Ως SOAP, έχει οριστεί μια προδιαγραφή πρωτοκόλλου που ορίζει με σαφήνεια την ανταλλαγή δομημένων πληροφοριών για την εφαρμογή των Web Services σε δίκτυα

υπολογιστών. Σε Extensible Markup Language (XML) περιγράφεται το περιεχόμενο των μηνυμάτων που μεταφέρει, και συνήθως βασίζεται σε άλλα Application Layer πρωτόκολλα, όπως οι Κλήσεις Απομακρυσμένης Διαδικασίας (Remote Procedure Call - RPC) και το Πρωτόκολλο Μεταφοράς Υπερκειμένου (Hypertext Transfer Protocol - HTTP), για τη διαπραγμάτευση και τη μετάδοση αυτών. Το SOAP μπορεί να σχηματίσει το θεμέλιο στρώμα της στοίβας πρωτοκόλλων μιας Web Service, που παρέχει ένα βασικό πλαίσιο ανταλλαγής μηνυμάτων. Πάνω σε αυτό η Υπηρεσία μπορεί να οικοδομηθεί. Αυτό το βασιζόμενο σε XML πρωτόκολλο αποτελείται από τρία μέρη: ένα φάκελο, ο οποίος προσδιορίζει τι είναι μέσα στο μήνυμα και πώς πρέπει να το διαχειριστεί κάποιος, ένα σύνολο κανόνων κωδικοποίησης για να εκφραστούν περιπτώσεις καθορισμένων τύπων δεδομένων, καθώς και μια σύμβαση για την σωστή αναπαράσταση κλήσεων διαδικασίας και απαντήσεων σε αυτές.



Εικόνα 3: Αυτός είναι ένας φάκελος μηνύματος SOAP που περιέχει 2 φακέλους. Τον φάκελο επικεφαλίδας και τον φάκελο σώματος που περιέχει τα δεδομένα.

Ως παράδειγμα του πώς οι SOAP διαδικασίες μπορούν να χρησιμοποιηθούν θα αναφέρουμε το εξής: Ένα μήνυμα SOAP θα μπορούσε να σταλεί σε μια Ιστοσελίδα που είναι ενεργοποιημένη για να χρησιμοποιήσει Υπηρεσίες Web όπως μια βάση δεδομένων των τιμών ακινήτων, με τις παραμέτρους που χρειάζονται για μια αναζήτηση. Η Ιστοσελίδα θα επιστρέψει τότε με μορφή εγγράφου XML τα δεδομένα που προκύπτουν, για παράδειγμα τιμές, τοποθεσία, και χαρακτηριστικά ακινήτων. Με τα δεδομένα να επιστρέφονται σε τυποποιημένη μορφή, μπορεί στη συνέχεια αυτό το μήνυμα να ενσωματωθεί - δημοσιευτεί απευθείας σε ιστοσελίδες ή εφαρμογές τρίτων.

Η αρχιτεκτονική SOAP αποτελείται από αρκετά στρώματα προδιαγραφών. Αυτά σχετίζονται με τη μορφή του μηνύματος, την υποκείμενη πρωτοκόλλων σύνδεση της μεταφοράς, τα μοντέλα επεξεργασίας μηνυμάτων και λεπτομέρειες για τυχόν επεκτασιμότητα του πρωτοκόλλου.

Μετά την θέσπιση του SOAP για πρώτη φορά και την πάροδο κάποιου χρόνου, το SOAP έγινε το υποκείμενο στρώμα ενός πιο σύνθετου συνόλου Υπηρεσιών Ιστού, βασισμένο στην Web Services Description Language (WSDL) και στην Universal Description, Discovery, and Integration (UDDI). Αυτές οι υπηρεσίες, ιδίως η UDDI, αποδείχθηκε ότι είναι πολύ μικρότερου ενδιαφέροντος και στην πορεία χάσανε την αρχική τους αίγλη σαν υποσύνολα όλου τους συστήματος ανάπτυξης Υπηρεσιών Web. Παρόλα αυτά μια καλή γνώση και εκτίμηση αυτών δίνει πληρέστερη κατανόηση του αναμενόμενου ρόλου του SOAP σε σύγκριση με το πώς οι Υπηρεσίες Web πραγματικά εξελίχθηκαν.

3.2.1 Προδιαγραφή

Η προδιαγραφή SOAP καθορίζει σαφώς το πλαίσιο ανταλλαγής μηνυμάτων και αυτό αποτελείται από τα εξής μοντέλα:

- Το μοντέλο επεξεργασίας SOAP που καθορίζει τους κανόνες για την επεξεργασία ενός μηνύματος SOAP.
- Στο μοντέλο επεκτασιμότητας του SOAP, προσδιορίζονται οι έννοιες της SOAP, τα χαρακτηριστικά και οι SOAP ενότητες.
- Το υποκείμενο πρωτόκολλο δεσμευτικού πλαισίου του SOAP, περιγράφει τους κανόνες για τον καθορισμό ενός δεσμευτικού σε ένα υποκείμενο πρωτόκολλο που μπορεί να χρησιμοποιηθεί για την ανταλλαγή μηνυμάτων SOAP μεταξύ των κόμβων SOAP.
- Η κατασκευή του μηνύματος SOAP καθορίζει τη δομή ενός μηνύματος SOAP.

3.2.2 Επεξεργασία μοντέλου

Το μοντέλο επεξεργασίας SOAP περιγράφει ένα κατανεμημένο μοντέλο επεξεργασίας, τους συμμετέχοντες, τους κόμβους SOAP και πώς ένας δέκτης SOAP επεξεργάζεται ένα μήνυμα SOAP. Ορίζονται οι ακόλουθοι κόμβοι SOAP:

- Αποστολέας SOAP

Είναι ένας κόμβος SOAP που μεταδίδει ένα μήνυμα SOAP.

- Δέκτης SOAP

Ένας κόμβος SOAP που δέχεται ένα μήνυμα SOAP.

- Μονοπάτι μηνύματος SOAP

Το σύνολο των SOAP κόμβων μέσω των οποίων ένα μήνυμα SOAP περνά.

- Αρχικός αποστολέας SOAP (εντολέας)

Ο αποστολέας SOAP από τον οποίο προέρχεται αρχικά ένα μήνυμα SOAP στο σημείο εκκίνησης ενός μονοπατιού μηνύματος SOAP.

- Ενδιάμεσος SOAP

Ένας μεσίτης SOAP είναι ταυτόχρονα δέκτης μηνύματος SOAP και αποστολέας μηνύματος SOAP και μπορεί να ανακαλυφθεί/στοχευθεί από ένα μήνυμα SOAP. Επεξεργάζεται τα μπλοκ κεφαλίδας του SOAP που απευθύνονται σε αυτόν και λαμβάνει μέτρα για να προωθήσει το μήνυμα SOAP προς μια τελική θέση δέκτη του SOAP.

- Τελικός δέκτης SOAP

Πρόκειται για τον δέκτη SOAP που είναι ο τελικός προορισμός ενός μηνύματος SOAP. Είναι υπεύθυνος για την επεξεργασία του περιεχομένου του σώματος SOAP και για κάθε μπλοκ κεφαλίδας SOAP που απευθύνεται σε αυτόν. Σε ορισμένες περιπτώσεις, ένα μήνυμα SOAP μπορεί να μην καταλήξει στον τελικό δέκτη SOAP, για παράδειγμα εξαιτίας ενός προβλήματος σε έναν ενδιάμεσο SOAP. Επίσης ένας τελικός δέκτης SOAP δεν μπορεί να είναι και ενδιάμεσος SOAP για το ίδιο μήνυμα SOAP.

3.2.3 Μέθοδοι μεταφοράς

Τόσο το SMTP αλλά και το HTTP είναι έγκυρα και αξιόπιστα πρωτόκολλα επιπέδου εφαρμογής, που χρησιμοποιούνται ως μεταφορείς για κάποιο μήνυμα SOAP. Παρόλα αυτά η HTTP έχει κερδίσει την ευρύτερη αποδοχή, δεδομένου ότι λειτουργεί καλά με τις υποδομές του σημερινού Διαδικτύου. Συγκεκριμένα, το HTTP λειτουργεί καλά με τα σημερινά εξελιγμένα τείχη προστασίας (firewalls) του δικτύου. Τα μηνύματα SOAP μπορούν επίσης να χρησιμοποιηθούν μέσω πρωτοκόλλου HTTPS[6], είτε απλό, ή τύπου αμοιβαίου ελέγχου μηχανισμού πιστοποίησης ταυτότητας. Αυτή είναι η υποστηριζόμενη WS-I μέθοδος που παρέχει ασφάλεια στην υπηρεσία Web, όπως αναφέρεται στο WS-I Basic Profile 1.1.

Αυτό το θέμα με την ασφάλεια είναι ένα σημαντικό πλεονέκτημα έναντι άλλων πρωτοκόλλων διανομής όπως τα GIOP / IIOP ή DCOM που συνήθως φιλτράρονται και παρεμποδίζονται από τα τείχη προστασίας. Μετάδοση SOAP πάνω από AMQP είναι ακόμα μια επιλογή και υποστηρίζεται από κάποιες υλοποιήσεις.

3.2.4 Μορφή μηνύματος

Η XML επιλέχθηκε ως η πρότυπη μορφή μηνύματος, λόγω της ευρείας χρήσης της από

6 είναι το ίδιο πρωτόκολλο με το HTTP και εργάζεται στο επίπεδο εφαρμογής της στοίβας των πρωτοκόλλων δικτύου, αλλά χρησιμοποιεί ένα κρυπτογραφημένο πρωτόκολλο μεταφοράς για την ασφάλεια των δεδομένων που μεταφέρει. Χρησιμοποιείται σε εφαρμογές οι οποίες μεταφέρουν ευαίσθητα δεδομένα και πληροφορίες, όπως αριθμούς πιστωτικών καρτών ή γενικότερα σε συναλλαγές μέσω Internet, κλπ.

μεγάλες εταιρείες και τις αναπτυξιακές προσπάθειες ανοικτού κώδικα. Επιπλέον, μια μεγάλη ποικιλία από ελεύθερα διαθέσιμα εργαλεία διευκολύνει σημαντικά τη μετάδοση ενός μηνύματος XML σε μια βασιζόμενη σε SOAP υλοποίηση. Η κάπως χρονοβόρα σύνταξη της XML μπορεί να είναι και πλεονέκτημα και μειονέκτημα. Ενώ η μορφή της προωθεί την αναγνωσιμότητα του κώδικα για τους ανθρώπους, άρα διευκολύνει τον εντοπισμό σφαλμάτων, και αποφεύγει τα προβλήματα δια-λειτουργικότητας, μπορεί να επιβραδύνει την ταχύτητα επεξεργασίας. Για παράδειγμα οι CORBA, GIOP, ICE, και DCOM χρησιμοποιούν πολύ πιο σύντομες, δυαδικές μορφές μηνυμάτων. Από την άλλη πλευρά, συσκευές υλικού είναι διαθέσιμες για την επιτάχυνση της επεξεργασίας των μηνυμάτων XML. Τύποι Binary XML επίσης διερευνώνται ως μέσο για τον εξορθολογισμό των απαιτήσεων απόδοσης της XML.

Παραθέτουμε αμέσως τώρα ένα μικρό παράδειγμα με τον κώδικα ενός μηνύματος SOAP σε σύνταξη XML φυσικά.

Περιγραφή:

POST /InStock HTTP/1.1

Host: www.example.org

Content-Type: application/soap+xml; charset=utf-8

Content-Length: 299

Κώδικας:

```
<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <soap:Header>
  </soap:Header>
  <soap:Body>
```

```
    <m:GetStockPrice xmlns:m="http://www.example.org/stock">
      <m:StockName>IBM</m:StockName>
    </m:GetStockPrice>
  </soap:Body>
</soap:Envelope>
```

3.2.5 Πλεονεκτήματα

Τα μηνύματα SOAP είναι ευέλικτα ώστε να επιτρέπεται η χρήση διαφορετικών πρωτοκόλλων μεταφοράς. Οι συνηθισμένες στοίβες χρησιμοποιούν HTTP ως πρωτόκολλο μεταφοράς, αλλά και άλλα πρωτόκολλα όπως το JMS και το SMTP, είναι επίσης δυνατό να χρησιμοποιηθούν.

Δεδομένου ότι οι σήραγγες του μοντέλου SOAP αριστεύουν συνεργαζόμενες με το HTTP get/responce μοντέλο, μπορεί εύκολα να μεταδοθεί το μήνυμα σε σχέση με τις υπάρχουσες υποδομές firewalls και proxies, χωρίς τροποποιήσεις. Μπορεί επίσης να χρησιμοποιήσει τις υπάρχουσες υποδομές χωρίς να απαιτούνται αλλαγές ή τροποποιήσεις που αυξάνουν και το κόστος σε υλικό (hardware).

3.2.6 Μειονεκτήματα

Λόγω της λεπτομερούς μορφής της XML, η SOAP μπορεί να είναι σημαντικά πιο αργή από ανταγωνιστικές middleware τεχνολογίες όπως η CORBA. Αυτό δεν αποτελεί πρόβλημα όταν αποστέλλονται μόνο μικρά μηνύματα. Για τη βελτίωση των επιδόσεων στην ειδική περίπτωση της XML με ενσωματωμένα δυαδικά αντικείμενα, εισήχθη μηχανισμός βελτιστοποίησης μετάδοσης μηνυμάτων.

Όταν οι υπηρεσίες Web βασίζονται σε HTTP πρωτόκολλο για την μεταφορά των μηνυμάτων και δεν χρησιμοποιούν WS-Addressing ή ESB, οι ρόλοι των μερών που αλληλεπιδρούν είναι καθορισμένοι. Μόνο ένα μέρος (ο πελάτης) μπορεί να χρησιμοποιήσει τις υπηρεσίες των άλλων. Οι προγραμματιστές πρέπει να χρησιμοποιήσουν ελέγχους αναμονής κατάστασης αντί της απλής κοινοποίησης, σε αυτές τις περιπτώσεις.

3.3. Web Services Description Language (WSDL)

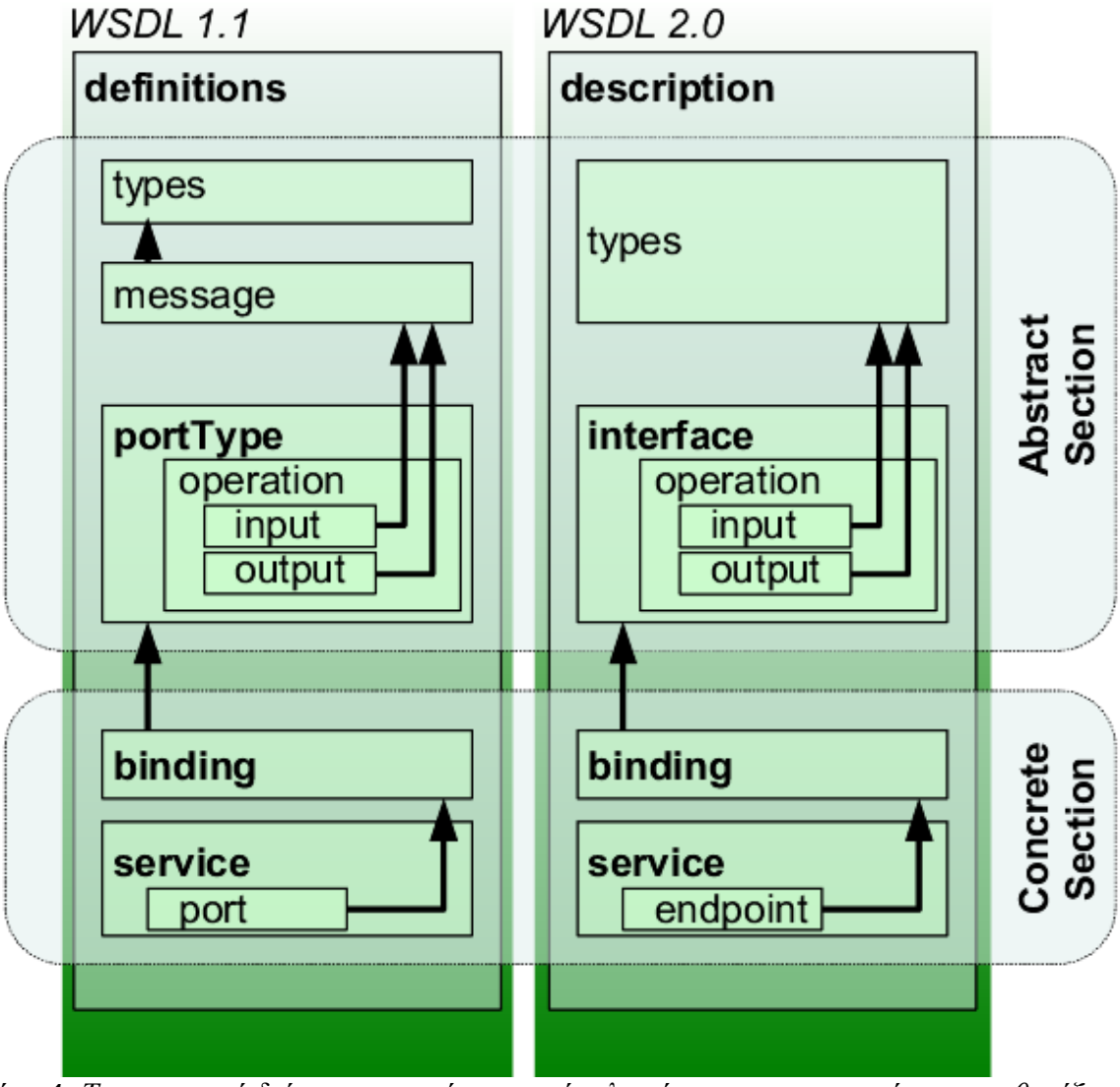
Η Web Services Description Language (WSDL, προφέρεται «wiz-del») είναι μια γλώσσα βασισμένη σε XML η οποία παρέχει ένα πρότυπο για την περιγραφή υπηρεσιών Web. Η έννοια του ακρωνυμίου έχει αλλάξει από την έκδοση 1.1, όπου το D σήμαινε Definition, δηλαδή ορισμός. Τώρα σημαίνει περιγραφή.

Η WSDL περιγράφει τις υπηρεσίες ως συλλογές τελικών σημείων του δικτύου, ή "πόρτες" ή "λιμένες". Η προδιαγραφή WSDL παρέχει μια μορφή XML για έγγραφα για το σκοπό αυτό. Οι αφηρημένοι ορισμοί των λιμένων και των μηνυμάτων διαχωρίζεται από την συγκεκριμένη χρήση τους, επιτρέποντας την επαναχρησιμοποίηση αυτών των ορισμών. Ένας λιμένας ορίζεται συνδέοντας μια διεύθυνση δικτύου με ένα επαναχρησιμοποιήσιμο δεσμευτικό, και μια συλλογή λιμένων ορίζει μια υπηρεσία Ιστού. Τα μηνύματα είναι αφηρημένες περιγραφές των δεδομένων που ανταλλάσσονται, και οι τύποι των λιμένων είναι αφηρημένες συλλογές των υποστηριζόμενων λειτουργιών. Το συγκεκριμένο πρωτόκολλο και οι προδιαγραφές μορφών δεδομένων για ένα συγκεκριμένο τύπο λιμένα αποτελεί ένα επαναχρησιμοποιήσιμο δεσμευτικό, όπου οι εργασίες και τα μηνύματα στη συνέχεια, είναι άρρηκτα συνδεδεμένες με ένα συγκεκριμένο πρωτόκολλο δικτύου και μια σαφώς ορισμένη μορφή μηνύματος. Με τον τρόπο αυτό η WSDL περιγράφει τη δημόσια διεπαφή στην υπηρεσία Web.

Η WSDL χρησιμοποιείται συχνά σε συνδυασμό με SOAP και ένα σχήμα XML για την παροχή υπηρεσιών Ιστού μέσω του Internet. Όταν ένα πρόγραμμα πελάτη συνδέεται σε μια υπηρεσία Web μπορεί να διαβάσει το WSDL αρχείο για να προσδιορίσει τι είδους λειτουργίες είναι διαθέσιμες στο διακομιστή. Τυχόν ειδικές μορφές τύπων δεδομένων που χρησιμοποιούνται έχουν ενσωματωθεί στο αρχείο WSDL, περιγραφόμενες στο XML σχήμα. Το πρόγραμμα πελάτη μπορεί στη συνέχεια να χρησιμοποιεί το πρωτόκολλο SOAP για να καλέσει μία από τις λειτουργίες που αναφέρονται στο αρχείο WSDL.

Η σημερινή διατύπωση της έκδοσης του WSDL είναι η 2.0. Η έκδοση 1.1 δεν είχει εγκριθεί από το W3C αλλά η έκδοση 2.0 είναι πλέον η προτεινόμενη έκδοση του οργανισμού W3C. Ουσιαστικά πρόκειται για την έκδοση 1.2 που μετονομάστηκε σε WSDL 2.0, λόγω των

ουσιαστικών διαφορών του από την WSDL 1.1. Με την αποδοχή όλων των μεθόδων αίτησης HTTP (όχι μόνο των GET και POST όπως στην έκδοση 1.1), η WSDL 2.0 προσφέρει καλύτερη υποστήριξη για RESTful Web Services, και είναι πολύ πιο απλή στην εφαρμογή της. Εντούτοις, η υποστήριξη για την προδιαγραφή αυτή είναι ακόμη ελλιπής στα πακέτα ανάπτυξης λογισμικού για Web Services τα οποία συχνά προσφέρουν εργαλεία μόνο για την υλοποίηση WSDL 1.1.



Εικόνα 4: Το σχηματικό διάγραμμα παρέχει μια εύκολη σύγκριση των εννοιών που καθορίζονται από WSDL 1,1 και 2,0 WSDL έγγραφα. Πηγή: wikipedia.org

3.3.1 Περιγραφή Αντικειμένων σε WSDL 1.1 & WSDL 2.0

Service: Η υπηρεσία μπορεί να θεωρηθεί ως ένα δοχείο για μια σειρά από λειτουργίες του συστήματος που έχουν εκτεθεί σε Web-based πρωτόκολλα.

Port ή Endpoint: Η Πόρτα ή τελικό σημείο δεν κάνει τίποτα περισσότερο από τον καθορισμό του σημείου διεύθυνσης ή σύνδεσης με μια υπηρεσία Web. Είναι συνήθως ένα απλό URL (HTTP string).

Binding: Το δεσμευτικό προδιαγράφει τη διεπαφή, καθώς και τον καθορισμό των SOAP δεσμευτικών (RPC / Document) όπως και των μεταφορών (πρωτόκολλο SOAP). Η δεσμευτική ενότητα καθορίζει επίσης τις λειτουργίες.

PortType ή Interface: Το στοιχείο port type, που μετονομάζεται σε interface στην προδιαγραφή WSDL 2.0, ορίζει μια υπηρεσία Web, τις λειτουργίες που μπορούν να εκτελεστούν, καθώς και τα μηνύματα που χρησιμοποιούνται για την εκτέλεση της λειτουργίας.

Operation: Κάθε εργασία μπορεί να συγκριθεί με μια μέθοδο ή κλήση συνάρτησης σε μια παραδοσιακή γλώσσα προγραμματισμού. Εδώ καθορίζονται οι δράσεις SOAP, ο τρόπος που το μήνυμα είναι κωδικοποιημένο.

Message: Δεν χρησιμοποιείται στην προδιαγραφή WSDL 2.0. Συνήθως, ένα μήνυμα αντιστοιχεί σε μια λειτουργία. Το μήνυμα περιέχει τις πληροφορίες που απαιτούνται για την εκτέλεση της λειτουργίας. Κάθε μήνυμα αποτελείται από ένα ή περισσότερα λογικά τμήματα. Τα μηνύματα αφαιρέθηκαν στην προδιαγραφή WSDL 2.0, στην οποία το XML σχήμα αποδίδει για ορισμένους φορείς εισόδων τα απαραίτητα σφάλματα ή εξόδους που αναφέρονται απλά και άμεσα.

Types: Ο σκοπός των τύπων στην WSDL είναι να περιγράψει τα δεδομένα. Το XML σχήμα χρησιμοποιείται (με αναφορές ή παραθέσεις) για το σκοπό αυτό.

3.3.2 Παραθέτουμε ένα πλήρες παράδειγμα Κώδικα με αντικείμενα WSDL

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

```
<description xmlns="http://www.w3.org/ns/wsdl"
             xmlns:tns="http://www.tmsws.com/wsdl20sample"
             xmlns:whttp="http://schemas.xmlsoap.org/wsdl/http/"
             xmlns:wsoap="http://schemas.xmlsoap.org/wsdl/soap/"
             targetNamespace="http://www.tmsws.com/wsdl20sample">



<!-- Abstract type -->
  <types>
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
               xmlns="http://www.tmsws.com/wsdl20sample"
               targetNamespace="http://www.example.com/wsdl20sample">


      <xs:element name="request">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="header" maxOccurs="unbounded">
              <xs:complexType>
                <xs:simpleContent>
                  <xs:extension base="xs:string">
                    <xs:attribute name="name" type="xs:string" use="required"/>
                  </xs:extension>
                </xs:simpleContent>
              </xs:complexType>
            </xs:element>
            <xs:element name="body" type="xs:anyType" minOccurs="0"/>
```

```
        </xs:sequence>
        <xs:attribute name="method" type="xs:string" use="required"/>
        <xs:attribute name="uri" type="xs:anyURI" use="required"/>
      </xs:complexType>
    </xs:element>

    <xs:element name="response">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="header" maxOccurs="unbounded">
            <xs:complexType>
              <xs:simpleContent>
                <xs:extension base="xs:string">
                  <xs:attribute name="name" type="xs:string" use="required"/>
                </xs:extension>
              </xs:simpleContent>
            </xs:complexType>
          </xs:element>
          <xs:element name="body" type="xs:anyType" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="status-code" type="xs:anySimpleType" use="required"/>
        <xs:attribute name="response-phrase" use="required"/>
      </xs:complexType>
    </xs:element>
  </xs:schema>
</types>
```

```
<!-- Abstract interfaces -->
  <interface name="RESTfulInterface">
    <fault name="ClientError" element="tns:response"/>
    <fault name="ServerError" element="tns:response"/>
    <fault name="Redirection" element="tns:response"/>
    <operation name="Get" pattern="http://www.w3.org/ns/wsdl/in-out">
      <input messageLabel="In" element="tns:request"/>
      <output messageLabel="Out" element="tns:response"/>
    </operation>
    <operation name="Post" pattern="http://www.w3.org/ns/wsdl/in-out">
      <input messageLabel="In" element="tns:request"/>
      <output messageLabel="Out" element="tns:response"/>
    </operation>
    <operation name="Put" pattern="http://www.w3.org/ns/wsdl/in-out">
      <input messageLabel="In" element="tns:request"/>
      <output messageLabel="Out" element="tns:response"/>
    </operation>
    <operation name="Delete" pattern="http://www.w3.org/ns/wsdl/in-out">
      <input messageLabel="In" element="tns:request"/>
      <output messageLabel="Out" element="tns:response"/>
    </operation>
  </interface>
```

```
<!-- Concrete Binding Over HTTP -->
  <binding name="RESTfulInterfaceHttpBinding" interface="tns:RESTfulInterface"
        type="http://www.w3.org/ns/wsdl/http">
    <operation ref="tns:Get" whttp:method="GET"/>
    <operation ref="tns:Post" whttp:method="POST"
          whttp:inputSerialization="application/x-www-form-urlencoded"/>
    <operation ref="tns:Put" whttp:method="PUT"
          whttp:inputSerialization="application/x-www-form-urlencoded"/>
    <operation ref="tns:Delete" whttp:method="DELETE"/>
  </binding>

<!-- Concrete Binding with SOAP-->
  <binding name="RESTfulInterfaceSoapBinding" interface="tns:RESTfulInterface"
        type="http://www.w3.org/ns/wsdl/soap"
        wsoap:protocol="http://www.w3.org/2003/05/soap/bindings/HTTP/"
        wsoap:mepDefault="http://www.w3.org/2003/05/soap/mep/request-response">
    <operation ref="tns:Get" />
    <operation ref="tns:Post" />
    <operation ref="tns:Put" />
    <operation ref="tns:Delete" />
  </binding>


<!-- Web Service offering endpoints for both bindings-->
  <service name="RESTfulService" interface="tns:RESTfulInterface">
```

```
    <endpoint name="RESTfulServiceHttpEndpoint"
              binding="tns:RESTfulInterfaceHttpBinding"
              address="http://www.example.com/rest/"/>
    <endpoint name="RESTfulServiceSoapEndpoint"
              binding="tns:RESTfulInterfaceSoapBinding"
              address="http://www.example.com/soap/"/>
  </service>
</description>
```

3.4. Universal Description, Discovery, and Integration (UDDI)

Το UDDI (προφέρεται Yu-di) είναι ένα μητρώο ανεξάρτητο πλατφόρμας, βασισμένο σε Extensible Markup Language (XML) ανεπτυγμένο για να στεγάσει τις επιχειρήσεις από όλο τον κόσμο, σε λίστα στο Διαδίκτυο, περιέχοντας επίσης ένα μηχανισμό για την καταγραφή και τον εντοπισμό των εφαρμογών των Web Services. Το UDDI είναι μια ανοικτή πρωτοβουλία της βιομηχανίας, υπό την αιγίδα του Οργανισμού για την Προώθηση Προτύπων των Δομημένων Πληροφοριών (OASIS), που επιτρέπει στις επιχειρήσεις να δημοσιεύουν προγράμματα παροχής υπηρεσιών, να ανακαλύπτουν η μια την άλλη, επίσης καθορίζει πώς οι υπηρεσίες ή οι εφαρμογές λογισμικού αλληλεπιδρούν μέσω του Διαδικτύου.

Το UDDI είχε αρχικά προταθεί ως βασικό πρότυπο των Web υπηρεσιών. Είναι σχεδιασμένο για να "ανακρίνεται" από τα μηνύματα SOAP και να παρέχει πρόσβαση σε έγγραφα Web Services Description Language (WSDL), που περιγράφουν τους δεσμούς πρωτοκόλλων και μορφοποίησης μηνυμάτων, που απαιτούνται για την αλληλεπίδραση των διαδικτυακών υπηρεσιών που απαριθμούνται στον κατάλογο του.

3.4.1 Η ιστορία του UDDI

Το UDDI γράφτηκε τον Αύγουστο του 2000, σε μια εποχή που οι συγγραφείς του οραματίζονταν έναν κόσμο στον οποίο οι καταναλωτές των Web Services θα συνδέονται με τους παρόχους μέσω ενός δημόσιου ή ιδιωτικού δυναμικού συστήματος μεσιτείας. Σύμφωνα με αυτό το σχέδιο, όποιος χρειάζονταν μια υπηρεσία, όπως για παράδειγμα η επικύρωση μιας πιστωτικής του κάρτας, θα ζητούσε μεσολάβηση για την υπηρεσία και θα επέλεγε μία

υποστηριζόμενη ή επιθυμητή SOAP ή άλλη υπηρεσία διασύνδεσης που πληροί όλα τα ανάλογα κριτήρια. Σε έναν τέτοιο κόσμο, οι δημόσιοι κόμβοι UDDI θα ήταν κρίσιμοι για όλους. Για τον καταναλωτή, για δημόσιους ή άλλους ανοικτούς διαμεσολαβητές, το UDDI θα επέστρεφε μόνο τις υπηρεσίες που αναφέρονται ως δημόσια πρόσβαση και η εύρεσή τους θα ήταν εφικτή σε όλους. Για έναν παραγωγό Web υπηρεσιών, το να πάρει μια καλή τοποθέτηση στις λίστες του μητρώου, επικαλούμενος μεταδεδομένα έγκυρων κατηγοριών και ορθών δεικτών θα ήταν κρίσιμης σημασίας για την αποτελεσματική του τοποθέτηση και την σωστή πρόσβαση στις δικές του υπηρεσίες.

Το UDDI περιελήφθη στο πρότυπο Web Services Interoperability (WS-I) ως κεντρικός πυλώνας της υποδομής των Web Services. Οι προδιαγραφές UDDI υποστήριξαν την δημόσια πρόσβαση στην Καθολική Γραμματεία Επιχειρήσεων (Universal Business Registry) στην οποία ένα σύστημα ονοματολογίας χτίστηκε γύρω από την μεσιτική υπηρεσία UDDI.

Το UDDI όμως δεν υιοθετήθηκε ποτέ ευρέως με τον τρόπο που οι σχεδιαστές του ήλπιζαν. Η IBM, η Microsoft και η SAP ανακοίνωσαν ότι κλείνουν δημόσιους κόμβους UDDI τον Ιανουάριο του 2006. Γενικά αυτή η υλοποίηση βρήκε πολύ κόσμο του Διαδικτύου αντίθετο. Το κίνημα του open source αλλά και πολλά κινήματα που υποστηρίζουν την ελευθερία και την διατήρηση της ανεξαρτησίας στο Διαδίκτυο επεσήμαναν ότι μια ευρεία αποδοχή αυτής της υλοποίησης θα οδηγούσε ουσιαστικά στον πλήρη έλεγχο και στην ουσιαστική ιδιωτικοποίηση του από λίγες μεγάλες πολυεθνικές.

Η ομάδα που καθόριζε τις προδιαγραφές του UDDI, το OASIS, δηλαδή η Universal Description, Discovery, and Integration (UDDI) τεχνική επιτροπή, ψήφισε τις ανάλογες διατάξεις ώστε να ολοκληρώσει τις εργασίες της στα τέλη του 2007 και έχει πλέον κλείσει. Τον Σεπτέμβριο του 2010, η Microsoft ανακοίνωσε ότι έχει πλέον αφαιρέσει τις UDDI υπηρεσίες από τις μελλοντικές εκδόσεις του λειτουργικού συστήματος Windows Server. Αυτή η δυνατότητα είναι πλέον πιθανό να μετακινηθεί σε Biztalk.

Κάποιοι ισχυρίζονται ότι το μέλλον, ή αν θέλετε ο πιο κοινός τόπος για ένα σύστημα UDDI, είναι να υπάρξει στο εσωτερικό μιας επιχείρησης όπου θα μπορούσε να χρησιμοποιηθεί για

την δυναμική σύνδεση/ζεύξη συστημάτων client με υλοποιήσεις νέες ή παλαιότερες. Μάλλον μεγάλο μέρος των αναζητήσεων μεταδεδομένων που επιτρέπονται στο UDDI δεν χρησιμοποιήθηκε για αυτό το σχετικά απλό ρόλο μέχρι σήμερα.

3.4.2 Η δομή του

Η καταχώριση των επιχειρήσεων στο UDDI αποτελείται από τρία στοιχεία:

- White Pages: διεύθυνση, στοιχεία επικοινωνίας, καθώς και γνωστά αναγνωριστικά.
- Yellow Pages: βιομηχανική κατηγοριοποίηση με βάση αποδεκτά πρότυπα ταξινόμησης.
- Green Pages: τεχνικές πληροφορίες για τις διαδικτυακές υπηρεσίες που εκτίθενται από την επιχείρηση.

Οι White Pages περιέχουν πληροφορίες για την επιχείρηση που παρέχει την υπηρεσία. Αυτές περιλαμβάνουν το όνομα της επιχείρησης και την περιγραφή της επιχείρησης - ενδεχομένως σε πολλές γλώσσες. Χρησιμοποιώντας αυτές τις πληροφορίες, είναι δυνατό να βρείτε μια υπηρεσία για την οποία κάποιες πληροφορίες είναι ήδη γνωστές (για παράδειγμα, εντοπίζοντας μια υπηρεσία με βάση το όνομα του παρόχου).

Πληροφορίες επικοινωνίας με την επιχείρηση αυτή επίσης διατίθενται, για παράδειγμα, διεύθυνση, αριθμός τηλεφώνου και άλλες πληροφορίες, όπως πιθανό και μερικά πιο επίσημα στοιχεία για την διαπίστευση της ταυτότητάς της.

Οι Yellow Pages παρέχουν μια κατάταξη των υπηρεσιών ή των επιχειρήσεων, με βάση επίσημα πρότυπα ταξινόμησης. Αυτές περιλαμβάνουν την πρότυπη βιομηχανική ονοματολογία (SIC), το βόρειο Αμερικανικό Σύστημα Ταξινόμησης βιομηχανίας (NAICS), ή τον Κώδικα των Ηνωμένων Εθνών για τυποποιημένα προϊόντα και υπηρεσίες (UNSPSC).

Επειδή μια ενιαία επιχείρηση μπορεί να παρέχει μια σειρά υπηρεσιών, είναι δυνατό να υπάρχουν αρκετές καταχωρήσεις Yellow Page (Χρυσού Οδηγού - περιγραφή κάθε μιας

υπηρεσίας) που συνδέεται με μία λευκή σελίδα (η οποία δίνει γενικές πληροφορίες για την επιχείρηση).

Οι Green Pages χρησιμοποιούνται για να περιγράψουν πώς αποκτάται πρόσβαση σε μια Web Service, με πληροφορίες για τις συνδέσεις της υπηρεσίας. Ορισμένες από τις πληροφορίες είναι σχετικές με το Web Service. Για παράδειγμα η διεύθυνση της υπηρεσίας, οι παράμετροι, καθώς και παραπομπές σε ορισμένες προδιαγραφές της διεπαφής. Άλλες πληροφορίες δεν σχετίζονται άμεσα με το Web Service, αυτές περιλαμβάνουν e-mail, FTP, CORBA και λεπτομέρειες τηλεφώνου σχετικές με την υπηρεσία. Επειδή ένα Web Service μπορεί να έχει πολλαπλές συνδέσεις (όπως αυτές ορίζονται στην WSDL περιγραφή της), μια υπηρεσία μπορεί να κατέχει πολλές πράσινες σελίδες, καθώς κάθε δεσμευτικό θα πρέπει να προσεγγιστεί με διαφορετικό τρόπο.

3.4.3 Τελικά Συμπεράσματα

Οι κόμβοι UDDI είναι διακομιστές που υποστηρίζουν την προδιαγραφή UDDI και ανήκουν σε ένα μητρώο UDDI, ενώ UDDI μητρώα είναι συλλογές από έναν ή περισσότερους κόμβους.

Το SOAP είναι ένα βασισμένο σε XML πρωτόκολλο για την ανταλλαγή μηνυμάτων μεταξύ του αιτούντος και του παρόχου μιας υπηρεσίας Web. Ο πάροχος δημοσιεύει το WSDL στο UDDI και ο αιτών μπορεί να ενταχθεί σε αυτό χρησιμοποιώντας SOAP.

4. Οι Web Services στην Ελλάδα.

Τα Web Services όπως είδαμε είναι μια καινοτομική αρχιτεκτονική με την οποία παρέχεται η δυνατότητα δημιουργίας και χρήσης ηλεκτρονικών υπηρεσιών στο διαδίκτυο με απλό και οικονομικό τρόπο. Ακόμη και στην χώρα μας την Ελλάδα που παραδοσιακά βρίσκει κάπως καθυστερημένα τον δρόμο της προς την τεχνολογία υπήρξε έντονο ενδιαφέρον για αυτή την τεχνολογία από την αρχή. Επιδοτήσεις της Ευρωπαϊκής Ένωσης και προγράμματα όπως το "Δικτυωθείτε" έδωσαν την ευκαιρία, ευτυχώς, σε πολλές Μικρό Μεσαίες Επιχειρήσεις να αναπτυχθούν Διαδικτυακά, να αποκτήσουν εξοπλισμό και τεχνογνωσία, ώστε να χρησιμοποιήσουν ή να αναπτύξουν Web Services.

Δεν είναι ιδιαίτερα δύσκολο, για μια ελληνική επιχείρηση να μπορέσει να παρέχει μία παραδοσιακή υπηρεσία της ως Web Service. Επίσης πολύ εύκολο καθίσταται να ακολουθήσει μία άλλη επιχείρηση-προμηθευτή της, χρησιμοποιώντας κάποια έτοιμη υπηρεσία που η εταιρία προμηθευτής προσφέρει από το διαδίκτυο. Μεγάλη πρόοδο σημείωσε μέσω των Web Services το η-Επιχειρείν[7], αλλά πραγματική επανάσταση έγινε στον τομέα της η-Διακυβέρνησης στη χώρα μας μέσω αυτών των τεχνολογιών.

Πριν περίπου 8 χρόνια το Internet αποτελούσε για τις επιχειρήσεις στην Ελλάδα κυρίως μέσο προβολής και δημοσιοποίησης των δραστηριοτήτων τους. Αυτή η κατάσταση άλλαξε πολύ δύσκολα κυρίως λόγω της έλλειψης υποδομών και ελλιπούς παιδείας στις τεχνολογίες. Τα τελευταία δέκα χρόνια όμως πολλές επιχειρήσεις παγκοσμίως άρχισαν να χρησιμοποιούν το Internet ως μέσο παροχής υπηρεσιών προς τους πελάτες τους αλλά και προς άλλες επιχειρήσεις. Σιγά-σιγά ακολουθούμε κι εμείς αναπτυσσόμενοι ταχύτερα τα τελευταία 5 χρόνια.

Με την εξέλιξη αυτή δημιουργήθηκε ένα σύνολο από θέματα που είχαν να κάνουν με την ευχρηστία, τη λειτουργικότητα και την απλότητα των παρεχόμενων υπηρεσιών. Και ενώ οι προσπάθειες που έγιναν για να δοθούν λύσεις ήταν πολλές, το αποτέλεσμα ήταν το ίδιο:

7 Αρκετές επιχειρήσεις ξεκίνησαν νωρίς και εύκολα τις ηλεκτρονικές τους δραστηριότητες και στη χώρα μας, κυρίως με τα προγράμματα Δικτυωθείτε και τα προγράμματα για τον τουρισμό & τα ξενοδοχεία. Πηγή: sepe.gr

υπήρχε ένα μεγάλο σύνολο από έτοιμες υπηρεσίες στο Internet αλλά η χρησιμοποίησή τους ήταν πάρα πολύ μικρή.

Οι web services όμως, ήρθαν για να δώσουν λύση σε προβλήματα ευχρηστίας και λειτουργικότητας των ηλεκτρονικών υπηρεσιών. Μέσα από την αρχιτεκτονική των web services ορίζεται ένα σύνολο από προδιαγραφές και κανόνες με τους οποίους είναι δυνατή η δημιουργία και η παροχή ηλεκτρονικών υπηρεσιών μέσα στο Internet με αρκετά απλό τρόπο. Όσο τα πράγματα γίνονται απλούστερα, τόσο γίνονται φθηνότερα, πιο προσιτά στους λιγότερο γνωστικούς με το αντικείμενο και άρα οι επενδύσεις πιο προσιτές και εφικτές.

Έτσι τα τελευταία τρία χρόνια οι ηλεκτρονικές υπηρεσίες πήραν νέα διάσταση και πλέον κάθε επιχείρηση μπορεί σχετικά εύκολα να δημιουργεί και να παρέχει υπηρεσίες στο Internet. Επίσης μπορεί με ακόμα μεγαλύτερη ευκολία και ελάχιστο κόστος, να χρησιμοποιεί έτοιμες υπηρεσίες.

Η τεχνολογία των web services αφορά συνήθως τους προγραμματιστές εφαρμογών στο Internet και όχι τους απλούς χρήστες και χειριστές ηλεκτρονικών υπολογιστών. Όμως η γνώση της ύπαρξης της συγκεκριμένης τεχνολογίας και του γεγονότος ότι η ενσωμάτωσή της αποτελεί μία σχετικά εύκολη διαδικασία, πρέπει να περάσει σε κάθε δραστήριο επιχειρηματία στη χώρα μας. Με αυτό τον τρόπο οι επενδύσεις στην πληροφορική και στο Internet θα είναι πραγματικά αξιόλογες και ανταγωνιστικές τόσο για τα Ελληνικά όσο και για τα παγκόσμια δεδομένα.

Μέχρι πρόσφατα η δημιουργία και η παροχή υπηρεσιών από επιχειρήσεις στο Internet γίνονταν με ακαθόριστο τρόπο ο οποίος διέφερε από επιχείρηση σε επιχείρηση. Έτσι, ενώ υπήρχε ένα αρκετά μεγάλο σύνολο από παρεχόμενες υπηρεσίες στο Internet, για να μπορέσει κάποιος να τις χρησιμοποιήσει θα έπρεπε για κάθε μία υπηρεσία να μελετήσει τον τρόπο με τον οποίο θα την καλέσει, να ελέγξει αν χρησιμοποιούν το ίδιο πρωτόκολλο επικοινωνίας (TCP/IP, Http, κλπ). Γενικά ήταν απαραίτητο να προσαρμόσει όλο το σύστημά του, έτσι ώστε να γίνει συμβατό με αυτό του παροχέα της υπηρεσίας.

Για παράδειγμα ας υποθέσουμε ότι κάποια επιχείρηση ενδιαφερόταν να χρησιμοποιήσει μία υποτιθέμενη υπηρεσία που παρείχε το Εθνικό Κέντρο Βιβλίου. Αυτή η υπηρεσία παρουσίαζε τις βασικές πληροφορίες που ήταν καταχωρημένες (τίτλο και συγγραφέα) και όλες τις συνοδευτικές πληροφορίες (εκδοτικό οίκο, τιμή κλπ), για τα βιβλία δοθέντος του κωδικού ISBN.

Σε αυτή την περίπτωση ο προγραμματιστής της επιχείρησης θα έπρεπε στην ουσία να δημιουργήσει ένα σύστημα συμβατό με αυτό του Εθνικού Κέντρου Βιβλίου. Θα έπρεπε να λάβει υπόψη του το πρωτόκολλο επικοινωνίας αλλά και τις τεχνολογίες του τρόπου κλήσης των ερωτημάτων και κατόπιν να το προσαρμόσει στις ανάγκες του συστήματος της δικής του επιχείρησης.

Πολλές φορές αυτό ήταν πολύ δύσκολο, αν όχι ακατόρθωτο, και ακόμα περισσότερες φορές οι επιχειρήσεις σχεδίαζαν τα συστήματά τους έτσι ώστε να αποφεύγουν τέτοιου είδους συνεργασίες με ξένες πηγές για λόγους πολυπλοκότητας και γενικότερα για λόγους κόστους.

Τα πράγματα όμως τα τελευταία τρία χρόνια φαίνεται να παίρνουν διαφορετική τροπή. Σχεδόν όλες οι επιχειρήσεις που δημιουργούν υπηρεσίες στο Internet, βασίζονται σε μία κοινή αρχιτεκτονική ανάπτυξης, δημοσίευσης και εκμετάλλευσης των υπηρεσιών τους, όπως αυτή καθορίζεται από το W3C και ορίζεται ως η αρχιτεκτονική των web services. Τα πλεονεκτήματα είναι πολλά και δεν πρέπει να παραγνωρίζονται από βιαστικές αποφάσεις. Για παράδειγμα στην παραπάνω προσπάθεια σύνδεσης με το Εθνικό Κέντρο Βιβλίου ένας επιχειρηματίας που ενδιαφέρεται να πουλήσει βιβλία μέσω ενός ηλεκτρονικού καταστήματος δεν θα χρειαζόταν να χρησιμοποιεί περισσότερο προσωπικό για την εισαγωγή των βιβλίων στο σύστημά του. Αυτά μαζί με όλα τα χαρακτηριστικά τους, θα μπορούσαν να εισάγονται αυτόματα μέσω Web Service από το Κέντρο Βιβλίου όποτε χρειάζεται.

4.1. Αρχιτεκτονική και Πλεονεκτήματα

Η αρχιτεκτονική των web services παρέχει αρκετά πλεονεκτήματα μερικά από τα οποία

αναφέρονται παρακάτω :

Δια-λειτουργικότητα.

Ένα web service παρέχει ανεξαρτησία τόσο από λειτουργικό σύστημα όσο και από το hardware. Οποιοδήποτε πρόγραμμα που συμβαδίζει με αυτή τη τεχνολογία μπορεί πολύ εύκολα να προσπελάσει μία τέτοια υπηρεσία.

Ενσωμάτωση.

Σε ένα υπάρχον λογισμικό σύστημα που λειτουργεί μέσα στο Internet η δημιουργία ενός web service δεν απαιτεί αλλαγές στον μηχανισμό του συστήματος.

Διαθεσιμότητα και δημοσίευση.

Οι πληροφορίες για τα web services δημοσιεύονται οπότε η εύρεση και η χρήση τους μπορεί να είναι ταχύτατες.

Επέκταση.

Ένα έτοιμο web service είναι δυνατό να ανανεωθεί με εύκολο τρόπο παρέχοντας έτσι επιπρόσθετες υπηρεσίες στους χρήστες του.

Μικρό κόστος δημιουργίας και χρήσης.

Εφόσον σε ένα λογισμικό σύστημα υπάρχει ήδη κάποια διαδικασία που χρειάζεται να επεκταθεί σε on-line υπηρεσία, η δημιουργία του web service κοστίζει ελάχιστα. Επίσης το κόστος ενσωμάτωσης ενός web service σε κάποιο website ή σε δικτυακή εφαρμογή είναι πάρα πολύ μικρό. Ακόμα και στις περιπτώσεις που η χρήση κάποιου web service γίνεται με ενοικίαση σίγουρα το συνολικό κόστος της χρήσης είναι αρκετά πιο μικρό από το κόστος δημιουργίας της υπηρεσίας αυτής.

Χρήση λογισμικών συστημάτων.

Όλα τα λογισμικά συστήματα και ειδικότερα τα websites που χρησιμοποιούν έτοιμες υπηρεσίες γίνονται πιο λειτουργικά και πιο φιλικά αφού παρέχουν περισσότερες υπηρεσίες στους χρήστες.

Υπάρχει μία μεγάλη λίστα από έτοιμα web services που θα μπορούσε να χρησιμοποιήσει κανείς, ακόμα και εντελώς δωρεάν. Ψάχνοντας για παράδειγμα στη διεύθυνση http://www.webservicelist.com/ μπορούμε να βρούμε μία πληθώρα από web services συνοδευόμενα από το wsdl file καθώς και με on-line demo της λειτουργίας τους.

Στον πίνακα που ακολουθεί αναγράφονται κάποια ενδεικτικά παραδείγματα όπως τα πήραμε από την παραπάνω διεύθυνση.

Web Service URL

Country Population Lookup Service

Δέχεται το όνομα μίας χώρας και επιστρέφει τον πληθυσμό της

http://www.cs.uga.edu/~sent

BN Quote Service

Δέχεται το ISBN ενός βιβλίου και επιστρέφει την τιμή του

http://www.xmethods.com/help/addspecs.html

Rich Credit Card Validator

Ελέγχει τα στοιχεία μίας πιστωτικής κάρτας αν αντιστοιχούν σε υπάρχουσα πιστωτική.

http://www.richsolutions.com

SOAP SMS

Στέλνει κείμενα των χρηστών ως sms σε κινητά.

http://www.redcoal.com

Google Web Service

Απαντά σε ερωτήσεις εύρεσης σελίδων και επιστρέφει τα αποτελέσματα σε μορφή επεξεργάσιμη.

http://www.google.com/apis

FreshScore

Δίνει τα αποτελέσματα των αγώνων σε πραγματικό χρόνο. http://www.freshscore.com/service/FreshScoreLiveScores.asmx

4.2. Οι Υπηρεσίες αυτές και οι Ελληνικές Επιχειρήσεις.

Οι περισσότερες από τις Ελληνικές επιχειρήσεις που έχουν συνεχή παρουσία στο Internet είτε μέσω ενός δυναμικού website είτε μέσω εξειδικευμένων δικτυακών εφαρμογών μπορούν με σχετικά απλό τρόπο να δημιουργήσουν το δικό τους web service. Υπάρχει όμως λόγος κάθε επιχείρηση να δημιουργεί και ένα web service; Η απάντηση αυτή μπορεί να καλυφθεί μόνο μέσα από τις ανάγκες της ίδιας επιχείρησης.

Σίγουρα η δημιουργία ενός web service έχει υψηλό κόστος κατασκευής, πιθανών συντήρησης και αναβάθμισης, αφού είναι δουλειά για προγραμματιστές και όχι για απλούς χειριστές. Μια επιχείρηση που αποφασίζει να δημιουργήσει ένα δικό της web service μπορεί να στοχεύει στην απόλυτη παραμετροποίηση των εργασιών που ζητά από αυτό. Κάτι τέτοιο σημαίνει πως η επιχείρηση αυτή στοχεύει είτε στην πώληση του web service σε άλλες επιχειρήσεις, είτε στη δημοσιοποίηση και διαφήμιση των παρεχόμενων υπηρεσιών της μέσω αυτού. Επαφίεται λοιπόν στα χέρια κάθε επιχειρηματία να αποφασίσει αν πρέπει να επενδύσει χρόνο και χρήμα σε δημιουργία υπηρεσιών που θα παρέχονται μέσω Internet είτε δωρεάν είτε επί πληρωμή.

Από την άλλη μεριά, αν σκεφτεί κανείς το κόστος που μπορεί να έχει για μία εταιρεία η δημιουργία από το μηδέν ενός συστήματος που θα καλύπτει κάποιες ανάγκες, σε σχέση με την επιλογή χρήσης έτοιμης web service από το Internet, μπορεί εύκολα να δικαιολογήσει γιατί μέρα με τη μέρα η λίστα τέτοιων διαθέσιμων υπηρεσιών συνεχώς μεγαλώνει.

Σε δύο απλά βήματα μπορεί να συνοψιστεί η χρησιμοποίηση ενός έτοιμου web service προς όφελός μας:

- Αναζητούμε και βρίσκουμε εύκολα το web service που καλύπτει τις ανάγκες μας.
- Ζητάμε από τον προγραμματιστή να αναπροσαρμόσει τη ιστοσελίδα μας ώστε να εμφανίζεται η υπηρεσία που επιλέξατε στο σημείο της αρεσκείας μας και να εξυπηρετεί τον σκοπό της.

Η παραπάνω διαδικασία είναι εύκολη και σύντομη. Εμείς απλώς επιλέγουμε τι θέλουμε να προσφέρουμε από την ιστοσελίδα μας. Η λογική της δημιουργίας των πάντων από το μηδέν είναι επιχειρηματικά πλέον ξεπερασμένη. Το κόστος δημιουργίας και συντήρησης μίας υπηρεσίας συνήθως είναι πολύ πιο μεγάλο από το κόστος αγοράς ή ενοικίασης της ίδιας υπηρεσίας από κάποιους τρίτους.

Για παράδειγμα πριν πολλά – πολλά χρόνια, το πολύ είκοσι, όταν μια εταιρία αισθανόταν την ανάγκη να αποκτήσει Διαδικτυακό όνομα (domain name) και κυρίως δικό της εταιρικό e-mail (domain e-mail) ήταν υποχρεωμένη να καταθέσει σεβαστά κεφάλαια σε επενδύσεις. Απαιτούνταν δικό της υλικό εξοπλισμού (Server, UPS, firewalls, κ.λ.π.), λογισμικό (Λειτουργικό σύστημα του διακομιστή της, προγράμματα υποστήριξης των υπηρεσιών π.χ. Mailing system, κλπ) Επίσης απαιτούνταν ανάλογο προσωπικό μιας και τέτοιες επενδύσεις απαιτούν εξειδίκευση για την σωστή χρησιμοποίησή τους.

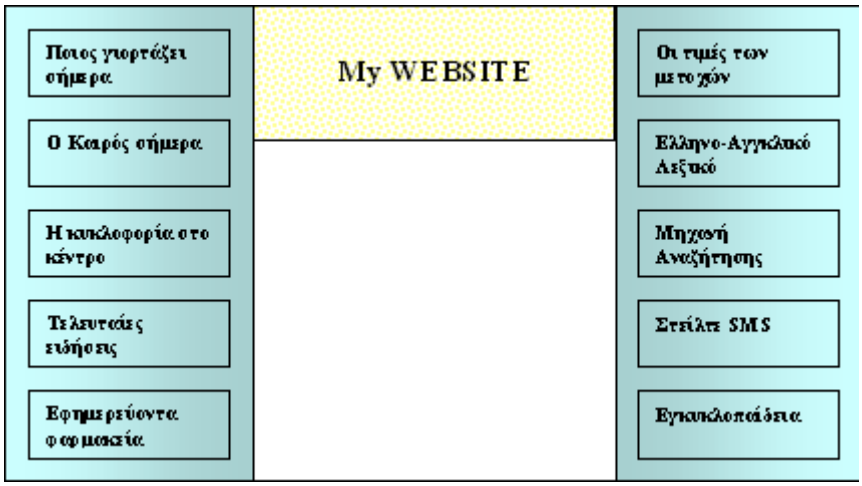
Σήμερα παρόμοιες υπηρεσίες (απλή απόκτηση domain και εταιρικό mail) μέσω web services μπορούν να μεταφερθούν στο επονομαζόμενο cloud, όπου έναντι μικρής ετήσιας αμοιβής μια εταιρία αποκτά όλα αυτά χωρίς καθόλου ρίσκο και χωρίς επενδύσεις για ασφάλεια δικτύων, συντήρηση και διαφύλαξη δεδομένων από καταστροφή.

Σήμερα για στην κατασκευή ενός δυναμικού website, η δυνατότητα ενσωμάτωσης μέσα σε αυτό επιπλέον πληροφοριών, χρήσιμων για τους επισκέπτες, είναι πλέον πολύ μεγαλύτερη και πιο εύκολη από ποτέ. Αυτές οι επιπλέον πληροφορίες μπορεί να είναι:

- Ο καιρός σήμερα
- Οι τιμές των μετοχών

- Η κυκλοφορία στο κέντρο της Αθήνας
- Εφημερεύοντα φαρμακεία
- Ποιος γιορτάζει σήμερα, κ.λ.π

Αλήθεια ένα website το οποίο να περιέχει τις παραπάτω υπηρεσίες πόσο δύσκολο τελικά είναι να υλοποιηθεί;



Εικόνα 5: Σχέδιο που απεικονίζει το βασικό layout ενός website με τρεις στήλες. Η μεσαία στήλη πιθανό να περιέχει υλικό που παρέχει ο ιδιοκτήτης του website, ενώ οι άλλες δύο, στα αριστερά και δεξιά, παρέχουν πληροφορίες τρίτων μέσω ενσωματωμένων web services.

Κάποιοι ίσως να σκέπτονται πως το να δημιουργήσουν ένα μεγάλο portal στο web από την αρχή, είναι τρομακτικό και μόνο σαν ιδέα. Κάποιοι άλλοι μπορεί να αναρωτιούνται πως θα μπορέσουν να ενσωματώσουν κάθε ημέρα τον καιρό ή τις πτήσεις των αεροπλάνων. Η απάντηση στην πρόκληση για τη δημιουργία ενός τέτοιου portal είναι καταφατική.

Αυτό που πρέπει να κάνει εδώ ο κατασκευαστής του website/portal είναι να ψάξει και να βρει στο Internet web services δωρεάν ή και επί πληρωμή που προσφέρουν αυτές τις πληροφορίες που χρειάζεται. Κατόπιν θα πρέπει να συμπληρώσει για κάθε κουτάκι (βλέπε εικόνα 5) στην σελίδα του κάποιες γραμμές εντολών οι οποίες επικοινωνούν με τον αντίστοιχο SOAP server. Αυτές οι γραμμές κώδικα παίρνουν τις πολύτιμες πληροφορίες και τις εμφανίζουν με όμορφο τρόπο. Το αποτέλεσμα θα είναι η δημιουργία ενός website με υπηρεσίες οι οποίες φαίνονται

στους χρήστες σαν να λειτουργούν στο ίδιο το site.

Έτσι τελικά κατασκευάζουμε μία ιστοσελίδα η οποία στην ουσία δημιουργείται τμηματικά από κλήσεις σε ξένες υπηρεσίες. Ο τελικός χρήστης θα βλέπει ένα σύνθετο και δυνατό αποτέλεσμα ενώ εμείς θα έχουμε ξοδέψει ελάχιστο χρόνο και χρήμα κάνοντας ταυτόχρονα την ιστοσελίδα μας πιο ελκυστική.

Είναι βέβαιο πως και στην Ελλάδα η λογική της παροχής on-line ηλεκτρονικών υπηρεσιών με την μορφή των web services θα ακολουθήσει την πορεία των υπόλοιπων χωρών του κόσμου. Οι Έλληνες επιχειρηματίες πλέον θα πρέπει να βλέπουν το Internet όχι μόνο ως ένα μέσο προώθησης και διαφήμισης των υπηρεσιών τους αλλά και σαν δυναμικό χώρο εκμετάλλευσης των υπαρχόντων υπηρεσιών τους και ανεύρεσης πελατών.

Ακόμα και στο επίπεδο της κατασκευής ενός website μίας επιχείρησης, πρέπει να υπάρχει πάντα μέσα στους κατασκευαστές το «μικρόβιο» της δημιουργίας κάποιας υπηρεσίας ή και της χρήσης κάποιας έτοιμης.

Για παράδειγμα όλες οι ξενοδοχειακές επιχειρήσεις θα επιθυμούσαν να αποκτήσουν on-line υπηρεσία κράτησης δωματίων. Αν αυτή η υπηρεσία βρίσκεται μόνο στο website της επιχείρησης τότε η επιχείρηση μειώνει δραματικά τον αριθμό των πιθανών κρατήσεων μέσω των δικών της επισκεπτών.

Σε αντίθεση αν παράλληλα με το website ο κατασκευαστής δημιουργήσει και ένα web service με αυτή την υπηρεσία τότε δίνει το δικαίωμα σε όλα τα μεγάλα τουριστικά - ενημερωτικά sites αλλά και σε συνεργαζόμενες επιχειρήσεις όπως τουριστικά γραφεία να κλείνουν αυτόματα δωμάτια στο κατάλυμα του.

Επειδή λοιπόν όλες οι επενδύσεις γίνονται με σκοπό το άμεσο ή έμμεσο κέρδος, κάθε επιχειρηματίας μπορεί εύκολα να καταλάβει πως η δημιουργία και η χρήση των web services στην ηλεκτρονική του επιχείρηση έχει τόσο άμεσα όσο και έμμεσα οφέλη:

- Χρησιμοποιώντας web services μειώνεται δραματικά ο χρόνος και το κόστος υλοποίησης των websites και γενικότερα των δικτυακών λογισμικών εφαρμογών. Ακόμα οι εφαρμογές γίνονται πιο χρηστικές και πιο ελκυστικές στους χρήστες.

- Δημιουργώντας web services αυξάνεται ραγδαία ο αριθμός των πιθανών πελατών αφού παρέχονται οι υπηρεσίες της επιχείρησης σε όλο τον δικτυακό κόσμο και μπορούν να χρησιμοποιηθούν με πολύ απλό τρόπο από άλλους προς όφελος όλων.

Σε μία εποχή που προωθούνται οι επιχειρηματικές συνεργασίες και το άνοιγμα της αγοράς μέσα από το Internet, οι Ελληνικές εταιρείες δεν θα πρέπει να αρκεστούν σε ένα μικρό βήμα προς τα εκεί. Όσο οι σχεδιασμοί των λογισμικών συστημάτων λαμβάνουν χώρα πάνω στο Internet και όσο το μέλλον της επιχειρηματικής Ελλάδας στρέφεται στον παγκόσμιο ιστό, τα web services πρέπει να αποτελούν βασικά στοιχεία στον σχεδιασμό των προγραμματιστών και απαίτηση από επιχειρηματίες και επενδυτές.

5. Συνοπτικά τεχνικά στοιχεία κατασκευής μιας Web Service

Το Web service είναι ένα λογισμικό σύστημα που αναγνωρίζεται από ένα URI [IETF RFC 2396] και που το περιβάλλον διεπαφής (interface) του, καθώς και οι δράσεις του, ορίζονται πλήρως και περιγράφονται σε eXtensble Markup Language (XML) μορφή.

Το web service μπορεί να βρεθεί και να χρησιμοποιηθεί εύκολα από άλλα λογισμικά συστήματα. Αυτά τα συστήματα μπορούν να αλληλεπιδρούν με το web service χρησιμοποιώντας υποχρεωτικά τρόπους επικοινωνίας μέσω του Internet και ανταλλάσσοντας πληροφορίες σε μορφή XML.

Παρόλο που ο ορισμός φαίνεται αρκετά γενικός και δεν περιορίζει τη χρήση των web services με συγκεκριμένα πρωτόκολλα, μετά από μία κοινή αποδοχή των μεγαλύτερων εταιρειών λογισμικού[8] στον κόσμο, αυτός έγινε πιο σαφής. Η αρχιτεκτονική αυτή έχει πλέον καθοριστεί σε ένα συγκεκριμένο μοντέλο, σύμφωνα με το οποίο θα πρέπει οι εταιρείες να παράγουν και να χρησιμοποιούν τα web services.

<u>Το πιο συνηθισμένο μοντέλο για τα web services προδιαγράφεται ως εξής:</u>

- Για την επικοινωνία χρησιμοποιείται συνήθως το πρωτόκολλο HTTP, το ίδιο δηλαδή πρωτόκολλο που χρησιμοποιούν και οι κοινοί browsers για την πλοήγηση στο Internet. Τα δεδομένα μιας Web Service μεταφέρονται όπως ακριβώς μεταφέρονται και οι ιστοσελίδες.

- Βασιζόμενο πάνω στο HTTP πρωτόκολλο (όχι αναγκαστικά) χρησιμοποιείται ένα άλλο πρωτόκολλο που ονομάζεται SOAP (Simple Object Access Protocol). Με τη χρήση του πρωτοκόλλου αυτού υλοποιείται ένας εξυπηρετητής (server), ο SOAP server.

8 Microsoft, IBM, Sun κ.α.

- Για κάθε μέθοδο της υπηρεσίας που θέλουμε να προσφέρουμε, ορίζουμε μία αντιστοιχία με μία λειτουργία του SOAP Server. Κατόπιν, το σύνολο των λειτουργιών του SOAP Server καθώς και τα υπόλοιπα χαρακτηριστικά του, περιγράφονται σε ένα αρχείο που ονομάζεται WSDL (Web Service Discription Language). Το αρχείο αυτό δημοσιεύεται στο Internet έτσι ώστε να δίνεται η δυνατότητα σε κάθε ενδιαφερόμενο χρήστη να μπορεί άμεσα να χρησιμοποιήσει την υπηρεσία.

Το SOAP (Simple Object Access Protocol) είναι επίσης άλλο ένα standard της W3C [W3C Simple Object Access Protocol (SOAP)] και χρησιμοποιείται αρκετά με στόχο την αποστολή απλών αντικειμένων (αρχείων, εφαρμογών, κλπ.) σε XML μορφή. Γι' αυτό και κάθε web service που χρησιμοποιεί SOAP μπορεί να λάβει αιτήσεις για συγκεκριμένες λειτουργίες, απλά δεχόμενο αντικείμενα σε XML. Η χρήση του SOAP πρωτοκόλλου γίνεται συνήθως πάνω από το πρωτόκολλο HTTP αλλά μπορεί να λειτουργήσει και με άλλα πρωτόκολλα π.χ. FTP, SMTP κ.α.

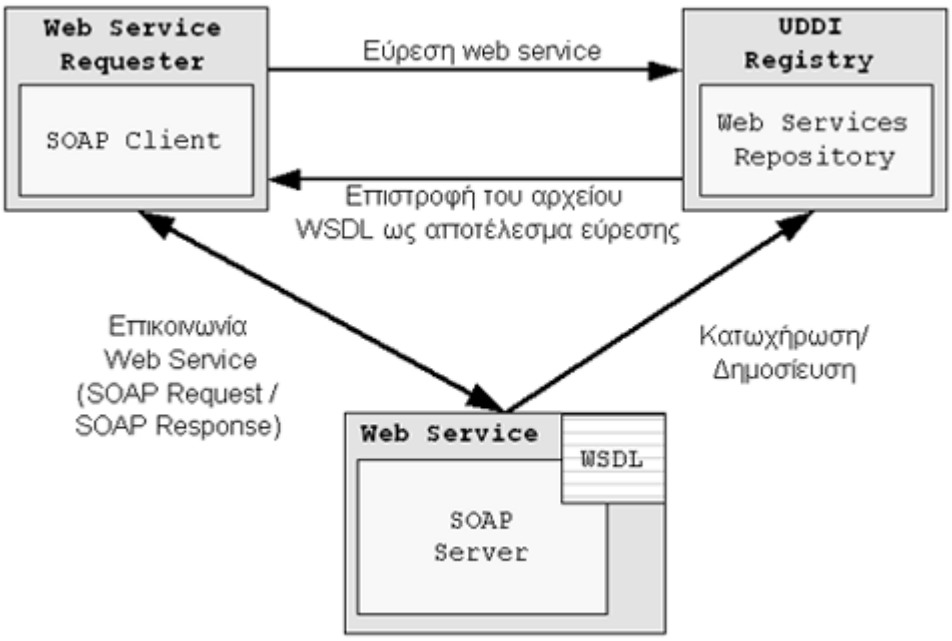
Πλέον, οι περισσότερες γλώσσες προγραμματισμού από την Delphi v7 μέχρι το Visual Studio .net, την PHP, την JSP και άλλες πολλές, υποστηρίζουν τη δημιουργία SOAP Servers με πάρα πολύ απλό τρόπο. Το μόνο που πρέπει να κάνει κανείς είναι να καθορίσει τις λειτουργίες που πρέπει να γίνουν όταν ο server δεχθεί κάποιο αίτημα προς εξυπηρέτηση.

Η WSDL (Web Service Description Language) [W3C Web Services Description Language (WSDL)] είναι μία γλώσσα σε XML μορφή η οποία περιγράφει απόλυτα ένα web service. Έτσι, για κάθε ένα web service που δημιουργείται, αντίστοιχα πρέπει να δημιουργείται ένα αρχείο WSDL, στο οποίο θα καταγράφονται όλες οι πληροφορίες για το ίδιο το service.

Πιο συγκεκριμένα εκεί καταγράφεται το πού βρίσκεται ο server (σε πια διεύθυνση), ποιες λειτουργίες υποστηρίζει, καθώς και πως δέχεται, πως επιστρέφει τα δεδομένα για κάθε λειτουργία. Παραδείγματα αρχείων WSDL μπορούν να βρεθούν σε κάθε διεύθυνση που αναφέρεται σε web service. Π.χ. http://www.capescience.com/webservices

Οι προγραμματιστές και εδώ δεν θα πρέπει να ανησυχούν πολύ αφού υπάρχουν πάρα πολλά

διαθέσιμα εργαλεία που δημιουργούν αυτόματα ένα wsdl αρχείο παράλληλα με τη δημιουργία του SOAP server.



Εικόνα 6: Βασικό διάγραμμα ροής της λειτουργίας του web service. Πηγή wikipedia.org

Τέλος το UDDI (Universal Description, Discovery, and Integration) αποτελεί ένα πρωτόκολλο καταχώρησης των web services. Χρησιμοποιείται για να μπορούμε να παρέχουμε ή να λαμβάνουμε πληροφορίες εύρεσης για τα web services και τους κατασκευαστές τους.

Κάθε καταχώρηση περιέχει το wsdl αρχείο και τη διεύθυνση που λειτουργεί η υπηρεσία στο Internet. Επιπρόσθετα σε κάθε καταχώρηση υπάρχουν και διάφορες άλλες πληροφορίες για την υπηρεσία που σχετίζονται με τον ιδιοκτήτη της, την πολιτική του κ.α.

Υπάρχουν διαφορετικοί τύποι καταχωρήσεων μίας υπηρεσίας. Πιο συγκεκριμένα υπάρχουν

καταχωρήσεις που μπορούν να γίνουν για υπηρεσίες από όλο τον κόσμο και που απευθύνονται σε όλο τον κόσμο, αλλά και καταχωρίσεις που απευθύνονται μόνο σε εξειδικευμένες επιχειρήσεις προωθώντας έτσι και το B2B (Business to Business) μοντέλο συνεργασίας. Τέλος υπάρχουν και καταχωρήσεις υπηρεσιών για πιο εξειδικευμένες περιπτώσεις.

Συνεπώς μία επιχείρηση γνωρίζοντας το UDDI μπορεί με πολύ απλό τρόπο να αναζητήσει και να βρει το αρχείο wsdl για κάποια συγκεκριμένη υπηρεσία και κατόπιν να την εκμεταλλευτεί εξίσου απλά και γρήγορα.

Υπάρχουν αρκετές διαφορετικές πλατφόρμες στις οποίες μπορεί να βασιστεί κανείς για τη δημιουργία ενός web service. Από τη μεριά της Microsoft, οι έτοιμες, δομημένες και εύκολες λύσεις που δίνει το περιβάλλον Visual Studio .NET, έχουν προσελκύσει πολλούς προγραμματιστές για να δημιουργούν τέτοιες υπηρεσίες σε αυτό.

Επίσης άλλες μεγάλες εταιρίες όπως η IBM και η ORACLE χρησιμοποιούν τα δικά τους προγραμματιστικά εργαλεία. Τέλος ακόμα και οι περισσότερες γλώσσες προγραμματισμού έχουν ενσωματώσει στις δυνατότητές τους την αυτόματη δημιουργία SOAP servers και την υποστήριξη των web services.

Μία δωρεάν λύση προσφέρεται και στους προγραμματιστές δυναμικών ιστοσελίδων που χρησιμοποιούν την γλώσσα PHP. Υπάρχουν έτοιμες βιβλιοθήκες (π.χ. nusoap) που μπορούν να χρησιμοποιηθούν από οποιονδήποτε για να δημιουργήσει απλά, προσθέτοντας μόνο 5 γραμμές εντολών, ένα web service.

6. Ένα απλό παράδειγμα υλοποίησης web service σε γλώσσα PHP

Σε αυτό το κεφάλαιο θα προσπαθήσουμε να δημιουργήσουμε ένα δικό μας λογισμικό, μια δικιά μας web service που να βασίζεται σε μια πραγματική ανάγκη για παροχή μιας ηλεκτρονικής υπηρεσίας μέσω του Ιστού.

Ας υποθέσουμε πως η εκκλησία της Ελλάδος επιθυμεί να υλοποιήσει μια υπηρεσία η οποία θα παρέχει υπηρεσίες εορτολογίου. Δηλαδή για κάθε ημερομηνία που θα δέχεται ή για την καθ' έκαστη, θα επιστρέφει τα ονόματα των Αγίων που εορτάζουν εκείνη την ημέρα. Είναι μια πρωτότυπη ιδέα που αφορά σε λογισμικό προσανατολισμένο στην στενή Ελληνική γεωγραφική επικράτεια.

6.1. Σχεδιάγραμμα δημιουργίας του Web Service και κώδικας

Πρώτα απ όλα πρέπει να δημιουργήσουμε μία απλή βάση δεδομένων με τις ημερομηνίες και τις αντίστοιχες εορτές. Δεν θα χρησιμοποιήσουμε κάτι έτοιμο και γι' αυτό η ακρίβεια και η πληθώρα των καταχωρημένων εορτών είναι σχετικά μικρή[9]. Κατόπι θα πρέπει να υλοποιήσουμε την υπηρεσία και τα κομμάτια αυτής που θα επικοινωνούν μεταξύ τους. Η διαδικασία της υλοποίησης περιγράφεται περιληπτικά παρακάτω:

- Δημιουργούμε τη διαδικασία ερώτησης στην βάση και επιστροφής αποτελεσμάτων. Αυτό γίνεται με μία συνάρτηση σε PHP π.χ. evresi_eortis(date) σε ένα αρχείο eortes.php
- Δημιουργούμε τον SOAP server. Χρησιμοποιώντας την έτοιμη βιβλιοθήκη nusoap[10], με 5 εντολές δημιουργούμε τον server μας σε ένα αρχείο π.χ. eortes.php
- Ορίζουμε στον server τη λειτουργία που επιθυμούμε να λαμβάνει χώρα, π.χ. ορίζουμε τη λειτουργία vres_giorti η οποία αντιστοιχεί στη συνάρτηση evresi_eortis.
- Δημιουργούμε το wsdl αρχείο με τις πληροφορίες για τον server μας, την λειτουργία τους και τα δεδομένα που δέχεται και επιστρέφει.

9 Η πηγή μας είναι το απλό συναξάρι που μπορούμε να προμηθευτούμε από οποιονδήποτε ναό της εκκλησίας της Ελλάδος.

10 Είναι ένα σετ από php κλάσεις που επιτρέπει, χωρίς την απαίτηση περαιτέρω επεκτάσεων, την δημιουργία υπηρεσιών web βασισμένων σε SOAP 1.1, WSDL 1.1 και HTTP 1.0/1.1.

Αρχείο eortes.php

```
// evresi eortis
function evresi_eortis($Date) {
  $query = "select onomata from imerologio where hmerominia = ". $Date;

  if (mysql_connect("localhost", "username", "passwd"))
  else { $error = "Database connection error";
    return $error;
  }
  if (mysql_select_db("imerologio"))
  else { $error = "Database not found";
    return $error;
  }
  if ($result = mysql_query($query))
  else {
    $error = "mysql_error()";
    return $error;
  }
  $onomata = mysql_result($result, 0, 0);
  return $onomata;
}
```

```
// SOAP – SERVER

require_once('nusoap.php');

$server = new soap_server;

$server->register('evresi_eortis');

$server->service($HTTP_RAW_POST_DATA);
```

Σε αυτό το σημείο έχουμε δημιουργήσει ένα web service το οποίο εκτελεί μόνο μία λειτουργία και μπορεί οποιοσδήποτε να το προσπελάσει στη διεύθυνση που έχουμε αποθηκεύσει το αρχείο eortes.php.

Κάποιες επιχειρήσεις λοιπόν είναι πιθανό να αναζητούν μία τέτοια ηλεκτρονική υπηρεσία η οποία πληροφορεί κάθε μέρα σχετικά με τις ονομαστικές εορτές. Αν και το παράδειγμα δείχνει απλό και εύκολο στην ανάπτυξή του από έμπειρους προγραμματιστές, θα πρέπει να επισημάνουμε πως είναι πάρα πολλά τα websites που θα μπορούσαν να κάνουν χρήση αυτής της υπηρεσίας. Για παράδειγμα μεγάλες επιχειρήσεις που εμπορεύονται δώρα, λουλούδια ή γλυκά θα μπορούσαν να κάνουν χρήση μιας τέτοιας υπηρεσίας ώστε συνδυάζοντάς την με δικό τους λογισμικό να δημιουργούν ενημερώσεις για τους πελάτες τους σχετικά με επικείμενες καταστάσεις ζήτησης. Ακόμη και απλά εμπορικά καταστήματα θα μπορούσαν να ενημερώνουν εγγεγραμμένους πελάτες τους με e-mails ή άλλους τρόπους ώστε να προκαλέσουν το ενδιαφέρον τους για αγορές.

6.2. Ο τρόπος ενσωμάτωσης της υπηρεσίας από τρίτους.

Ας υποθέσουμε ότι ένα δικτυακό ανθοπωλείο αναζητά αυτή την πληροφορία για να την ενσωματώσει στην ηλεκτρονική σελίδα του. Γνωρίζοντας την ύπαρξη του συγκεκριμένου web service η διαδικασία ενσωμάτωσης είναι πάρα πολύ απλή. Με αυτό τον απλό τρόπο θα ενημερώνονται οι πελάτες του (επιχειρήσεις και ιδιώτες) και θα προβαίνουν στις ανάλογες αγορές.

Για να χρησιμοποιηθεί αυτή η υπηρεσία web από ένα website αρκεί να ενσωματωθούν

κάποια στοιχεία, δηλαδή να γίνουν τα εξής βήματα :

- Στις σελίδες που ενδιαφέρει να εμφανιστούν τα αποτελέσματα μιας web service μπορεί να χρησιμοποιηθεί οποιαδήποτε γλώσσα δυναμικού προγραμματισμού ιστοσελίδων όπως η PHP, η ASP και άλλες. Κατόπι να ενσωματωθεί απλά (πάλι με χρήση περίπου 5 εντολών) ένας SOAP client ο οποίος έχει ως καθήκον να παρουσιάσει τα αποτελέσματα τις απάντησης από την υπηρεσία.

- Ορίζουμε στον SOAP client την τοποθεσία που βρίσκεται το αρχείο wsdl της υπηρεσίας των εορτών και ο client αναλαμβάνει μόνος του την ενημέρωσή του μέσω της επικοινωνίας του με τον SOAP server.

- Τέλος καλούμε με μία εντολή τη λειτουργία vres_giorti με παράμετρο την σημερινή ημερομηνία και εμφανίζουμε την επιστρεφόμενη τιμή στο σημείο της σελίδας που επιθυμούμε.

Κλήση του web service

```
// SOAP – clientrequire_once('nusoap.php');
// ορισμός της ημέρας
$param = array('Date'=>'02/10/2011');
$serverpath ='http: //...../eortes.php';
$client = new soapclient($serverpath);
$onomata = $client->call('vres_giorti',$param,$namespace);
print "Σήμερα γιορτάζουν οι ".$onomata;
unset($client);
```

6.3. Η βάση δεδομένων που χρησιμοποιήσαμε.

Η μορφοποίηση του αρχείου της βάσης δεδομένων που εισάγει τα περιεχόμενά της, όπου

μπορούμε καθαρά να δούμε ποιες μέρες του χρόνου θα επιστρέψουν ένα θετικό αποτέλεσμα, είναι η εξής:

```
CREATE DATABASE `eortologio`;
USE eortologio;

DROP TABLE IF EXISTS `days`;
CREATE TABLE `days` (
  `day` int(2) NOT NULL default '0',
  `month` int(2) NOT NULL default '0',
  `text` varchar(255) NOT NULL default ''
) TYPE=MyISAM DEFAULT CHARSET=greek COLLATE=greek_general_ci;

INSERT INTO `days` VALUES (7, 1, 'Ιωάννης, Ιωάννα, Ιώ, Πρόδρομος');
INSERT INTO `days` VALUES (1, 1, 'Βασίλης, Βασιλική');
INSERT INTO `days` VALUES (6, 1, 'Φώτης, Φωτεινή, Θεοφάνης, Θεοφανία, Φανή');
INSERT INTO `days` VALUES (9, 1, 'Ευστράτιος, Ευστρατία');
INSERT INTO `days` VALUES (11, 1, 'Θεοδόσιος');
INSERT INTO `days` VALUES (12, 1, 'Τατιάνα');
INSERT INTO `days` VALUES (17, 1, 'Αντώνης, Αντωνία');
INSERT INTO `days` VALUES (18, 1, 'Θανάσης, Αθανασία');
INSERT INTO `days` VALUES (20, 1, 'Ευθύμιος, Ευθυμία');
INSERT INTO `days` VALUES (21, 1, 'Μάξιμος');
INSERT INTO `days` VALUES (22, 1, 'Τιμόθεος');
INSERT INTO `days` VALUES (24, 1, 'Ξένη');
INSERT INTO `days` VALUES (25, 1, 'Γρηγόρης, Γρηγορία');
```

```
INSERT INTO `days` VALUES (26, 1, 'Ξενοφώντας');
INSERT INTO `days` VALUES (1, 2, 'Τρύφωνας');
INSERT INTO `days` VALUES (2, 2, 'Ιορδάνης');
INSERT INTO `days` VALUES (4, 2, 'Ισίδωρος');
INSERT INTO `days` VALUES (5, 2, 'Αγαθή');
INSERT INTO `days` VALUES (7, 2, 'Παρθένα, Παρθένιος');
INSERT INTO `days` VALUES (8, 2, 'Ζαχαρίας, Ζάχος, Ζαχαρούλα');
INSERT INTO `days` VALUES (9, 2, 'Νικηφόρος');
INSERT INTO `days` VALUES (10, 2, 'Χαράλαμπος, Χαρίκλεια, Κλειώ');
INSERT INTO `days` VALUES (11, 2, 'Βλάσσης');
INSERT INTO `days` VALUES (12, 2, 'Μελέτης');
INSERT INTO `days` VALUES (14, 2, 'Βαλεντίνος, Βαλεντίνα');
INSERT INTO `days` VALUES (18, 2, 'Λέων');
INSERT INTO `days` VALUES (19, 2, 'Φιλοθέη');
INSERT INTO `days` VALUES (23, 2, 'Πολύκαρπος');
INSERT INTO `days` VALUES (26, 2, 'Πορφύριος');
INSERT INTO `days` VALUES (27, 2, 'Ασκληπιός');
INSERT INTO `days` VALUES (1, 3, 'Ευδοκία');
INSERT INTO `days` VALUES (2, 3, 'Θάλια');
INSERT INTO `days` VALUES (8, 3, 'Ερμής');
INSERT INTO `days` VALUES (17, 3, 'Αλέξιος, Αλεξία, Αλίκη');
INSERT INTO `days` VALUES (18, 3, 'Θεόδωρος');
INSERT INTO `days` VALUES (19, 3, 'Χρύσανθος, Χρυσάνθη');
INSERT INTO `days` VALUES (25, 3, 'Βαγγέλης, Ευαγγελία');
INSERT INTO `days` VALUES (6, 4, 'Ευτύχιος, Ευτυχία');
INSERT INTO `days` VALUES (14, 4, 'Ανθή');
```

```
INSERT INTO `days` VALUES (15, 4, 'Λεωνίδας');
INSERT INTO `days` VALUES (20, 4, 'Ζωή, Πηγή');
INSERT INTO `days` VALUES (22, 4, 'Ναθαναήλ');
INSERT INTO `days` VALUES (23, 4, 'Γιώργος, Γεωργία');
INSERT INTO `days` VALUES (24, 4, 'Ελισσάβετ, Λίζα');
INSERT INTO `days` VALUES (25, 4, 'Μάρκος');
INSERT INTO `days` VALUES (29, 4, 'Ιάσωνας');
INSERT INTO `days` VALUES (30, 4, 'Ιάκωβος');
INSERT INTO `days` VALUES (1, 5, 'Ιερεμίας');
INSERT INTO `days` VALUES (5, 5, 'Ειρήνη');
INSERT INTO `days` VALUES (8, 5, 'Αρσένιος, Θεολόγος');
INSERT INTO `days` VALUES (9, 5, 'Ησαΐας, Χριστόφορος');
INSERT INTO `days` VALUES (11, 5, 'Μεθόδιος');
INSERT INTO `days` VALUES (13, 5, 'Γλυκερία');
INSERT INTO `days` VALUES (18, 5, 'Ιουλία');
INSERT INTO `days` VALUES (21, 5, 'Κωνσταντίνος, Ελένη');
INSERT INTO `days` VALUES (29, 5, 'Θεοδοσία');
INSERT INTO `days` VALUES (4, 6, 'Μάρθα');
INSERT INTO `days` VALUES (5, 6, 'Δωροθέος, Δωροθέα, Απόλλωνας');
INSERT INTO `days` VALUES (8, 6, 'Καλλιόπη');
INSERT INTO `days` VALUES (11, 6, 'Βαρναβάς, Βαρθολομαίος');
INSERT INTO `days` VALUES (22, 6, 'Ευσέβιος');
INSERT INTO `days` VALUES (27, 6, 'Παντελής');
INSERT INTO `days` VALUES (29, 6, 'Παύλος, Πέτρος, Πετρούλα, Παυλίνα, Πωλίνα');
INSERT INTO `days` VALUES (30, 6, 'Απόστολος');
INSERT INTO `days` VALUES (1, 7, 'Αργύρης, Αργυρώ, Δαμιανός, Κοσμάς');
```

```
INSERT INTO `days` VALUES (7, 7, 'Κυριάκος, Κυριακή');

INSERT INTO `days` VALUES (8, 7, 'Θεόφιλος, Προκόπης');

INSERT INTO `days` VALUES (11, 7, 'Ευθημία, Όλγα');

INSERT INTO `days` VALUES (17, 7, 'Μαρίνος, Μαρίνα');

INSERT INTO `days` VALUES (18, 7, 'Αιμίλιος, Αιμιλία');

INSERT INTO `days` VALUES (20, 7, 'Ηλίας, Ηλιάνα');

INSERT INTO `days` VALUES (22, 7, 'Μαγδαληνή, Μαρκέλα, Άννα');

INSERT INTO `days` VALUES (26, 7, 'Παρασκευή, Παρασκευάς');

INSERT INTO `days` VALUES (27, 7, 'Παντελεήμων,  Παντελής, Λάκης');

INSERT INTO `days` VALUES (29, 7, 'Καλλίνικος');

INSERT INTO `days` VALUES (31, 7, 'Ιωσήφ');

INSERT INTO `days` VALUES (6, 8, 'Σωτήρης, Σωτηρία');

INSERT INTO `days` VALUES (8, 8, 'Αιμιλιανός, Τριαντάφυλλος');

INSERT INTO `days` VALUES (10, 8, 'Λαυρέντης');

INSERT INTO `days` VALUES (15, 8, 'Παναγιώτης, Παναγιώτα, Μαριάννα, Μαρία, Μάριος, Δέσποινα');

INSERT INTO `days` VALUES (20, 8, 'Σαμουήλ');

INSERT INTO `days` VALUES (25, 8, 'Τίτος, Πάστης');

INSERT INTO `days` VALUES (26, 8, 'Ναταλία');

INSERT INTO `days` VALUES (27, 8, 'Φανούριος, Φανουρία');

INSERT INTO `days` VALUES (30, 8, 'Αλέξανδρος, Αλεξάνδρα, Αλέκα, Αλέκος');

INSERT INTO `days` VALUES (1, 9, 'Συμεών');

INSERT INTO `days` VALUES (3, 9, 'Άνθιμος');

INSERT INTO `days` VALUES (5, 9, 'Ζαχαρίας');

INSERT INTO `days` VALUES (9, 9, 'Ιωακείμ');

INSERT INTO `days` VALUES (11, 9, 'Ευανθία, Εύα');
```

INSERT INTO `days` VALUES (14, 9, 'Σταύρος, Σταυρούλα');

INSERT INTO `days` VALUES (15, 9, 'Νικήτας');

INSERT INTO `days` VALUES (16, 9, 'Ευθημία, Έφη');

INSERT INTO `days` VALUES (17, 9, 'Αγάπη, Σοφία, Ελπίδα');

INSERT INTO `days` VALUES (18, 9, 'Αριάδνη');

INSERT INTO `days` VALUES (20, 9, 'Στάθης, Ευσταθία');

INSERT INTO `days` VALUES (23, 9, 'Ξανθιππη, Πολυξένη, Ξένια');

INSERT INTO `days` VALUES (24, 9, 'Θέκλα, Μυρτώ');

INSERT INTO `days` VALUES (25, 9, 'Φρόσω');

INSERT INTO `days` VALUES (2, 10, 'Κυπριανός');

INSERT INTO `days` VALUES (3, 10, 'Διονύσιος, Διονυσία');

INSERT INTO `days` VALUES (4, 10, 'Ιερόθεος, Φραγκίσκος');

INSERT INTO `days` VALUES (5, 10, 'Χαριτίνη');

INSERT INTO `days` VALUES (7, 10, 'Σέργιος');

INSERT INTO `days` VALUES (8, 10, 'Πελαγία, Πέλλα');

INSERT INTO `days` VALUES (10, 10, 'Ευλάμπιος, Ευλαμπία');

INSERT INTO `days` VALUES (15, 10, 'Λουκιανός');

INSERT INTO `days` VALUES (18, 10, 'Λουκάς');

INSERT INTO `days` VALUES (20, 10, 'Αρτέμιος, Γεράσιμος');

INSERT INTO `days` VALUES (21, 10, 'Σωκράτης');

INSERT INTO `days` VALUES (23, 10, 'Ιάκωβος');

INSERT INTO `days` VALUES (26, 10, 'Δήμητρα, Δημήτρης');

INSERT INTO `days` VALUES (27, 10, 'Νέστωρας');

INSERT INTO `days` VALUES (1, 11, 'Αργύρης, Αργυρώ, Δαμιανός, Κοσμάς');

INSERT INTO `days` VALUES (8, 11, 'Αγγελική, Άγγελος, Ταξιάρχης, Γαρβιήλ, Μιχάλης, Σεραφείμ, Σταμάτης, Σταματία');

```
INSERT INTO `days` VALUES (9, 11, 'Νεκτάριος, Νεκταρία');
INSERT INTO `days` VALUES (10, 11, 'Ορέστης');
INSERT INTO `days` VALUES (11, 11, 'Βίκτωρας, Βικτώρια, Μηνάς');
INSERT INTO `days` VALUES (13, 11, 'Χρυσόστομος');
INSERT INTO `days` VALUES (14, 11, 'Φίλιππος');
INSERT INTO `days` VALUES (16, 11, 'Ματθαίος');
INSERT INTO `days` VALUES (18, 11, 'Πλάτωνας');
INSERT INTO `days` VALUES (21, 11, 'Μαρία, Δέσποινα, Παναγιώτα');
INSERT INTO `days` VALUES (25, 11, 'Κατερίνα, Κάτια, Μερκούρης');
INSERT INTO `days` VALUES (26, 11, 'Στυλιαννός, Στέλιος, Στέλλα');
INSERT INTO `days` VALUES (30, 11, 'Ανδρέας, Ανδριαννή');
INSERT INTO `days` VALUES (4, 12, 'Βαρβάρα');
INSERT INTO `days` VALUES (5, 12, 'Σάββας, Διογένης');
INSERT INTO `days` VALUES (6, 12, 'Νίκος, Νικολέτα');
INSERT INTO `days` VALUES (7, 12, 'Αμβρόσιος');
INSERT INTO `days` VALUES (9, 12, 'Άννα');
INSERT INTO `days` VALUES (12, 12, 'Σπύρος, Σπυριδούλα');
INSERT INTO `days` VALUES (13, 12, 'Λουκία, Στράτος');
INSERT INTO `days` VALUES (15, 12, 'Λευτέρης, Ελευθερία');
INSERT INTO `days` VALUES (17, 12, 'Δανιήλ, Διονύσης, Διονυσία');
INSERT INTO `days` VALUES (19, 12, 'Αγλαΐα');
INSERT INTO `days` VALUES (20, 12, 'Ιγνάτιος');
INSERT INTO `days` VALUES (21, 12, 'Θεμιστοκλής');
INSERT INTO `days` VALUES (22, 12, 'Αναστασία');
INSERT INTO `days` VALUES (24, 12, 'Ευγένιος, Ευγενία');
INSERT INTO `days` VALUES (25, 12, 'Χρήστος, Χριστίνα, Χρύσα, Μανώλης');
```

INSERT INTO `days` VALUES (27, 12, 'Στέφανος, Στεφανία');

6.4. Τελικό αποτέλεσμα

Προσπαθήσαμε να βρούμε λύση στην ανάγκη μας να στεγάσουμε την υπηρεσία στο web για περιορισμένο χρονικό διάστημα. Τελικά καταφέραμε να το κάνουμε σε έναν υπό-φάκελο ενός φιλικού website. Φορτώσαμε τον κώδικα καθαρά για επίδειξη σε Apache web sever. Το αρχείο της βάσης δεδομένων εισήχθηκε σε έναν mySQL server με τον οποίο η εφαρμογή μας μπορεί να επικοινωνήσει. Παρακάμψαμε την υλοποίηση UDDI μιας και ήταν σχετικά πολύπλοκη και δεν θεωρήσαμε θεμιτό να καταχωρήσουμε σε καταλόγους εύρεσης υπηρεσιών δικτύου μια υπηρεσία που είναι προσωρινή στην διαθεσιμότητά της. Άλλωστε οι υλοποιήσεις UDDI έτσι κι αλλιώς δεν είναι πλέον εφαρμοστέες λόγω μη κοινής αποδοχής και μάλιστα όπως είπαμε τείνουν προς κατάργηση. Υλοποιημένο το παράδειγμά μας μπορείτε να το βρεθεί στο url: http://www.makdel.gr/eortologiophp/eortes.php

7. Συμπεράσματα και περαιτέρω ανάπτυξη των Web Services

Ενώ με την αρχιτεκτονική των web services αντιμετωπίστηκαν τα περισσότερα προβλήματα συμβατότητας και δια-λειτουργικότητας μεταξύ των δικτυακών λογισμικών συστημάτων, υπάρχουν ακόμα αρκετά ανοικτά θέματα που έχουν να κάνουν με την ασφάλεια, την ποιότητα επικοινωνίας και με τον αυτόματο τρόπο αναζήτησης και ενσωμάτωσης έτοιμων υπηρεσιών σε ένα λογισμικό σύστημα.

Γι' αυτό είναι αναγκαίο η τεχνολογία των web services να εξελίσσεται συνεχώς προτείνοντας και θεσπίζοντας νέα standards πάνω στον τρόπο ανάπτυξης και λειτουργίας των υπηρεσιών ώστε να συμβαδίζουν με τις νέες εξελίξεις και τα νέα πρότυπα Ιστού. Ιδιαίτερη έμφαση πρέπει να δοθεί σε αυτό που πολλοί επιστήμονες στο εξωτερικό ονομάζουν «semantic web»[11], δηλαδή στη σημασιολογική ερμηνεία όλων των κόμβων πληροφορίας του παγκόσμιου ιστού.

Με νέα έρευνα και ανάπτυξη θα καλύψουμε τις απαιτήσεις για την αυτόματη σημασιολογικά ερμηνεία μιας on-line υπηρεσίας από ένα σύστημα[12]. Η επιστήμη προχωρά σε νέες δομές αυτόματης ανακάλυψης και εκμετάλλευσης των web services. Τεχνολογίες όπως οι Rails[13] και Axis2[14] θα βελτιώσουν πολύ την λειτουργία των web services.

Όμως αυτή η τεχνολογική εξέλιξη που συμβαίνει στον κόσμο, δεν έχει βρει την ευρεία αποδοχή των Ελληνικών εταιρειών πράγμα που κάνει την χρήση των αμιγώς ελληνικών web services να σπανίζει. Για να επιβεβαιώσει κανείς αυτήν την παρατήρηση μπορεί απλά να δοκιμάσει να αναζητήσει κάποιο Ελληνικό web service ή έστω βιβλιογραφία στα Ελληνικά για αυτήν την αρχιτεκτονική και θα απογοητευτεί από το ελάχιστο πλήθος των αποτελεσμάτων.

Υπάρχει λοιπόν ένας λόγος παραπάνω για τις επιχειρήσεις στην χώρα μας να επενδύσουν

11 Πηγή: http://semanticweb.org/wiki/Main_Page
12 Παρουσιάστηκε στην πρώτη διεθνή συνεδρίαση για το Semantic Web: Ankolenkar, A., et al. DAML-S: Web Service Description for the Semantic Web. (ISWC), 2002.
13 Πηγή: Dave Thomas & David Heinemeier Hansso, Agile Web Development with Rails
14 Πηγή: http://www.jaxmag.com/itr/online_artikel/psecom,id,747,nodeid,147.html

προς αυτήν την κατεύθυνση. Με αυτό τον τρόπο δεν θα αποδείξουν μόνο ότι είναι τόσο ανταγωνιστικές και σύγχρονες όσο και τεχνολογικά ενήμερες, για τις πλατφόρμες παροχής υπηρεσιών στον κόσμο μέσω του web. Αλλά θα εκμεταλλευτούν τις μελλοντικές ευκαιρίες που θα προκύψουν και στην χώρα μας με την διαρκή αυξανόμενη διείσδυση του Ιστού και των τεχνολογιών Internet σε κάθε σπίτι και κάθε επιχείρηση.

Αναφορές / Βιβλιογραφία

- Chatterjee - Webber, Developing Enterprise Web Services, Prentice Hall PTR , N.Jersey 2004
- Brian Winston, Media Technology and Society, Rout ledge, N. York – London 1998
- Κώστας Τζωρτζάκης, Οργάνωση και Διοίκηση, Rosili, Αθήνα 1999
- # ^ "Web Services Glossary". W3C. February 11, 2004. http://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/. Retrieved 2011-04-22.
- # ^ "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.
- # ^ Benslimane, Djamal; Schahram Dustdar, and Amit Sheth (2008). "Services Mashups: The New Generation of Web Applications". IEEE Internet Computing, vol. 12, no. 5. Institute of Electrical and Electronics Engineers. pp. 13–15. http://dsonline.computer.org/portal/site/dsonline/menuitem.9ed3d9924aeb0dcd82ccc6716bbe36ec/index.jsp?&pName=dso_level1&path=dsonline/2008/09&file=w5gei.xml&xsl=article.xsl.
- # ^ "Mashup Dashboard". ProgrammableWeb.com. 2009. http://www.programmableweb.com/mashups.
- # ^ "Web Services Description Language (WSDL) Version 2.0 Part 2: Adjuncts". W3C. http://www.w3.org/TR/2007/REC-wsdl20-adjuncts-20070626/#_http_binding_default_rule_method.
- # ^ "Help - Creating bottom-up Web services". Eclipse. http://help.eclipse.org/help33/index.jsp?topic=/org.eclipse.jst.ws.doc.user/concepts/cwsbtmup.html. Retrieved 2011-04-22.
- # ^ "Help - Creating top-down Web services". Eclipse. http://help.eclipse.org/help33/index.jsp?topic=/org.eclipse.jst.ws.doc.user/concepts/cwstopdown.html. Retrieved 2011-04-22.
- # ^ Bray, Tim (October 28, 2004). "WS-Pagecount". TBray.org.

http://www.tbray.org/ongoing/When/200x/2004/09/21/WS-Research. Retrieved 2011-04-22.

- # ^ "Rethinking the Java SOAP Stack". HP. http://www.hpl.hp.com/techreports/2005/HPL-2005-83.html. Retrieved 2011-04-22.
- # ^ Gray, N. A. B. (2005). "Performance of Java Middleware - Java RMI, JAXRPC, and CORBA". University of Wollongong. pp. 31–39. http://ro.uow.edu.au/infopapers/676/. Retrieved January 11, 2011. "The results presented in this paper show that the nature of response data has a greater impact on relative performance than has been allowed for in most previous studies."
- "SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)". W3C. April 27, 2007. http://www.w3.org/TR/soap12-part1/#intro. Retrieved 2011-02-01.
- Exclusive .NET Developer's Journal "Indigo" Interview with Microsoft's Don Box
- IBM Datapower
- IBM Zurich XML Accelerator Engine
- "SOAP over Java Message Service 1.0". W3C. June 4, 2009. http://www.w3.org/TR/2009/CR-soapjms-20090604/. Retrieved 2011-02-01.
- "SOAP Version 1.2 Part 0: Primer". W3C. December 17, 2001. http://www.w3.org/TR/2001/WD-soap12-part0-20011217/#SMTP. Retrieved 2011-02-01.
- Olson, Mike; Ogbuji, Uche (July 3, 2002). "The Python Web services developer: Messaging technologies compared". IBM developerWorks. http://www-128.ibm.com/developerworks/library/ws-pyth9/. Retrieved 2011-02-01.
- "Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language". http://www.w3.org/TR/wsdl20/. Retrieved 2007-06-27.
- "Web Services Description Language (WSDL) Version 2.0 Part 2: Adjuncts". http://www.w3.org/TR/2007/REC-wsdl20-adjuncts-20070626/#_http_binding_default_rule_method.
- ^ "W3C"
- ^ [1] UDDI R.I.P
- ^ Microsoft, IBM, SAP To Discontinue UDDI Web Services Registry Effort @ SOA

WORLD MAGAZINE

- ^ [2] Message announcing closure of Technical Committee
- ^ [3]
- ^ Taylor, Ian J. From P2P to Web Services and Grids - Peers in a Client/Server World. Springer, 2005
- ^ Taylor, Ian J. From P2P to Web Services and Grids - Peers in a Client/Server World. Springer, 2005
- [Fremantle, P., Weerawarana, S., and Khalaf, R. Enterprise Services, Communications of the ACM, October 2002/Vol.45.No 10, pp.77-82
- Leonard Richardson & Sam Ruby, RESTful Web Services.
- Dave Thomas & David Heinemeier Hansso, Agile Web Development with Rails
- http://www.cognizant.com/InsightsWhitepapers/FutureofWebServices.pdf

Web Links

- Messaging Design Pattern and a distributed component/service model
- W3C Web Services Activity home page
- Web Services Architecture (W3C Working Group Note)
- Where to find Web Services on the Web: Investigating Web Services on the World Wide Web (2008)
- ↑ W3C® DOCUMENT LICENSE
- ↑ XML 1.0 Origin and Goals
- ↑ XML Applications and Initiatives
- ↑ Introduction to iWork Programming Guide. Mac OS X Reference Library.
- ↑ XML 1.0 Specification
- W3C SOAP page
- SOAP Version 1.2 specification
- SOAP Tutorial
- WSDL 1.0 Specification
- WSDL 1.1 Specification
- WSDL 2.0 Specification Part 0: Primer (Latest Version)
- WSDL 2.0 Specification Part 1: Core (Latest Version)
- WSDL 2.0 Specification Part 2: Adjuncts (Latest Version)
- Web Services Description Working Group
- XML protocol activity
- JSR-110: Java APIs for WSDL
- JSR 172: Java ME Web Services Specification
- Online WSDL Validator
- W3Schools WSDL 1.1 tutorial
- WSDL programmatic visualization with Linguine Maps

- SSDL - The SOAP Service Description Language
- WSDL Java Bindings for XMLBeans and JAXB.
- W3C Working Draft
- UDDI specifications
- OASIS UDDI Technical Committee
- UDDI XML.org community site
- UDDI Browser
- C++ Data Binding for UDDI
- Introduction to UDDI: Important Features and Concepts
- Web Service Deployment
- jUDDI (pronounced "Judy") is an open source Java implementation of the Universal Description, Discovery, and Integration (UDDI v3) specification for Web Services.