



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ**

**ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ**

**ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ**

**ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ ΚΑΙ ΔΙΚΤΥΩΝ**

**ΥΨΗΛΗΣ ΑΠΟΔΟΣΗΣ  
ΥΠΗΡΕΣΙΕΣ ΔΙΑΔΙΚΤΥΟΥ  
ΓΙΑ ΕΠΙΣΤΗΜΟΝΙΚΟΥΣ ΥΠΟΛΟΓΙΣΜΟΥΣ**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

της

**ΓΡΗΓΟΡΙΑΣ ΚΑΤΣΟΓΙΑΝΝΟΥ**

Βόλος, Απρίλιος 2010

Η σελίδα αυτή είναι σκόπιμα λευκή.



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ  
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ ΚΑΙ ΔΙΚΤΥΩΝ

# ΥΨΗΛΗΣ ΑΠΟΔΟΣΗΣ ΥΠΗΡΕΣΙΕΣ ΔΙΑΔΙΚΤΥΟΥ ΓΙΑ ΕΠΙΣΤΗΜΟΝΙΚΟΥΣ ΥΠΟΛΟΓΙΣΜΟΥΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

ΓΡΗΓΟΡΙΑΣ ΚΑΤΣΟΓΙΑΝΝΟΥ

**Επιβλέπων :** Μανόλης Βάβαλης  
Αναπληρωτής Καθηγητής Τ.Μ.Η.Υ.Τ.Δ

Εγκρίθηκε από τη διμελή εξεταστική επιτροπή τον Οκτώβριο 2010.

(Υπογραφή)

.....  
Ηλίας Χούστης  
Καθηγητής Τ.Μ.Η.Υ.Τ.Δ.

(Υπογραφή)

.....  
Μανόλης Βάβαλης  
Αναπληρωτής Καθηγητής Τ.Μ.Η.Υ.Τ.Δ.

Βόλος, Οκτώβριος 2010

*(Υπογραφή)*

.....

**ΓΡΗΓΟΡΙΑ ΚΑΤΣΟΓΙΑΝΝΟΥ**

Διπλωματούχος Μηχανικός Ηλεκτρονικών Υπολογιστών, Τηλεπικοινωνιών και Δικτύων  
Πανεπιστημίου Θεσσαλίας

© 2010 – All rights reserved

Η σελίδα αυτή είναι σκόπιμα λευκή.

## Πίνακας περιεχομένων

1	<b>Εισαγωγή</b> .....	<b>1</b>
1.1	Αντικείμενο διπλωματικής.....	2
1.2	Οργάνωση κειμένου.....	3
2	<b>Τεχνολογικό υπόβαθρο</b> .....	<b>4</b>
2.1	Web Services .....	4
2.1.1	<i>Τα επίπεδα τεχνολογιών των Web Services</i> .....	5
2.2	Η βιβλιοθήκη LAPACK .....	10
2.2.1	<i>Περιγραφή</i> .....	10
2.2.2	<i>Χρήση</i> .....	11
3	<b>Σχεδίαση και υλοποίηση υπηρεσιών</b> .....	<b>13</b>
3.1	Σχεδίαση διαδικασίας .....	13
3.2	Υλοποίηση διαδικασίας .....	14
4	<b>Electronic Business XML (ebXML)</b> .....	<b>21</b>
4.1	Περιγραφή αρχιτεκτονικής και λειτουργίας .....	21
4.2	Χαρακτηριστικά γνωρίσματα .....	24
4.3	EbXML Registry Information Model (eb - RIM).....	28
4.4	Παραδείγματα χρήσης .....	31
4.5	Σύγκριση με UDDI .....	32
5	<b>Graphical User Interface (GUI)</b> .....	<b>34</b>
5.1	Τεχνολογίες που χρησιμοποιήθηκαν.....	34
5.1.1	<i>Content Management System(CMS) / Wordpress</i> .....	34
5.1.2	<i>Hyper Text Markup Language (HTML)</i> .....	35
5.1.3	<i>Cascading Style Sheets (CSS)</i> .....	35
5.1.4	<i>JQuery</i> .....	35
5.2	Περιγραφή διεπαφής.....	36
5.2.1	<i>Απαιτήσεις</i> .....	36

5.2.2	Περιβάλλον χρήσης.....	40
5.3	Σενάριο χρήσης.....	45
6	<b>Σύνοψη και μελλοντικές προοπτικές.....</b>	<b>53</b>
7	<b>Αναφορές .....</b>	<b>55</b>

## Πίνακας εικόνων

### ΚΕΦΑΛΑΙΟ 2

Εικόνα 2 - 1 Η αρχιτεκτονική των web services .....	5
Εικόνα 2 - 2 Τα επίπεδα τεχνολογιών των Web Services .....	6

### ΚΕΦΑΛΑΙΟ 4

Εικόνα 4 - 1 Ο τρόπος λειτουργίας του eb-XML.....	23
Εικόνα 4 - 2 Τα χαρακτηριστικά του ebXML Registry με μια ματιά .....	24
Εικόνα 4 - 3 Ο κύκλος ζωής ενός registry object .....	27
Εικόνα 4 - 4 Μορφή του μοντέλου πληροφορίας του ebXML Registry .....	28
Εικόνα 4 - 5 Παράδειγμα Guideline .....	31

### ΚΕΦΑΛΑΙΟ 5

Εικόνα 5 - 1 Οι ρουτίνες πραγματικών αριθμών που υποστηρίζονται από τη διεπαφή.....	37
Εικόνα 5 - 2 Οι ρουτίνες μιγαδικών αριθμών που υποστηρίζονται από τη διεπαφή.....	38
Εικόνα 5 - 3 Η φόρμα επιλογής ρουτίνας.....	41
Εικόνα 5 - 4 Οι φόρμες της σελίδας SELECT .....	41
Εικόνα 5 - 5 Σελίδα USE – Φόρμα καθορισμού εισόδου.....	42
Εικόνα 5 - 6 Σελίδα USE – Φόρμα καθορισμού εξόδου.....	43
Εικόνα 5 - 7 Σελίδα USE – Φόρμα παρακολούθησης εκτέλεσης .....	43
Εικόνα 5 - 8 Σελίδα CALL.....	44
Εικόνα 5 - 9 Σελίδα Contribute .....	45
Εικόνα 5 - 10 Home Page.....	46
Εικόνα 5 - 11 Ο χρήστης έχει επιλέξει τη ρουτίνα DGTSV.....	46



Εικόνα 5 - 12	Ο χρήστης διαβάζει την περιγραφή της ρουτίνας στο popup παράθυρο .....	47
Εικόνα 5 - 13	Ο χρήστης διαβάζει την περιγραφή της επιλογής Cluster .....	48
Εικόνα 5 - 14	Ο χρήστης διαβάζει το αρχείο WSDL της υπηρεσίας .....	48
Εικόνα 5 - 15	Η φόρμα επιλογής εισόδου της σελίδας USE .....	49
Εικόνα 5 - 16	Η φόρμα συμπλήρωσης παραμέτρων για την κλήση της DGTSV .....	50
Εικόνα 5 - 17	Ο χρήστης συμπληρώνει τα πεδία των παραμέτρων .....	51
Εικόνα 5 - 18	Η υπηρεσία επιστρέφει το αποτέλεσμα .....	52

Ο όρος επιστημονικός υπολογισμός [1] αναφέρεται στο αντικείμενο της μελέτης και κατασκευής μαθηματικών μοντέλων και τεχνικών για την επίλυση επιστημονικών προβλημάτων με τη χρήση υπολογιστικών συστημάτων. Συγκεκριμένα, οι αναλυτές δημιουργούν προγραμματιστικές εφαρμογές με τη βοήθεια των οποίων εκτελούν αλγόριθμους επίλυσης αριθμητικών προβλημάτων και μελετούν τα αποτελέσματά τους. Κρίνοντας από το μέγεθος των αριθμητικών δεδομένων που δίνονται ως παράμετροι σε τέτοιες εφαρμογές, γίνεται αντιληπτό ότι απαιτείται πολύς χρόνος και δέσμευση μεγάλης υπολογιστικής μνήμης για την ολοκλήρωση τέτοιων πράξεων. Συνεπώς, κρίνεται απαραίτητη η υλοποίηση και παροχή υψηλής απόδοσης υπολογισμών (High Performane Computing) οι οποίοι χρησιμοποιούν computer clusters για την πραγματοποίηση τέτοιου είδους υπολογισμών.

Σημαντικό ρόλο στους επιστημονικούς υπολογισμούς διαδραματίζει το διαδίκτυο. Στους κόλπους του κυκλοφορεί ποικιλία εμπορικών βιβλιοθηκών αριθμητικής ανάλυσης (όπως η NAG (Numerical Algorithms Group) ) αλλά και βιβλιοθήκες λογισμικού που περιέχουν πηγαίους κώδικες. Οι περισσότεροι από τους τελευταίους είναι γραμμένοι σε γλώσσα Fortran εξαιτίας της ταχύτητας με την οποία εκτελεί υπολογισμούς. Χαρακτηριστικό παράδειγμα αποτελεί η κύρια βιβλιοθήκη NETLIB [2] της οποίας το repository διαθέτει ελεύθερο λογισμικό αλλά και έγγραφα επιστημονικού ενδιαφέροντος. Συγκεκριμένα, παρέχει τη βιβλιοθήκη της Lapack προσφέροντας παράλληλα εναλλακτικές υλοποιήσεις της όπως η CLapack, η Lapack++ και η ScaLapack. Πρόκειται για βιβλιοθήκη συναρτήσεων γραμμικής άλγεβρας που κατέχει περίοπτη θέση στην επιστημονική κοινότητα. Ενδεικτικός των παραπάνω είναι ο αριθμός<sup>1</sup> αιτήσεων για τη συγκεκριμένη βιβλιοθήκη, ο οποίος στις αρχές Οκτωβρίου του 2010 ξεπέρασε τα 80 εκατομμύρια.

---

<sup>1</sup> [http://www.netlib.org/master\\_counts2.html#lapack/single](http://www.netlib.org/master_counts2.html#lapack/single)

Επιπρόσθετα, η σύγχρονη τάση παροχής λογισμικού υπό μορφή υπηρεσίας (Software as a Service) τείνει να κυριαρχήσει στο διαδίκτυο. Επιφέροντας τη λεγόμενη «χρήση λογισμικού κατ'απαίτηση» (software on demand) στον προσωπικό υπολογιστή του χρήστη, τον ελευθερώνει από την κατανάλωση χρόνου, κόπου και υπολογιστικής μνήμης που θα έπρεπε να σπαταληθούν για την εγκατάσταση του επιθυμητού λογισμικού στο μηχάνημά του. Πλέον οι απλές εφαρμογές που χρησιμοποιούνταν για την εκτέλεση λειτουργιών έχουν μεταμορφωθεί σε διαδικτυακές εφαρμογές (web applications) που μπορούν να προσπελαστούν από οποιοδήποτε χρήστη σε οποιοδήποτε υπολογιστή διαθέτει σύνδεση στο διαδίκτυο. Οι τελευταίες έχουν σημειώσει σημαντική εξέλιξη τα τελευταία χρόνια δημιουργώντας τις λεγόμενες υπηρεσίες διαδικτύου (web services), δηλαδή τμήματα λογισμικού προσφερόμενα μέσω διαδικτύου που είναι ικανά να εκτελέσουν απλούς υπολογισμούς έως και πολύπλοκες δραστηριότητες ανεξαρτήτως πλατφόρμας χρήσης.

## 1.1 Αντικείμενο διπλωματικής

Μελετώντας τη βιβλιοθήκη της Lapack αντιληφθήκαμε ότι πρόκειται για μια χρήσιμη βιβλιοθήκη για την επίλυση αριθμητικών προβλημάτων της οποίας η εγκατάσταση και χρήση στο μηχάνημα του χρήστη ίσως αποδειχθεί δύσκολη ή και ακατόρθωτη σε περίπτωση που ο χρήστης δε διαθέτει κατάλληλες γνώσεις. Εξαιτίας της πολυπλοκότητας της παραπάνω διαδικασίας, θεωρήσαμε σκόπιμο να προσφέρουμε τη βιβλιοθήκη αυτή ως υπηρεσία διαδικτύου. Με αυτό τον τρόπο, παρακάμπτεται το βήμα της εγκατάστασης της στο μηχάνημα και πλέον ο χρήστης αρκεί να καλέσει την αντίστοιχη υπηρεσία διαδικτύου.

Για την υλοποίηση υψηλής απόδοσης υπολογισμών οι υπηρεσίες δεν προσφέρονται μέσω ενός απλού desktop personal computer αλλά μέσω ενός desktop cluster 48 πυρήνων όπου έχει εγκατασταθεί ο Glassfish Application Server. Για την ανακάλυψη και κλήση των προσφερόμενων υπηρεσιών απαιτείται η δήλωση και δημοσίευσή τους σε ένα registry. Μέχρι πρότινος, η συγκεκριμένη διαδικασία γινόταν μέσω του UDDI που είχε δημιουργηθεί για αυτό ακριβώς το σκοπό. Να εξυπηρετεί δηλαδή στην εγγραφή και δήλωση οργανισμών έτσι ώστε να μπορεί οποιαδήποτε επιχείρηση να ανακαλύψει τον κατάλληλο συνεργάτη και να επικοινωνήσει μαζί του. Στη συγκεκριμένη εργασία αντί του συγκεκριμένου registry χρησιμοποιούμε το ebXML, ένα framework ηλεκτρονικού εμπορίου με σύνθετα χαρακτηριστικά και περισσότερες δυνατότητες συγκριτικά με το UDDI.

Επίσης, για την αυτοματοποίηση της διαδικασίας και διευκόλυνση του χρήστη, σχεδιάσαμε και υλοποιήσαμε ένα γραφικό περιβάλλον όπου μπορεί ο κάθε χρήστης να επιλέξει την

κατάλληλη ρουτίνα μέσα από μία σειρά ρουτίνων επίλυσης γραμμικών συστημάτων και να την καλέσει.

Συνοψίζοντας στην παρούσα εργασία ασχοληθήκαμε με τα παρακάτω ζητήματα:

1. Περιγραφή της διαδικασίας για την κλήση Lapack ρουτίνων μέσω υπηρεσιών διαδικτύου
2. Μελέτη των χαρακτηριστικών του ebXML και χρήση του ως web service registry
3. Σχεδίαση και υλοποίηση γραφικού περιβάλλοντος για την κλήση υπηρεσιών

## 1.2 Οργάνωση κειμένου

Το υπόλοιπο της παρούσας εργασίας είναι οργανωμένο ως εξής:

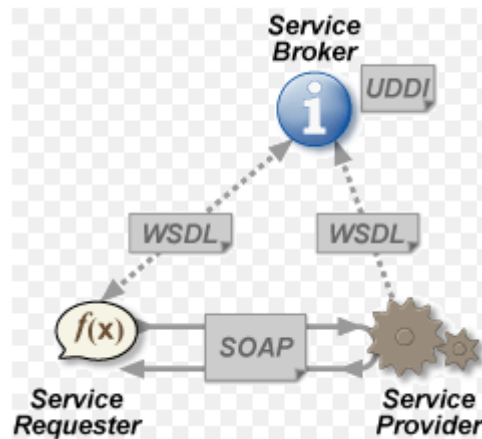
Στο 2<sup>ο</sup> κεφάλαιο περιγράφουμε τις τεχνολογίες που χρησιμοποιήσαμε, ενώ στο 3<sup>ο</sup> τη διαδικασία και την υλοποίηση μιας υπηρεσίας διαδικτύου. Στο 4<sup>ο</sup> κεφάλαιο παρουσιάζεται η μελέτη για το ebXML και τον τρόπο με τον οποίο μπορεί να χρησιμοποιηθεί στις υπηρεσίες διαδικτύου. Τέλος, στο 5<sup>ο</sup> κεφάλαιο παρουσιάζεται ο σχεδιασμός και η υλοποίηση ενός γραφικού περιβάλλοντος σχεδιασμένου για την κλήση των υπηρεσιών.

## 2.1 Web Services

Τα web services εμφανίστηκαν το 2001 όταν άρχισε να γίνεται λόγος για αυτο-περιγραφόμενες και αυτο-προσδιοριζόμενες εφαρμογές οι οποίες μπορούν να ανακαλυφθούν και να προσπελαστούν μέσω διαδικτύου από άλλες εφαρμογές. Η σχετική δραστηριότητα ξεκίνησε λίγο αργότερα, το 2002 όταν συστάθηκαν τα working groups στα οποία συμμετείχαν οι IBM, W3C (World Wide Web Consortium), Microsoft και Oracle [20]. Ουσιαστικά, πρόκειται για μία συλλογή open protocols και standards τα οποία εξελίσσονται συνεχώς χάρη στους οργανισμούς W3C και OASIS (Organization for the Advancement of Structured Information Systems).

Web service [3] είναι οποιοδήποτε τμήμα λογισμικού ή συστατικό εφαρμογής που είναι διαθέσιμο μέσω διαδικτύου και το οποίο μπορεί να ολοκληρώνει εργασίες και να λύνει προβλήματα. Στην αρχιτεκτονική του συμμετέχουν οι παρακάτω οντότητες:

- **Service provider ( ή publisher )** : Είναι ο πάροχος της υπηρεσίας διαδικτύου. Αυτός υλοποιεί την υπηρεσία και την κάνει διαθέσιμη στο διαδίκτυο.
- **Service requestor ( ή consumer )** : Είναι ο κάθε χρήστης της υπηρεσίας διαδικτύου, ο οποίος καλεί την υπάρχουσα υπηρεσία ανοίγοντας μία δικτυακή σύνδεση και στέλνοντας μία XML αίτηση.
- **Service broker ( ή service registry )** : Είναι ένας λογικός κεντρικός κατάλογος των υπηρεσιών. Εδώ παρέχεται ένα κεντρικό σημείο όπου οι δημιουργοί μπορούν να δημοσιεύουν τις νέες υπηρεσίες και οι χρήστες να εντοπίζουν τις ήδη υπάρχουσες.

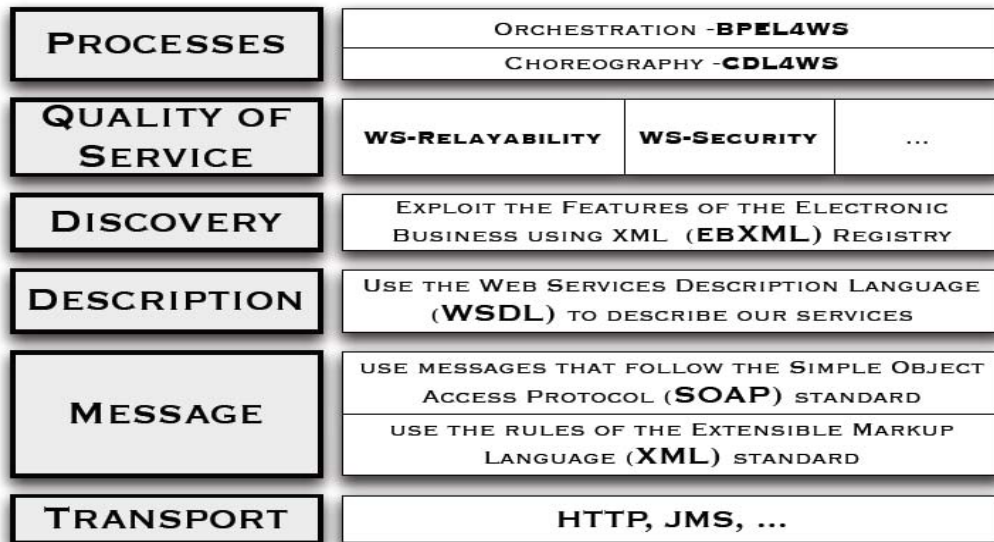


Εικόνα 2 - 1 Η αρχιτεκτονική των web services

Επίσης, τα web services αποτελούν συστήματα ανταλλαγής πληροφορίας μεταξύ των εφαρμογών μέσω του διαδικτύου και μπορούν να περιλαμβάνουν προγράμματα, αντικείμενα, μηνύματα ή έγγραφα. Μπορούν να χρησιμοποιηθούν από άλλες εφαρμογές και ανακαλύπτονται μέσω ενός απλού μηχανισμού αναζήτησης με ή χωρίς ανθρώπινη παρέμβαση επιτρέποντας στις εφαρμογές να επικοινωνούν μεταξύ τους, ανεξαρτήτως γλώσσας προγραμματισμού ή λειτουργικού συστήματος και παρέχοντας διαλειτουργικότητα και ευελιξία. Η αρχιτεκτονική τους είναι service- oriented (Service Oriented Architecture = SOA) καθώς επικεντρώνονται στην παροχή υπηρεσιών και παρέχουν τη δυνατότητα κατασκευής ενός περιβάλλοντος εργασίας μέσα από το συνδυασμό επαναχρησιμοποιούμενων υπηρεσιών οι οποίες θα ανταλλάσουν πληροφορία δημιουργώντας χαλαρά συνδεδεμένους δεσμούς. Αυτό συνεπάγεται ότι ο service requestor δε γνωρίζει τις τεχνικές λεπτομέρειες της υλοποίησης της υπηρεσίας από τον service provider, όπως τη γλώσσα προγραμματισμού. Συνεπώς, οι δύο πλευρές είναι ανεξάρτητες η μία από την άλλη και μπορούν να πραγματοποιήσουν αλλαγές χωρίς να επηρεαστεί η μεταξύ τους επικοινωνία εφόσον το σχήμα της επικοινωνίας τους παραμένει αμετάβλητο.

### 2.1.1 Τα επίπεδα τεχνολογιών των Web Services

Στη συνέχεια, θα αναφερθούμε στις τεχνολογίες οι οποίες χρησιμοποιούνται από τις web services για την ορθή επικοινωνία μεταξύ των εφαρμογών:



Εικόνα 2 - 2 Τα επίπεδα τεχνολογιών των Web Services

### **Transport Layer**

Πρόκειται για το επίπεδο που είναι υπεύθυνο για τη μεταφορά μηνυμάτων ανάμεσα στις εφαρμογές. Περιλαμβάνει πρωτόκολλα μεταφοράς όπως το HTTP (Hypertext Transfer Protocol) που είναι το πρωτόκολλο επιπέδου εφαρμογής του web και αποτελεί τον πυρήνα του και το JMS (Java message service) που είναι διεπαφή για την ανταλλαγή μηνυμάτων ανάμεσα σε clients.

### **Message Layer**

Πρόκειται για το επίπεδο που είναι υπεύθυνο για την ανταλλαγή μηνυμάτων των εφαρμογών. Εδώ περιλαμβάνεται η γλώσσα στην οποία βασίζονται όλα τα μηνύματα που ανταλλάσσονται XML (Extensible Markup Language) αλλά και το πρωτόκολλο επικοινωνίας SOAP (Simple Object Access Protocol) βάσει του οποίου ανταλλάσσονται τα μηνύματα. Αναλυτικά:

- Η XML είναι σχεδιασμένη για τη μεταφορά και αποθήκευση δεδομένων και χρησιμοποιείται για την κωδικοποίηση και αποκωδικοποίηση των δεδομένων που αποστέλλονται. Χρησιμοποιείται παγκοσμίως και παρέχει τη δυνατότητα λειτουργικότητας στις web services καθώς όλα τα μηνύματα που ανταλλάσσονται είναι βασισμένα στην XML.

- Το SOAP είναι ένα πρωτόκολλο επικοινωνίας βασισμένο στο XML, το οποίο επιτρέπει σε εφαρμογές να ανταλλάσσουν πληροφορία πάνω από πρωτόκολλα μεταφοράς όπως το HTTP και SMTP. Ειδικότερα, χρησιμοποιείται για τη μεταφορά δεδομένων ανάμεσα στον πάροχο της υπηρεσίας και το χρήστη, ορίζοντας ποια πληροφορία πρέπει να σταλεί και πώς. Είναι ανεξάρτητο γλώσσας και πλατφόρμας και μέσω αυτού μπορούν να ανταλλαχθούν ολόκληρα έγγραφα ή να κληθεί μια απομακρυσμένη διαδικασία (remote procedure). Το σημαντικό χαρακτηριστικό του που το κάνει να ξεχωρίζει σε σχέση με άλλα καταναμημένα πρωτόκολλα επικοινωνίας είναι πως ενισχύει τη διαλειτουργικότητα ανάμεσα σε πλατφόρμες και γλώσσες προγραμματισμού. Αντίθετα, το κύριο μειονέκτημά του σε σχέση με ανάλογες τεχνολογίες είναι πως εξαιτίας της χρήσης της XML, τα μηνύματα που δημιουργούνται είναι μεγάλα και η επικοινωνία πιο αργή. Παρόλ' αυτά, πρόκειται για ένα καθολικά αποδεκτό και απλό πρωτόκολλο καθώς βασίζεται στο XML το οποίο είναι δομημένο και εύκολο στην ανάλυση.

### **Description Layer**

Είναι το επίπεδο που ευθύνεται για την περιγραφή των web services και βασίζεται στη γλώσσα περιγραφής υπηρεσιών διαδικτύου WSDL (Web Services Description Language). Πρόκειται για μία γλώσσα βασισμένη στην XML η οποία περιγράφει τις web services και τον τρόπο προσέγγισής τους. Πρόκειται δηλαδή για ένα έγγραφο γραμμένο σε XML το οποίο περιγράφει μία υπηρεσία διαδικτύου καθορίζοντας τη θέση της και τις λειτουργίες που προσφέρονται από αυτή. Συγκεκριμένα, σε ένα έγγραφο WSDL υπάρχουν οι ακόλουθες πληροφορίες:

- Τα μηνύματα που ανταλλάσσονται και η δομή τους
- Οι λειτουργίες που υλοποιούνται και οι τύποι δεδομένων που περιλαμβάνονται σε αυτές
- Το πρωτόκολλο που χρησιμοποιείται
- Τη διεύθυνση για τον εντοπισμό της



## *Discovery Layer*

Πρόκειται για το επίπεδο που είναι υπεύθυνο για τη δημοσίευση των web services και την εγγραφή τους σε ένα κοινό registry όπου μπορούν να ανακαλυφθούν από τις υπόλοιπες υπηρεσίες. Μέχρι πρότινος το κύριο framework αυτού του επιπέδου ήταν το UDDI (Universal Description, Discovery and Integration). Πρόκειται για ένα framework[4], βασισμένο στην XML και ανεξάρτητο πλατφόρμας στα πλαίσια του οποίου περιγράφονται υπηρεσίες και ανακαλύπτονται οργανισμοί. Ουσιαστικά, λειτουργεί ως ευρετήριο υπηρεσιών στο οποίο εγγράφονται επιχειρήσεις και ανακαλύπτουν υπηρεσίες.

Δημιουργήθηκε το 2000 από την OASIS (Organization for the Advancement of Structured Information Standards) και μέχρι τα τέλη του 2005 την υποστήριζαν μεγάλοι οργανισμοί όπως Microsoft, IBM, Oracle, Dell και HP. Το 2006 η IBM και η Microsoft ανακοίνωσαν ότι κλείνουν τα δημόσια τους registries και τελικά το 2007 το UDDI ολοκλήρωσε τη λειτουργία του.

Όσον αφορά στη λειτουργία του, η διαδικασία είναι η ακόλουθη:

Ο πάροχος της υπηρεσίας δημοσιεύει το WSDL αρχείο στο UDDI, ενώ ο χρήστης της υπηρεσίας μέσω του πρωτοκόλλου SOAP ανταλλάσσει μηνύματα με τον πάροχο και προσπαθεί να συνδεθεί σε αυτή.

Η εγγραφή ενός οργανισμού στο UDDI περιλαμβάνει 3 βήματα:

- **White Pages** : παρέχονται πληροφορίες για τον οργανισμό που προσφέρει την υπηρεσία όπως το όνομα του οργανισμού, την περιγραφή του και στοιχεία επικοινωνίας.
- **Yellow Pages** : γίνεται ταξινόμηση της υπηρεσίας ή του οργανισμού με βάσει standard ταξινομήσεις.
- **Green Pages** : γίνεται περιγραφή της πρόσβασης σε μια υπηρεσία, όπως η διεύθυνση της υπηρεσίας και οι παράμετροί της.

Τα τελευταία χρόνια σημαντική εξέλιξη σε αυτό τον τομέα έχει σημειώσει ένα άλλο framework ηλεκτρονικού εμπορίου, το ebXML. Αναλυτική περιγραφή του και σύγκριση με το UDDI ακολουθούν στο Κεφάλαιο 4.

### **Quality of service Layer**

Πρόκειται για το επίπεδο που είναι υπεύθυνο για την ποιότητα των υπηρεσιών αλλά και της απόδοσης της επικοινωνίας. Περιλαμβάνει πρότυπα που αφορούν την αξιοπιστία και τη μέγιστη ασφάλεια των μηνυμάτων που ανταλλάσσονται όπως τα ws-reliability και ws-security που περιλαμβάνει υπογραφές και αλγόριθμους κρυπτογράφησης για την αυθεντικοποίηση της ταυτότητας. Για την τήρηση των παραπάνω, η επικοινωνία μεταξύ των υπηρεσιών προϋποθέτει τη σύναψη ενός είδος «συμβολαίου», του λεγόμενου Service Level Agreement (SLA). Σε αυτή τη συμφωνία αναφέρονται οι όροι που διέπουν τις συναλλαγές των 2 πλευρών και αφορούν τις υποχρεώσεις τους, τις εγγυήσεις συνεργασίας αλλά και απαραίτητες λεπτομέρειες για την παροχή υπηρεσιών.

### **Processes Layer**

Πρόκειται για το επίπεδο που είναι υπεύθυνο για το συνδυασμό ήδη υλοποιημένων υπηρεσιών ώστε να προκύψουν νέες σύνθετες υπηρεσίες. Επιτρέπει στους χρήστες να περιγράφουν αλλά και να υλοποιούν δραστηριότητες μέσω web services ορίζοντας τον τρόπο με τον οποίο θα συνεργαστούν για να πραγματοποιήσουν διάφορες λειτουργίες.

Εδώ περιλαμβάνονται δύο γλώσσες/πρότυπα:

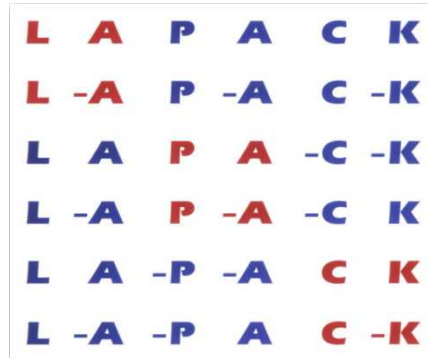
#### **BPEL (Business Process Execution Language)**

Πρόκειται για μία orchestration γλώσσα που δημιουργήθηκε από τον οργανισμό OASIS με στόχο τη σύνθεση νέων υπηρεσιών χρησιμοποιώντας ήδη υπάρχουσες, ορίζοντας τις ροές εργασίας μέσα στο πλαίσιο των υπηρεσιών, όπως για παράδειγμα αν θα εκτελεστούν σειριακά ή παράλληλα. Συγκεκριμένα, αποτελεί μία γλώσσα «ενορχήστρωσης» που καθορίζει το ρόλο της κάθε υπηρεσίας στη νέα σύνθεση.

#### **CDL (Choreography Description Language)**

Πρόκειται για μία choreography γλώσσα που δημιουργήθηκε από το W3C με στόχο τον καθορισμό της επικοινωνίας ανάμεσα στις υπηρεσίες. Συγκεκριμένα, καθορίζει τη συνεργασία ανάμεσα σε ομότιμες υπηρεσίες, όπως και την αναμενόμενη συμπεριφορά κατά την ανταλλαγή μηνυμάτων.

## 2.2 Η βιβλιοθήκη LAPACK



### 2.2.1 Περιγραφή

Η LAPACK (Linear Algebra Package) είναι μια βιβλιοθήκη λογισμικού για την αριθμητική γραμμική άλγεβρα. Κυκλοφόρησε το 1992 αρχικά γραμμένη σε Fortran 77, ενώ η τρέχουσα έκδοση είναι γραμμένη σε Fortran 90. Παρέχει ρουτίνες για την επίλυση συστημάτων γραμμικών εξισώσεων, προβλημάτων ελαχίστων τετραγώνων και προβλημάτων ιδιοτιμών, όπως επίσης για την εφαρμογή LU παραγοντοποίησης και ανάλυσης Cholesky.

Οι ρουτίνες της μπορούν να χειριστούν πραγματικούς και μιγαδικούς πίνακες τόσο απλής όσο και διπλής ακρίβειας.

Διακρίνονται στις εξής κατηγορίες [5]:

- Driver routines: κάθε μία από αυτές λύνει ένα ολόκληρο πρόβλημα, όπως την επίλυση ενός γραμμικού συστήματος.
- Computational routines: κάθε μία από αυτές ολοκληρώνει μία ξεχωριστή υπολογιστική δραστηριότητα, όπως μία LU παραγοντοποίηση. Καλούνται απευθείας για να συμβάλλουν στην ολοκλήρωση μιας δραστηριότητας την οποία δεν μπορούν να χειριστούν οι driver ρουτίνες.
- Utility routines: υλοποιούν μη αριθμητικές λειτουργίες όπως ο χειρισμός σφαλμάτων και η σύγκριση χαρακτήρων.
- Auxiliary routines: καλούνται από τις ρουτίνες της Lapack για υπολογισμούς χαμηλού επιπέδου, όπως ο υπολογισμός της νόρμας ενός πίνακα.
- Testing routines: χρησιμοποιούνται για την επαλήθευση των αριθμητικών αποτελεσμάτων των computational ρουτινών.
- Timing routines: χρησιμοποιούνται για την αξιολόγηση της απόδοσης των computational ρουτινών.

Το όνομα της κάθε ρουτίνας είναι ενδεικτικό των χαρακτηριστικών της. Αποτελείται από 5-6 χαρακτήρες ο καθένας από τους οποίους υποδηλώνει λεπτομέρειες για τα αριθμητικά μεγέθη. Συγκεκριμένα,

- Ο 1<sup>ος</sup> χαρακτήρας υποδηλώνει τον τύπο των αριθμητικών δεδομένων και μπορεί να έχει τις ακόλουθες τιμές :  
S για τους πραγματικούς αριθμούς απλής ακρίβειας  
D για τους πραγματικούς αριθμούς διπλής ακρίβειας  
C για τους μιγαδικούς αριθμούς απλής ακρίβειας  
Z για τους μιγαδικούς αριθμούς διπλής ακρίβειας
- Ο 2<sup>ος</sup> και ο 3<sup>ος</sup> χαρακτήρας υποδηλώνουν τον τύπο του πίνακα ή του πιο σημαντικού πίνακα αν υπάρχουν περισσότεροι από ένας, όπως στην περίπτωση της επίλυσης συστημάτων
- Οι υπόλοιποι χαρακτήρες υποδηλώνουν το είδος του υπολογισμού που εκτελείται

Για παράδειγμα, το όνομα της ρουτίνας DPOSV υποδηλώνει ότι πρόκειται για αριθμούς πραγματικούς διπλής ακρίβειας (D = Double), ο κύριος πίνακας είναι θετικά ορισμένος (PO = Positive) και ο υπολογισμός είναι η επίλυση συστήματος (SV = Solve).

Οι ρουτίνες της Lapack χρησιμοποιούν για την εκτέλεση των υπολογισμών τους τις ρουτίνες της BLAS (Basic Linear Algebra Subprograms). Πρόκειται για μία βιβλιοθήκη που δημιουργήθηκε το 1979 και παρέχει ρουτίνες για την εκτέλεση απλούστερων υπολογισμών, όπως ο πολλαπλασιασμός διανυσμάτων και πινάκων και ο υπολογισμός της νόρμας. Η λειτουργικότητά τους χωρίζεται σε 3 επίπεδα:

- Το 1<sup>ο</sup> επίπεδο περιλαμβάνει υπολογισμούς που αφορούν διανύσματα
- Το 2<sup>ο</sup> επίπεδο περιλαμβάνει υπολογισμούς που αφορούν διανύσμα και πίνακα
- Το 3<sup>ο</sup> επίπεδο περιλαμβάνει υπολογισμούς μεταξύ πινάκων

### **2.2.2 Χρήση**

Το πρώτο βήμα είναι η εγκατάσταση της βιβλιοθήκης στο μηχάνημα του χρήστη, διαδικασία που ίσως αποδειχθεί απαιτητική. Αρχικά, πρέπει ο χρήστης να κατεβάσει το πακέτο της βιβλιοθήκης στο μηχάνημά του και στη συνέχεια να «χτίσει» εκεί τη βιβλιοθήκη. Εάν στο μηχάνημα βρίσκεται εγκατεστημένο λειτουργικό σύστημα Unix, η διαδικασία είναι σχετικά απλή για κάποιο χρήστη που είναι εξοικειωμένος με τις εντολές κονσόλας. Εάν το μηχάνημα λειτουργεί με Windows, η διαδικασία γίνεται πιο δύσκολη καθώς απαιτείται η εγκατάσταση και χρήση διάφορων άλλων λογισμικών για το «χτίσιμο» της βιβλιοθήκης, όπως το Microsoft

Visual Studio και Intel Fortran Compiler for Windows, όπου πρέπει να δοθεί πολλή προσοχή στη συμβατότητα των διαφόρων εκδόσεων των παραπάνω εργαλείων.

Βέβαια, το συγκεκριμένο βήμα θα μπορούσε να παραληφθεί μέσω της απόκτησης έτοιμης και ήδη χτισμένης βιβλιοθήκης. Στο συγκεκριμένο τρόπο, όμως, υπάρχει ο εξής περιορισμός: τα χαρακτηριστικά του μηχανήματος στο οποίο έχει χτιστεί η συγκεκριμένη έκδοση πρέπει να είναι ίδια με του δικού μας μηχανήματος. Για παράδειγμα, αν το μηχάνημά μας έχει 32-bits λειτουργικό σύστημα δεν μπορούμε να χρησιμοποιήσουμε βιβλιοθήκη που έχει χτιστεί για σύστημα 64bits. Για αυτό το λόγο είναι προτιμότερο η βιβλιοθήκη που θα χρησιμοποιήσουμε να έχει χτιστεί στο δικό μας μηχάνημα.

Μόλις η παραπάνω διαδικασία ολοκληρωθεί, ο χρήστης πρέπει να διαλέξει τη ρουτίνα η οποία ικανοποιεί τις απαιτήσεις του και συμβαδίζει με τα χαρακτηριστικά του προβλήματος. Για παράδειγμα, στην επίλυση ενός συστήματος της μορφής  $AX=B$ , οι ιδιότητες του πίνακα  $A$  είναι αυτές που καθορίζουν ποια ρουτίνα πρέπει να χρησιμοποιηθεί.

Στη συνέχεια, πρέπει να δημιουργηθεί πρόγραμμα γραμμένο σε κάποια γλώσσα προγραμματισμού της επιλογής του χρήστη, συνήθως σε Fortran, C ή C++ στο οποίο θα πρέπει να δηλωθεί ο φάκελος που έχει αποθηκευτεί η βιβλιοθήκη ώστε να είναι ορατή από τον κώδικα. Αυτό είναι εφικτό μόνο αν ο χρήστης έχει γνώσεις προγραμματισμού, διαφορετικά η χρήση της είναι αδύνατη.

Στο επόμενο κεφάλαιο παρουσιάζεται η διαδικασία με την οποία μπορεί να κληθεί μία ρουτίνα Lapack μέσω μία υπηρεσίας διαδικτύου.

## 3

# Σχεδίαση και υλοποίηση υπηρεσιών

Στο κεφάλαιο αυτό περιγράφονται αναλυτικά τα βήματα που απαιτούνται για την υλοποίηση μιας Java web service η οποία καλεί ρουτίνα της Lapack, δηλαδή ρουτίνα γραμμένη σε γλώσσα Fortran. Πρόκειται για μία διαδικασία συγκεκριμένων βημάτων που μπορούν να ακολουθηθούν για την υλοποίηση όλων των ρουτίνων της Lapack ως web services. [6]

### 3.1 Σχεδίαση διαδικασίας

Για την υλοποίηση web services θα χρησιμοποιήσουμε τα παρακάτω εργαλεία:

#### NetBeans IDE

Ως IDE (Integrated development environment) επιλέχθηκε το NetBeans. Υποστηρίζει τη δημιουργία εφαρμογών σε πολλές γλώσσες προγραμματισμού, web services αλλά και αποτελεί ένα χρήσιμο εργαλείο για τη δημιουργία Graphical User Interfaces (GUI). Είναι εύκολο σε χρήση και εγκατάσταση και διαθέσιμο σε πολλές γλώσσες. Οι μόνες απαιτήσεις που υπάρχουν για τη λειτουργία και χρήση του είναι η ύπαρξη Java Virtual Machine (JVM), καθώς το NetBeans είναι γραμμένο σε Java αλλά και του Java Developers Kit (JDK), το οποίο είναι απαραίτητο για την ανάπτυξη Java εφαρμογών.

#### Glassfish Application Server

Πρόκειται για έναν Java Application Server, ελεύθερο, open-source, ιδιαίτερα γνωστό, εύκολο στη χρήση και απλό στην εγκατάσταση. Η μόνη προϋπόθεση για την εγκατάστασή του είναι η ύπαρξη JDK στο μηχάνημα. Εδώ θα γίνονται deploy οι υπηρεσίες που δημιουργούνται και εδώ θα αναφέρονται οι εφαρμογές που επιθυμούν να τις καλέσουν.

## Java

Ως γλώσσα προγραμματισμού επιλέχθηκε η Java. Πρόκειται για μία απλή γλώσσα, εύκολη στην κατανόηση και υψηλής απόδοσης που ενδείκνυται για την ανάπτυξη διαδικτυακών εφαρμογών. Το ιδιαίτερο σε αυτή την περίπτωση είναι πως πρέπει μέσω της Java να καλούνται ρουτίνες που είναι γραμμένες σε Fortran, όμως οι δύο αυτές γλώσσες δεν μπορούν να «μιλήσουν» μεταξύ τους. Σε αυτή την περίπτωση, θα χρησιμοποιήσουμε το JNI (Java Native Interface). Πρόκειται για ένα software framework το οποίο επιτρέπει σε κώδικα γραμμένο σε Java να καλέσει εφαρμογές ή ρουτίνες που είναι γραμμένες σε native κώδικα, δηλαδή σε μία διαφορετική γλώσσα, όπως C, C++ ή assembly, η οποία με τη σειρά της θα καλεί τη ρουτίνα της fortran. Συγκεκριμένα, θα χρησιμοποιήσουμε τη γλώσσα C, συνεπώς τα βήματα της κλήσης είναι:

**Java → (JNI) → C → Fortran**

Επίσης, είναι απαραίτητη η δημιουργία μιας βιβλιοθήκης την οποία θα χρησιμοποιεί η υπηρεσία για την κλήση της συνάρτησης και την οποία θα «φορτώνει» ο Glassfish Application Server κάθε φορά που επανεκκινείται.

Συγκεκριμένα, για την κλήση της ρουτίνας από την υπηρεσία (δηλαδή από τον κώδικα Java) απαιτούνται τα παρακάτω:

- Να γίνει η δήλωση της native C ρουτίνας στον κώδικα της Java.
- Να δημιουργήσουμε ένα header file .h που θα χρησιμοποιηθεί από τον κώδικα της C. Σε αυτό φαίνεται η δήλωση της native C ρουτίνας δίνοντας έμφαση στους τύπους των παραμέτρων.
- Υλοποίηση της native C ρουτίνας, χρησιμοποιώντας το περιεχόμενο του header file και κλήση της ρουτίνας Fortran. Ο συγκεκριμένος κώδικας χρησιμοποιείται για τη δημιουργία της βιβλιοθήκης.

## 3.2 Υλοποίηση διαδικασίας

Έστω πως θέλουμε να υλοποιήσουμε την υπηρεσία που καλεί τη ρουτίνα DPOSV της Lapack και η οποία λύνει το γραμμικό σύστημα  $AX = B$  όπου ο A είναι ένας τετράγωνος και θετικά ορισμένος πίνακας.

(Θεωρούμε ότι η δημιουργία των υπηρεσιών γίνεται σε Unix σύστημα)

### Βήμα 1 - Δημιουργία project

Δημιουργούμε ένα νέο project τύπου WebApplication στο NetBeans με όνομα DposvWs.

### Βήμα 2 - Δημιουργία υπηρεσίας

Για τη δημιουργία της java class που αποτελεί την υπηρεσία μας κάνουμε δεξί κλικ στο όνομα του project επιλέγουμε New → WebService. Την ονομάζουμε DposvWs και θέτουμε όνομα πακέτου το uth.sws4hpsc.lapack.

### Βήμα 3 - Δημιουργία του αρχείου .c και της απαιτούμενης βιβλιοθήκης

Αυτό το βήμα είναι απαραίτητο για να επιτευχθεί η κλήση του κώδικα fortran από κώδικα java.

- i. Πηγαίνουμε στο φάκελο του project: **../DposvWs/src/java** και δημιουργούμε το **c\_dposv.c** αρχείο από όπου καλείται η συγκεκριμένη ρουτίνα.
- ii. Πηγαίνουμε στο φάκελο του project: **../DposvWs/src/java** και δημιουργούμε το **c\_dposv.c** αρχείο από όπου καλείται η συγκεκριμένη ρουτίνα.
- iii. Προσθέτουμε τη δήλωση της native C ρουτίνας στην κλάση java class :  
**private native int dposv(char UPLO, long N, long NRHS, double A[], long LDA, double B[], long LDB);**
- iv. Κάνουμε compile το εν λόγω αρχείο με την εντολή :  
**Javac DposvWs.java**
- v. Δημιουργούμε το αρχείο .h μέσω της εντολής javah :  
**Javah -classpath . -jni uth.sws4hpsc.lapack.DposvWs**  
Μετά την εκτέλεση αυτής της εντολής, δημιουργείται το εν λόγω αρχείο που περιέχει τη δήλωση της συνάρτησης .c όπου δίνεται έμφαση στον τύπο των παραμέτρων.
- vi. Η παραπάνω δήλωση προστίθεται στο .c αρχείο και πλέον το αρχείο είναι έτοιμο για να γίνει compile με την εντολή:  
**Cc -c -I/<path\_pointing\_to\_jdk>/include  
-I/<path\_pointing\_to\_jdk>/include/linux c\_dposv.c**



- vii. Δημιουργούμε τη βιβλιοθήκη που πρέπει να φορτώνει η υπηρεσία για να είναι δυνατή η κλήση της fortran ρουτίνας:

```
Ld -G -z defs c_dposv.o -o libsws4hpscDposv.so <path_to_lapack_library>  
/usr/lib/libgfortran.so.3 -lm -lc
```

Πλέον, δημιουργήσαμε μία Java web service που μπορεί να καλεί τη συγκεκριμένη Fortran ρουτίνα.

#### Βήμα 4 - Προσθήκη λειτουργικότητας

Σε αυτό το βήμα προσθέτουμε λειτουργίες στην υπηρεσία οι οποίες θα τις επιτρέπουν όχι απλά να την καλέσει λαμβάνοντας το αποτέλεσμα αλλά και να χειριστεί τα αριθμητικά δεδομένα.

Συγκεκριμένα, προσθέτουμε τις 4 παρακάτω λειτουργίες:

- Καλείται η ρουτίνα δίνοντας όλες τις παραμέτρους. Επιστρέφεται το αποτέλεσμα της επίλυσης του συστήματος, το μέγεθός του και ένα μοναδικό ID μέσω του οποίου μπορεί να γίνει μελλοντική αναφορά στο αποτέλεσμα.
- Δεν καλείται η ρουτίνα, αλλά γίνεται αναφορά σε ήδη αποθηκευμένο αποτέλεσμα. Επιστρέφεται το αποτέλεσμα, καθώς και το μέγεθός του.
- Καλείται η ρουτίνα, δίνοντας όλες τις παραμέτρους εκτός από τον πίνακα A. Σε αυτόν γίνεται αναφορά μέσω του μοναδικού ID. Επιστρέφεται το αποτέλεσμα της επίλυσης του συστήματος, το μέγεθός του και ένα μοναδικό ID μέσω του οποίου μπορεί να γίνει μελλοντική αναφορά στο αποτέλεσμα.
- Καλείται η ρουτίνα, δίνοντας όλες τις παραμέτρους εκτός από τους πίνακες A και B. Σε αυτούς γίνεται αναφορά μέσω των μοναδικών IDs. Επιστρέφεται το αποτέλεσμα της επίλυσης του συστήματος, το μέγεθός του και ένα μοναδικό ID μέσω του οποίου μπορεί να γίνει μελλοντική αναφορά στο αποτέλεσμα.

Επίσης, χρησιμοποιούμε βοηθητικές κλάσεις αλλά και υπηρεσίες<sup>2</sup>:

---

<sup>2</sup> Υλοποιημένες από το Δημήτρη Μπένη.

### Υπηρεσία IdManager

Αντιστοιχεί σε κάθε αποτέλεσμα που παράγεται αλλά και στους πίνακες που συμμετέχουν ένα μοναδικό ID. Γνωρίζοντάς το, ο χρήστης μπορεί να αναφερθεί σε αυτό.

### Υπηρεσία DataManager

Προσφέρει τη δυνατότητα της αποθήκευσης αλλά και της ανάκτησης πληροφορίας (μέσω του ID).

### Υπηρεσία DirectoryManager

Δημιουργεί directories εκεί που τρέχει η υπηρεσία (στην προκειμένη περίπτωση στο cluster) για την ομαδοποίηση των δεδομένων ανά κλήση. Για παράδειγμα, ομαδοποίηση σε δεδομένα εισόδου και δεδομένα εξόδου.

### Υπηρεσία PathManager

Διαβάζει το περιβάλλον και καθορίζει το μονοπάτι που υποδεικνύει που θα δημιουργηθούν τα παραπάνω directories.

### Κλάση LapackService

Αποτελεί τη base class για κάθε υπηρεσία που δημιουργούμε. Παρέχει βοηθητικές συναρτήσεις για το χειρισμό των δεδομένων και την κλήση της υπηρεσίας, όπως για παράδειγμα, τη μετατροπή των πινάκων από byte σε αριθμητική πληροφορία. Για να είναι ορατή από την java class που φτιάχνουμε, δημιουργούμε φάκελο `../java/uth/common/utills` και κάνουμε copy & paste τη συγκεκριμένη κλάση εκεί, δηλώνοντας την παράλληλα στον κώδικα της υπηρεσίας μας.

Για να είναι δυνατή η χρήση των 4 βοηθητικών υπηρεσιών που αναφέρθηκαν πρέπει να προστεθούν 4 web service references. Αυτό γίνεται κάνοντας δεξί κλικ στο όνομα του project και επιλέγοντας New Web Service Client. Στην επιλογή WSDL γράφουμε την αναφορά στο αντίστοιχο WSDL αρχείο (θεωρούμε ότι γίνεται η αναφορά στις υπηρεσίες που έχουν χτιστεί στο Cluster). Δηλαδή, προστίθενται οι αναφορές:

<http://195.251.17.2:8080/IdManager/IdManagerService?wsdl>

<http://195.251.17.2:8080/PathManager/PathManagerService?wsdl>

<http://195.251.17.2:8080/DirectoryManager/DirectoryManagerService?wsdl>  
<http://195.251.17.2:8080/DataManager/DataManagerService?wsdl>

### Βήμα 5 – Service Deployment

Για να ολοκληρωθεί η διαδικασία πρέπει η υπηρεσία να γίνει deploy στον Glassfish Application Server. Κάνοντας Clean and Build στο project παράγεται το .war αρχείο. Απαραίτητη είναι η αντιγραφή της βιβλιοθήκης που δημιουργήθηκε νωρίτερα στο Cluster και συγκεκριμένα στο φάκελο που βρίσκονται όλες οι βιβλιοθήκες από όπου φορτώνονται κάθε φορά που ο Server επανεκκινείται. Στη συνέχεια, κάνουμε login στο Server και επιλέγουμε Deploy a Web Service. Πλέον η υπηρεσία είναι έτοιμη να κληθεί και ικανή να επιστρέψει αποτέλεσμα.

Πρέπει να σημειωθεί εδώ πως αν η βιβλιοθήκη έχει δημιουργηθεί σε 32-bitο λειτουργικό σύστημα, ενώ Glassfish Server λειτουργεί σε 64-bitο, τότε η βιβλιοθήκη πρέπει να ξαναχτιστεί για 64-bitο σύστημα. Συγκεκριμένα, πρέπει να:

- Αντιγραφούν τα αρχεία .c και .h εκεί που θα χτιστεί η βιβλιοθήκη (σε κάποιο directory του Cluster).
- Γίνει compile το αρχείο .c με την εντολή:

```
cc -c -I/usr/java/jdk1.6.0_20/include -I/usr/java/jdk1.6.0_20/include/linux  
c_dposv.c
```

- Δημιουργηθεί εκ νέου η βιβλιοθήκη με την εντολή:

```
ld -G -z defs c_dposv.o -o libsws4hpscDposv.so  
/opt/acml4.4.0/gfortran/lib/libacml.a /usr/lib64/libgfortran.so.3 -lm -lc
```

Τέλος, χρησιμοποιώντας τον παραπάνω οδηγό υλοποιήσαμε 3 υπηρεσίες διαδικτύου τις DposvWs, DpbsWs και DppsvWs οι οποίες χρησιμοποιούν αντίστοιχα τις Lapack ρουτίνες:

**Dposv** : επιλύει γραμμικό σύστημα  $AX = B$  όπου A τετράγωνος, θετικά ορισμένος πίνακας

**Dpbsv** : επιλύει γραμμικό σύστημα  $AX = B$  όπου A τετράγωνος, θετικά ορισμένος πίνακας σε band μορφή

**Dppsv** : επιλύει γραμμικό σύστημα  $AX = B$  όπου A τετράγωνος, θετικά ορισμένος πίνακας σε packed μορφή

Τα αντίστοιχα αρχεία WSDL είναι τα:

<http://195.251.17.2:8080/DposvWs/DposvWsService?wsdl>

<http://195.251.17.2:8080/DpbsvWs/DpbsvWsService?wsdl>  
<http://195.251.17.2:8080/DppsvWs/DppsvWsService?wsdl>

### Βήμα 6 – Create a client

Απαραίτητο στοιχείο για την κλήση μιας υπηρεσίας είναι η εφαρμογή του πελάτη. Εδώ δίνονται οι αναγκαίες παράμετροι για την κλήση της υπηρεσίας (οι πίνακες A, B και ενδεχομένως άλλες παράμετροι) και εδώ επιστρέφεται το αποτέλεσμα (στην προκειμένη περίπτωση η λύση X του συστήματος). Για τη δημιουργία ενός java client στο Net Beans που καλεί την υπηρεσία DposvWs απαιτούνται τα ακόλουθα βήματα:

- Δημιουργία νέου project: New Project > Java > Java Application.
- Για να είναι δυνατή η κλήση της συγκεκριμένης υπηρεσίας πρέπει να προστεθεί μία αναφορά σε αυτή. Με δεξί κλικ στο όνομα του project επιλέγουμε New Web Service Client και στο πεδίο WSDL προσθέτουμε :

<http://195.251.17.2:8080/DposvWs/DposvWsService?wsdl>

- Κάνοντας extend το φάκελο Web Service References του project, βλέπουμε όλες τις μεθόδους που υποστηρίζει η υπηρεσία. Κάνοντας drag & drop στον κώδικα μας, προστίθεται η δήλωση παραμέτρων αλλά και η κλήση της υπηρεσίας στη java class του πελάτη. Θέτοντας τις κατάλληλες τιμές στις παραμέτρους και πατώντας Run, επιστρέφονται τα αποτελέσματα της κλήσης.

Στο συγκεκριμένο βήμα η κλήση της υπηρεσίας γίνεται προγραμματιστικά μέσω java client. Για την αυτοματοποίηση της κλήσης, δημιουργήσαμε μία γραφική διεπαφή (graphical interface) όπου καθορίζοντας την υπηρεσία που επιθυμούμε να καλέσουμε και συμπληρώνοντας τις τιμές των παραμέτρων σε μία φόρμα κλήσης, παίρνουμε άμεσα το αποτέλεσμα (Κεφάλαιο 5).

### Βήμα 7 – Υπολογισμός ακρίβειας αποτελέσματος

Στο προηγούμενο βήμα, η κλήση της υπηρεσίας επιστρέφει το αποτέλεσμα της επίλυσης του συστήματος  $AX = B$ . Επειδή ενδέχεται η λύση αυτή να μην ταυτίζεται με την ιδανική αλλά να αποκλίνει, υπάρχει το προαιρετικό βήμα του υπολογισμού ακρίβειας αποτελέσματος. Σε αυτό υπολογίζουμε την ευκλείδεια νόρμα της παράστασης  $(AX - B)$  και βάσει της τιμής της παράστασης  $\|AX - B\|_2$  διακρίνονται 2 περιπτώσεις:

- Εάν είναι ίση με το 0, τότε η λύση X αποτελεί την ιδανική λύση του συστήματος.

- Εάν η λύση είναι διάφορη του 0, τότε η λύση X δεν αποτελεί την ιδανική αλλά αποκλίνει. Όσο μικρότερη είναι η τιμή απόκλισης και τείνει προς το 0, τόσο πιο ακριβής θεωρείται.

Στην προκειμένη περίπτωση, θα χρησιμοποιηθεί μία BLAS ρουτίνα για τη διευκόλυνση των υπολογισμών η οποία θα υλοποιηθεί και ως υπηρεσία. Πρόκειται για την DGEMM [21] η οποία εκτελεί την πράξη:  $\alpha * op(A) * op(B) + \beta * C$ , όπου A, B και C πίνακες διπλής ακρίβειας και η οποία θα υλοποιηθεί ως υπηρεσία DgemmWs με τον τρόπο που περιγράφηκε νωρίτερα και το αντίστοιχο WSDL αρχείο είναι το :

<http://195.251.17.2:8080/DgemmWs/DgemmWsService?wsdl>

Υποστηρίζονται 2 λειτουργίες :

- Καλείται η ρουτίνα δίνοντας όλες τις παραμέτρους. Επιστρέφεται το αποτέλεσμα της επίλυσης του συστήματος, το μέγεθός του και ένα μοναδικό ID μέσω του οποίου μπορεί να γίνει μελλοντική αναφορά στο αποτέλεσμα.
- Δεν καλείται η ρουτίνα, αλλά γίνεται αναφορά σε ήδη αποθηκευμένο αποτέλεσμα. Επιστρέφεται το αποτέλεσμα, καθώς και το μέγεθός του.

Συγκεκριμένα, οι πίνακες  $op(A)$  και  $op(B)$  μπορεί να είναι οι δεδομένοι A και B ή οι αντίστροφοί τους, ενώ οι μεταβλητές alpha και beta είναι αριθμοί διπλής ακρίβειας που καθορίζουν την κλιμάκωση των πινάκων. Στη συγκεκριμένη περίπτωση επιθυμούμε την εκτέλεση ενός απλού πολλαπλασιασμού πινάκων ( $A \times B$ ), συνεπώς ο πίνακας C θα είναι ένας μηδενικός πίνακας ενώ οι παράμετροι alpha και beta θα λάβουν κατά την κλήση της υπηρεσίας τις τιμές 1.0 και 0.0 αντίστοιχα.

Στην εφαρμογή πελάτη που θα την καλεί, οι δικοί μας πίνακες A και X (λύση του συστήματος) θα αποτελούν τις παραμέτρους A και B τις υπηρεσίας. Συνεπώς, η υπηρεσία μας επιστρέφει το γινόμενο της παράστασης ( $A \times B$ ) και απομένει ο υπολογισμός της νόρμας. Η εφαρμογή πελάτη μόλις λάβει το επιστρεφόμενο γινόμενο υπολογίζει και επιστρέφει για κάθε γραμμή του συστήματος την ευκλείδεια νόρμα.

Η παραπάνω διαδικασία μπορεί να κληθεί ως μεμονωμένη υπηρεσία για τον υπολογισμό ποιότητας της λύσης και ακρίβειας αποτελέσματος. Ως μελλοντική υλοποίηση, θα μπορούσε μέσω της BPEL να συνδυαστεί με την υπηρεσία που επιστρέφει τη λύση του συστήματος για τη δημιουργία ενός workflow το οποίο εκτός από τη λύση του συστήματος να επέστρεφε και την ποιότητά της.

Πλέον, έχουμε δημιουργήσει μία υπηρεσία διαδικτύου η οποία πρέπει να δηλωθεί σε ένα registry ώστε να μπορούν οι clients να την ανακαλύψουν. Για αυτό το σκοπό χρησιμοποιήσαμε το ebXML framework, ένα πρότυπο ηλεκτρονικού εμπορίου.

## 4

## Electronic Business XML

### (ebXML)

Σε αυτό το κεφάλαιο θα παρουσιάσουμε το framework ebXML περιγράφοντας την αρχιτεκτονική και τη λειτουργία του και εστιάζοντας στα σύνθετα χαρακτηριστικά του.

#### 4.1 Περιγραφή αρχιτεκτονικής και λειτουργίας

Πρόκειται για ένα πρότυπο παγκόσμιου ηλεκτρονικού εμπορίου που δημιουργήθηκε το 2000 από την OASIS και την UN/CEFACT (United Nations Center For Trade Facilitation & Electronic Business). Αποτελεί ένα ολοκληρωμένο framework ηλεκτρονικού εμπορίου που ευνοεί την ηλεκτρονική διαλειτουργικότητα και επιτρέπει σε οργανισμούς ανεξαρτήτως μεγέθους και προέλευσης να επικοινωνούν και να συνεργάζονται ανταλλάσσοντας μηνύματα βασισμένα σε XML. [7]

Η αρχιτεκτονική του διακρίνεται σε 5 πεδία [8] :

##### **Business Process and Information Models :**

Ορίζουν τον τρόπο με τον οποίο περιγράφονται τα business processes έτσι ώστε να γίνεται κατανοητό από τις υπόλοιπες επιχειρήσεις. Ακολουθούν το Business Process Specification Schema (BPSS) και πρόκειται για έγγραφο XML.

##### **Company Profiles :**

Περιγράφεται η επιχείρηση ως οντότητα ανεξάρτητα από το πώς εκτελεί τις λειτουργίες της. Ένα τυπικό company profile template περιλαμβάνει μία σύντομη περιγραφή της επιχείρησης (ιστορικό, τομέας δραστηριοποίησης, παραγόμενα προϊόντα) καθώς και λεπτομέρειες

(εγκαταστάσεις, προσωπικό, οικονομικά μεγέθη – πρώτο προϊόν σε πωλήσεις-, ανταγωνιστές, εμπόδια).

### **Messaging Services :**

Περιλαμβάνει ένα σύνολο από υπηρεσίες και πρωτόκολλα που επιτρέπουν την επικοινωνία των επιχειρήσεων και την ανταλλαγή δεδομένων. Επιτρέπεται να χρησιμοποιηθεί οποιοδήποτε πρωτόκολλο επιπέδου εφαρμογής, ενώ τα πιο συνηθισμένα είναι HTTP, FTP, SMTP. Επίσης, για την επίτευξη ασφαλέστερης επικοινωνίας μπορούν να χρησιμοποιηθούν secure protocols ή και κρυπτογραφικές τεχνικές, ενώ οι ψηφιακές υπογραφές μπορούν να εγγυηθούν την αυθεντικότητα της ταυτότητας του συνομιλητή. Τα μηνύματα που ανταλλάσσονται είναι XML και διαχωρίζονται σε 2 μέρη:

- Soap message container : πρόκειται για το υποχρεωτικό τμήμα του μηνύματος και περιέχει τα αναγνωριστικά του μηνύματος και πληροφορίες δρομολόγησης.
- Payload containers : πρόκειται για το προαιρετικό τμήμα του μηνύματος και περιέχει την πληροφορία που θα ανταλλαχθεί.

### **Collaborative Partner Profiles and Agreement :**

Πρόκειται για τη συμφωνία την οποία συνάπτουν οι επιχειρήσεις και η οποία καθορίζει τους όρους της ανταλλαγής – συνεργασίας. Αρχικά, όλες οι επιχειρήσεις συμπληρώνουν το collaborative partner profile (CPP), ένα XML έγγραφο το οποίο περιέχει πληροφορίες ταυτοποίησης (globally unique identifier), υποστηριζόμενες λειτουργίες για ασφαλή επικοινωνία και στοιχεία επικοινωνίας (delivery channels, transfer protocols). Στη συνέχεια, για να επέλθει η συμφωνία οι 2 πλευρές ανταλλάσσουν CPPs και ορίζουν ένα collaborative partner agreement (CPA) που αποτελεί την επίσημη συμφωνία των δύο πλευρών για τις συναλλαγές τους και καθορίζει τεχνικές και μεθόδους που θα χρησιμοποιηθούν για την εγγύηση της ασφάλειας και της αυθεντικοποίησης κατά τις συναλλαγές τους.

### **Registry and Repository :**

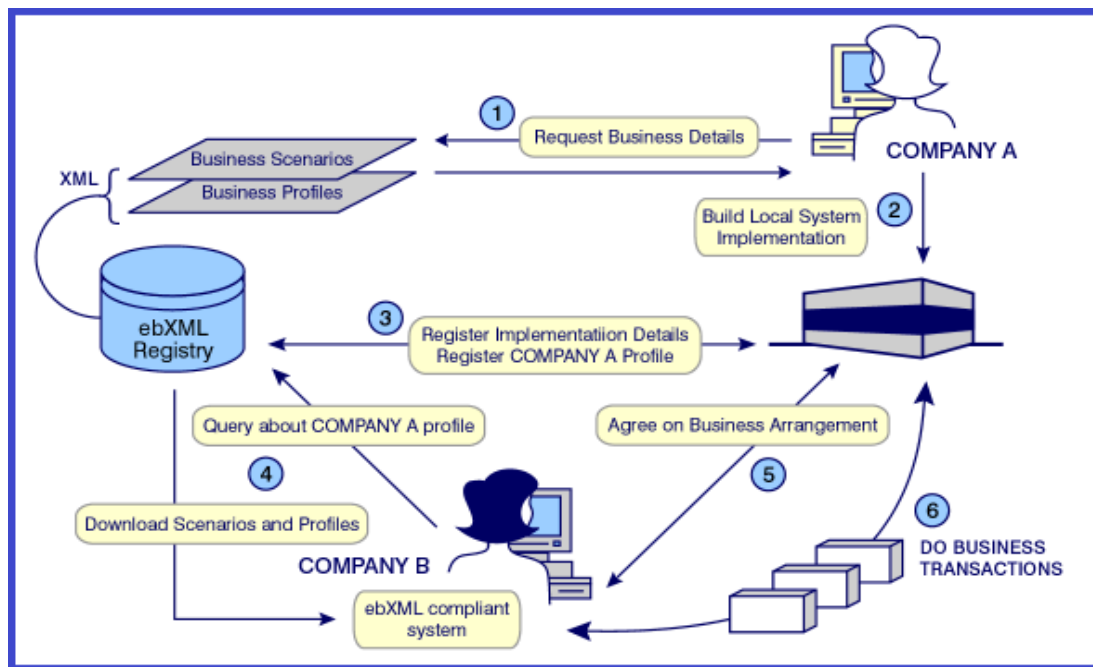
Πρόκειται για το πιο σημαντικό τμήμα του ebXML, στο οποίο βασίζεται ολόκληρη η λειτουργία του. Αποτελεί για την ψηφιακή πληροφορία (digital content) ό,τι και η βιβλιοθήκη για τα βιβλία και το έντυπο υλικό. Εδώ αποθηκεύονται όλα τα XML έγγραφα τα οποία αναφέρθηκαν νωρίτερα και εδώ γίνονται όλες οι επερωτήσεις για την ανακάλυψη business partners.

Συγκεκριμένα, όλη η πληροφορία (δεδομένα) αποθηκεύεται στο ebXML Repository, ενώ στο ebXML Registry διατηρούνται δεδομένα που αφορούν την αποθηκευμένη πληροφορία, τα λεγόμενα «μεταδεδομένα». Ο ρόλος του τελευταίου είναι πολύ σημαντικός καθώς λειτουργεί

ως ευρετήριο για την ανακάλυψη της επιθυμητής πληροφορίας αλλά και μπορεί να χρησιμοποιηθεί για την κατηγοριοποίηση, ομαδοποίηση και συσχέτιση των αποθηκευμένων στο Repository δεδομένων.

Τα ιδιαίτερα χαρακτηριστικά του το καθιστούν ξεχωριστό και αποτελούν το λόγο για τον οποίο δε χρησιμοποιήθηκε μία απλή βάση δεδομένων αντί για το ίδιο.

Η αλληλεπίδραση και επικοινωνία των οργανισμών είναι μία διαδικασία που αποτελείται από μία σειρά βημάτων, όπως φαίνεται στην εικόνα που ακολουθεί.



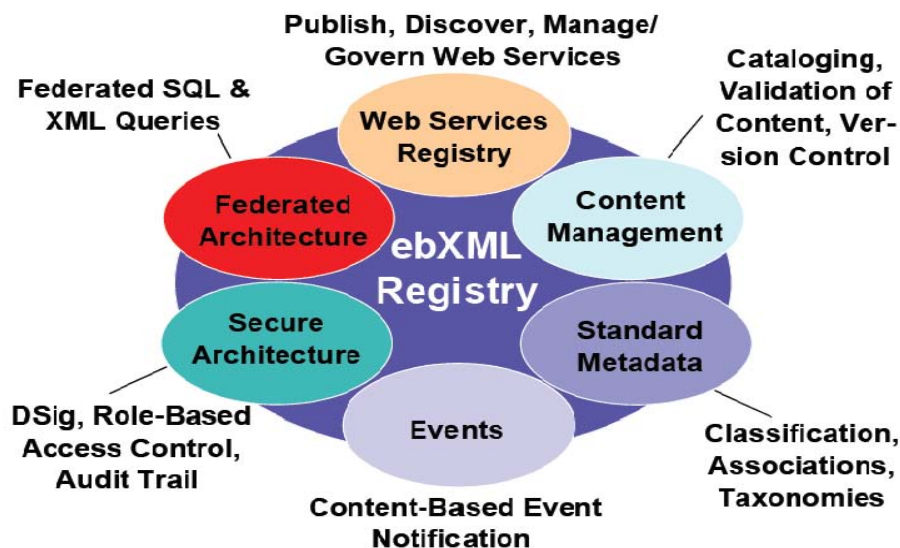
Εικόνα 4 - 1 Ο τρόπος λειτουργίας του eb-XML

1. Έστω ότι ο οργανισμός A θέλει να εγγραφεί στο ebXML. Τότε ζητάει τα specifications ebXML.
2. Χτίζει το δικό του προφίλ αφού προηγουμένως έχει ορίσει ποια θα είναι τα χαρακτηριστικά του. Δηλαδή, ποιες υπηρεσίες θα προσφέρει, ποιες είναι οι απαιτήσεις του από το συνεργάτη, ποια τα μεταδεδομένα και τέλος ποιες οι συσχετίσεις των αντικειμένων.
3. Πρέπει να ορίσει το δικό του CPP και να δημοσιεύσει το προφίλ του στο ebXML Registry ώστε να μπορούν οι υπόλοιποι οργανισμοί να τον ανακαλύψουν.



4. Έστω ότι ο οργανισμός Β επιθυμεί να συνεργαστεί με τον οργανισμό Α. Σε αυτή την περίπτωση, αναζητά το CPP του Α στο ebXML Registry και το εξετάζει για να διαπιστώσει αν συμβαδίζουν οι απαιτήσεις τους.
5. Σε αυτό το σημείο, οι δύο οργανισμοί έχουν τη δυνατότητα να διαπραγματευτούν και ενδεχομένως να συμφωνήσουν να συνεργαστούν σύμφωνα με τους όρους που θα επιβάλει το CPA που θα οριστεί.
6. Σε περίπτωση συμφωνίας, η συνεργασία και οι ανταλλαγές πληροφοριών και εγγράφων μπορούν να υλοποιηθούν.

## 4.2 Χαρακτηριστικά γνωρίσματα



Εικόνα 4 - 2 Τα χαρακτηριστικά του ebXML Registry με μια ματιά

Παρακάτω περιγράφονται τα κύρια χαρακτηριστικά του ebXML Registry [9]:

### **Web service registry ( Ευρετήριο υπηρεσιών διαδικτύου )**

Αποτελεί ένα χρήσιμο εργαλείο για τη δημοσίευση, ανακάλυψη και το χειρισμό των web services. Συγκεκριμένα, τα αρχεία WSDL που περιγράφουν τις web services αποθηκεύονται στο repository, ενώ στο registry καταχωρούνται σχετικά μεταδεδομένα που βοηθούν στην περιγραφή των αντίστοιχων υπηρεσιών.

### **Federated architecture ( Ομοσπονδιακή αρχιτεκτονική )**

Η συγκεκριμένη ιδιότητα οφείλεται στο γεγονός πως το ebXML Registry υποστηρίζει την κατανομημένη διαχείριση περιεχομένου. Δύο ή περισσότερα registries μπορούν να δημιουργήσουν χαλαρά διασυνδεδεμένη αρχιτεκτονική (ομοσπονδιακή αρχιτεκτονική) και να εμφανίζονται ως ένα ενιαίο registry. Όταν ο πελάτης θελήσει να πραγματοποιήσει μια αναζήτηση κάνει επερώτηση σε ένα από τα registries, αλλά η αναζήτηση πραγματοποιείται παράλληλα σε όλα τα registries της ομοσπονδίας. Ουσιαστικά, γίνεται αναζήτηση κατά μήκος όλης της ομοσπονδιακής αρχιτεκτονικής, συνεπώς εκτελείται μία ομοσπονδιακή επερώτηση (federated query). Ίδιας λογικής είναι και οι συσχετίσεις που μπορούν να δημιουργηθούν ανάμεσα στα αντικείμενα. Δηλαδή, ένα αντικείμενο ενός registry μπορεί να συσχετιστεί με αντικείμενο που ανήκει σε άλλο registry της ομοσπονδιακής αρχιτεκτονικής μέσω απομακρυσμένης αναφοράς αντικειμένου.

Η σημασία αυτής της ιδιότητας διαφαίνεται από το ακόλουθο παράδειγμα: αν όλες οι χώρες που αποτελούν κράτη-μέλη της Ευρωπαϊκής Ένωσης διατηρούσαν ένα ebXML registry για το Υπουργείο Υγείας της χώρας τους και όλα μαζί συνδέονταν σε μία ομοσπονδιακή αρχιτεκτονική τότε θα γινόταν πολύ πιο εύκολη η διαχείριση μιας κατάστασης επιδημίας πανευρωπαϊκά.

### **Secure architecture ( Ασφαλής αρχιτεκτονική )**

Η συγκεκριμένη ιδιότητα οφείλεται στις επιμέρους ιδιότητες του που το καθιστούν ικανό να καταγράφει τις κινήσεις των χρηστών, να επιβεβαιώνει την ταυτότητα του χρήστη μέσω ψηφιακών υπογραφών και να ελέγχει την ελευθερία αλλά και την πρόσβαση των χρηστών στα δεδομένα του. Το registry υποστηρίζει Access Control Policies (ACP) τα οποία καθορίζουν για κάθε αντικείμενο του ευρετηρίου ποιος χρήστης έχει δικαίωμα πρόσβασης σε αυτό και τι είδους ενέργειες μπορεί να πραγματοποιήσει. Για παράδειγμα, το default ACP επιτρέπει σε όλους να διαβάζουν το αντικείμενο, αλλά μόνο στον ιδιοκτήτη να το μεταβάλλει ή να το διαγράψει.

Το πιο απλό είδος ελέγχου πρόσβασης που υποστηρίζεται είναι η δυνατότητα επιλογής δημόσιου ή ιδιωτικού ευρετηρίου όταν αυτό δημιουργείται. Στην πρώτη περίπτωση οποιοσδήποτε χρήστης μπορεί να κάνει επερωτήσεις στο ευρετήριο, ενώ στη δεύτερη μόνο οι οργανισμοί – συνεργάτες, περιορίζοντας έτσι την πρόσβαση του κοινού στις υπηρεσίες που προσφέρονται.

### **Events ( Ειδοποίηση γεγονότων )**

Υποστηρίζεται η ειδοποίηση γεγονότων με βάση το περιεχόμενο. Ο χειρισμός γίνεται με τον ακόλουθο τρόπο: όταν ένας χρήστης θελήσει να ειδοποιηθεί μόλις συμβεί ένας τύπος γεγονότος (το γεγονός αυτό μπορεί να είναι δημοσίευση ενός αντικειμένου συγκεκριμένου τύπου ή μεταβολή ενός ήδη υπάρχοντος αντικειμένου), δημιουργείται μία συνδρομή που περιλαμβάνει πληροφορίες σχετικές με το πώς και το πού θα σταλεί η ειδοποίηση γεγονότος αλλά και ποια επερώτηση να εκτελεστεί. Από τα γεγονότα που συνέβησαν πρόσφατα και χρησιμοποιώντας την επιλεγμένη επερώτηση, εξετάζεται αν υπάρχει κάποιο γεγονός που να είναι το επιθυμητό. Σε αυτή την περίπτωση, αποστέλλεται η ειδοποίηση στον ενδιαφερόμενο συνδρομητή με τον τρόπο που ορίζεται από τη συνδρομή (για παράδειγμα, αποστολή e-mail σε έναν προκαθορισμένο λογαριασμό) η οποία περιέχει το αντικείμενο που προέκυψε από την επερώτηση.

### **Standard metadata ( Τυποποίηση μεταδεδομένων )**

Παρέχεται η δυνατότητα κατηγοριοποίησης, ομαδοποίησης και συσχέτισης μεταξύ των δεδομένων του ευρετηρίου. Όλη η πληροφορία που μπορεί να αποθηκευτεί στο registry για τα μεταδεδομένα παρουσιάζεται αναλυτικά στο ebXML Registry Information Model [ebRIM].

### **Content management ( Χειρισμός περιεχομένου )**

Παρέχεται η δυνατότητα χειρισμού του περιεχομένου του repository με διάφορους τρόπους.

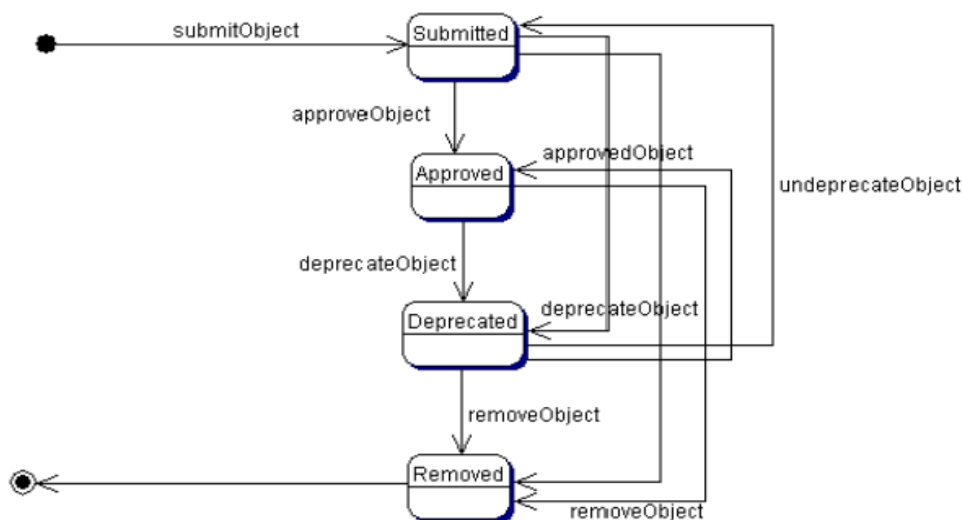
- **Content validation and cataloging**

Όταν τα δεδομένα υποβάλλονται στο registry, γίνεται έλεγχος της εγκυρότητας του περιεχομένου τους. Διαφορετικά, τα δεδομένα δε γίνονται αποδεκτά. Για παράδειγμα, εάν κατατεθεί ένα αρχείο WSDL είναι απαραίτητο όλα τα portTypes να έχουν ορισμένο SOAP binding. Επίσης, γίνεται έλεγχος του περιεχομένου τους από μία cataloging service. Από αυτό τον έλεγχο προκύπτουν μεταδεδομένα για τα συγκεκριμένα δεδομένα που βοηθούν στην ένταξη των τελευταίων σε κάποιο κατάλογο. Αυτή η πληροφορία ενδέχεται να χρησιμοποιηθεί ως παράμετρος για μελλοντική αναζήτηση των δεδομένων που κατατέθηκαν. Για παράδειγμα, όταν υποβληθεί μία εικόνα, τότε μπορεί να κατηγοριοποιηθεί ως έγχρωμη ή ασπρόμαυρη.

Το ebXML δε διαθέτει ουσιαστική γνώση για να μπορεί να διακρίνει τα περιεχόμενα, αλλά ούτε και για τη σωστή μορφή των δεδομένων για να τα αποδεχτεί. Για αυτό το λόγο έχει δημιουργηθεί μία web service η οποία καλείται κάθε φορά που υποβάλλονται δεδομένα ώστε να κάνει τους παραπάνω ελέγχους.

- **Life cycle management ( Χειρισμός του κύκλου ζωής )**

Από τη στιγμή που κάποιο δεδομένο γίνει αποδεκτό στο registry, τότε το σύστημα χειρίζεται όλα τα στάδια του κύκλου ζωής του ελέγχοντας την τρέχουσα έκδοσή του. Οι πιθανές καταστάσεις του είναι: να έχει υποβληθεί (submitted), να βρίσκεται σε έγκυρη κατάσταση και να έχει αναβαθμιστεί (approved), να είναι απαρχαιωμένο και να μη συμβαδίζει με την τρέχουσα έκδοση (deprecated/obsolete) και τέλος να αφαιρείται από το registry (removed). Στην ακόλουθη εικόνα, παρατηρούμε τον κύκλο ζωής ενός δεδομένου καθώς και τις πιθανές μεταβάσεις από μία κατάσταση σε άλλη.



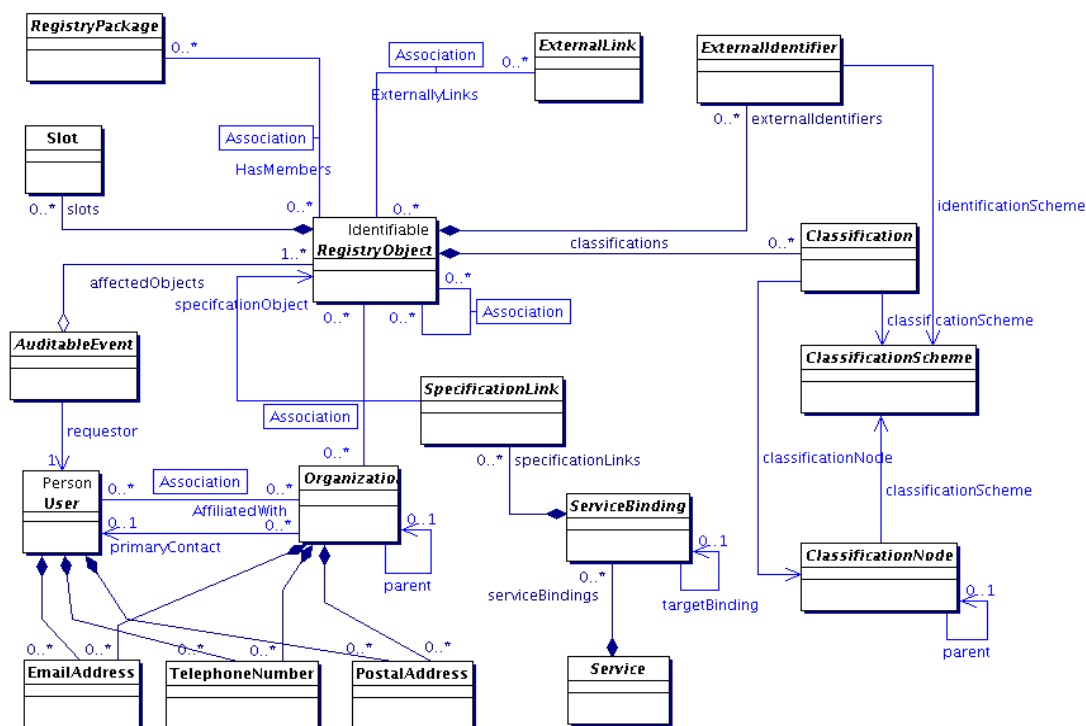
**Εικόνα 4 - 3 Ο κύκλος ζωής ενός registry object**

- **Content discovery ( Ανακάλυψη περιεχομένου )**

Σε περίπτωση που ο χρήστης επιθυμεί να ανακαλύψει κάποιο δεδομένο που διαθέτει κάποιες συγκεκριμένες ιδιότητες, μπορεί να πραγματοποιήσει αναζήτηση στο registry εκτελώντας ad hoc ερωτήσεις σε σύνταξη SQL92 ή XML.

Οι λεπτομέρειες της ερωτήσης δεν είναι εμφανείς στο χρήστη, ο οποίος είναι υπεύθυνος μόνο για να δώσει τις κατάλληλες παραμέτρους για την αναζήτηση.

## 4.3 EbXML Registry Information Model (eb - RIM)



Εικόνα 4 - 4 Μορφή του μοντέλου πληροφορίας του ebXML Registry

Παρακάτω θα αναλύσουμε τα βασικά στοιχεία του μοντέλου πληροφορίας του ebXML Registry [10] [11] :

### Registry Object

Είναι η βασική κλάση που χρησιμοποιείται από όλες τις κλάσεις του μοντέλου και παρέχει τα ελάχιστα μεταδεδομένα για τα registry objects. Αυτά είναι τα εξής:

- Name and Description (Όνομα και περιγραφή)

Οι τιμές αυτών των γνωρισμάτων αποτελούνται από μία ή περισσότερες συμβολοσειρές και μπορούν να είναι ίδιες ανάμεσα στα registry objects.

Οι τιμές τους είναι πολύ σημαντικές γιατί σε αυτά τα γνωρίσματα εκτελούνται αναζητήσεις στο registry για την ανακάλυψη αντικειμένων.

- Unique ID

Κάθε registry object έχει ένα καθολικά μοναδικό id βάσει του οποίου ξεχωρίζει από τα υπόλοιπα. Αποτελεί μία τιμή Uniform Resource Identifier, δηλαδή απαρτίζεται από ακολουθία χαρακτήρων και αριθμών.

- Logical ID

Σε περίπτωση που το registry object έχει πολλές εκδόσεις, τότε χρησιμοποιείται το λογικό id. Η τιμή του πρέπει να είναι διαφορετική για κάθε λογικό αντικείμενο, αλλά ίδια για τα αντικείμενα που αποτελούν διαφορετικές εκδόσεις του ίδιου registry object. Η τιμή του είναι επίσης Uniform Resource Identifier αλλά ενδέχεται να έχει και μορφή φιλική προς τον άνθρωπο.

### **Slot**

Σε περίπτωση που τα παραπάνω ήδη γνωρισμάτων δεν επαρκούν για την περιγραφή ενός registry object, τότε υπάρχει η δυνατότητα χρήσης του slot attribute για την επίδοση περαιτέρω γνωρισμάτων στο αντικείμενο. Πιο συγκεκριμένα, μπορούν να χρησιμοποιηθούν πολλά στιγμιότυπα για την απόδοση διαφορετικών πρόσθετων γνωρισμάτων με το όνομα του στιγμιότυπου να αποτελεί το όνομα του γνωρίσματος. Υπάρχει ο ακόλουθος περιορισμός: τα ονόματα των γνωρισμάτων που αφορούν το ίδιο αντικείμενο πρέπει να είναι απαραίτητα διαφορετικά μεταξύ τους, ενώ δεν ισχύει ο ίδιος περιορισμός μεταξύ των γνωρισμάτων που ανήκουν σε διαφορετικά αντικείμενα. Για παράδειγμα, εάν το αντικείμενο ήταν το προφίλ ενός ανθρώπου, τότε το συγκεκριμένο γνώρισμα μπορούσε να έχει το όνομα του επαγγέλματος ή της οικογενειακής κατάστασης.

### **External Identifier**

Για την ενίσχυση του identification του κάθε αντικειμένου και σε περίπτωση που το id και το logical id είναι ανεπαρκή, παρέχεται η επιλογή των external identifiers που μπορεί να έχει οποιαδήποτε τιμή που να ανήκει σε κάποιο σχήμα κατηγοριοποίησης.

### **Registry Package**

Χρησιμοποιείται για την ομαδοποίηση και την ιεράρχιση των registry objects. Συγκεκριμένα, τα στιγμιότυπα των registry package λειτουργούν ως το φάκελο και registry objects στιγμιότυπα λειτουργούν ως τα αρχεία που αποθηκεύονται στο φάκελο.

### **Association**

Παρέχει τη δυνατότητα δημιουργίας συσχετίσεων ανάμεσα στα registry objects. Από τα δύο αντικείμενα που συσχετίζονται, το ένα παίζει το ρόλο του sourceObject και το άλλο του targetObject. Η κάθε συσχέτιση ενδεχομένως να χαρακτηρίζεται από έναν τύπο συσχέτισης ο οποίος καθορίζει το είδος της συσχέτισης.

### **External Link**

Παρέχει τη δυνατότητα δημιουργίας συσχετίσεων ανάμεσα σε registry objects και σε περιεχόμενο που βρίσκεται εκτός registry σε κάποιο Uniform Resource Locator (URL). Για παράδειγμα, αν το registry object είναι ένας οργανισμός που παρέχει υπηρεσίες, τότε το external link του ενδεχομένως να είναι το website του οργανισμού.

### **Classification**

Παρέχει τη δυνατότητα κατηγοριοποίησης των registry objects χρησιμοποιώντας κάποιο από τα σχήματα κατηγοριοποίησης (classification scheme) που προσφέρονται με τους κόμβους κατηγοριοποίησης (classification nodes) να αποτελούν τις κατηγορίες που δημιουργούνται. Για παράδειγμα, εάν το αντικείμενο ήταν ένα πρόσωπο τότε η κατηγοριοποίηση με βάση το σχήμα «φύλο» κατηγοριοποιεί το αντικείμενο ως γυναίκα ή άντρα. Υπάρχει μία κατηγορία σχημάτων κατηγοριοποίησης, που ονομάζονται «κανονικά» (canonical) και υποστηρίζονται από κάθε ebXML Registry. Σε περίπτωση που δεν επαρκούν για τη λειτουργία του registry, υπάρχει η δυνατότητα επέκτασης των ήδη υπάρχοντων σχημάτων (για παράδειγμα, με προσθήκη νέων κόμβων κατηγοριοποίησης) αλλά και η δημιουργία νέων σχημάτων κατηγοριοποίησης.

### **Organization**

Πρόκειται για τον οργανισμό ο οποίος έχει δημοσιεύσει το αντίστοιχο αντικείμενο και σχετίζεται με τη σειρά του με κλάσεις που παρέχουν τα στοιχεία επικοινωνίας όπως email address, postal address και telephone number αλλά και person που αποτελεί την ομάδα των ανθρώπων που δουλεύουν για αυτό τον οργανισμό.

### **Auditable Event**

Πρόκειται για την καταγραφή των γεγονότων που επηρεάζουν το αντίστοιχο αντικείμενο. Σχετίζεται επίσης με την κλάση Person και συγκεκριμένα καταγράφει ακριβώς πώς επηρέασε το συγκεκριμένο άτομο το αντικείμενο.

### **Services**

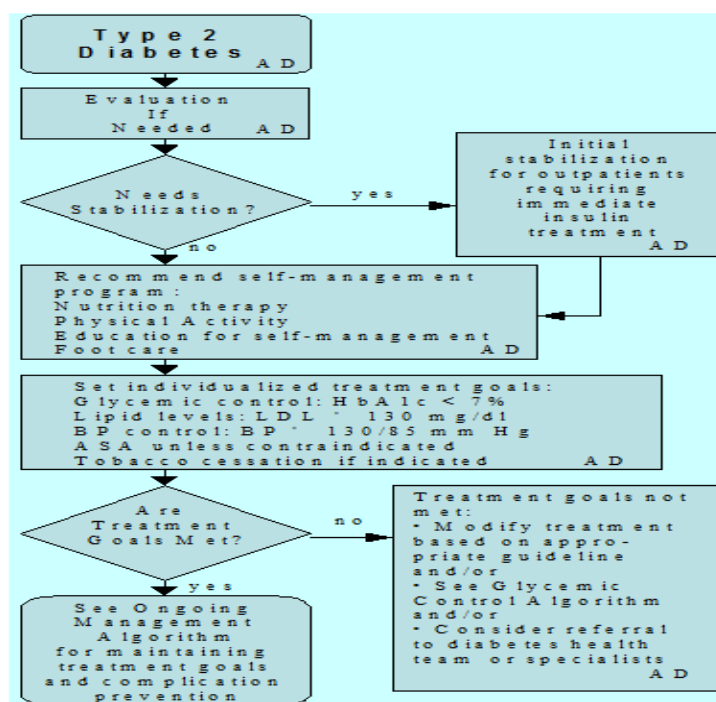
Πρόκειται για τις τεχνικές πληροφορίες που είναι διαθέσιμες και χρήσιμες για την προσπέλαση της κάθε υπηρεσίας. Εδώ περιλαμβάνονται τα αρχεία WSDL για την προσπέλαση των υπηρεσιών διαδικτύου αλλά και τα collaboration protocol profiles για την ανακάλυψη ενδεχόμενων συνεργατών.

## 4.4 Παραδείγματα χρήσης

Το freebXML αποτελεί μία open-source υλοποίηση του προτύπου ebXML και στη συγκεκριμένη εργασία χρησιμοποιήθηκε ως web service registry. Παρακάτω παρουσιάζονται περιπτώσεις διαφορετικής χρήσης [12].

### **SAGE ( Shareable Active Guideline Environment ) Clinical Guideline Registry**

Δημιουργήθηκε από την Arelon και συγκεκριμένα από τον Derrick Butler, ενώ μερικοί από τους συνεργαζόμενους φορείς είναι οι : Mayo Clinic<sup>3</sup>, Stanford Medical Informatics<sup>4</sup>, University of Nebraska Medical Center. Ο σκόπος της δημιουργίας του ήταν η ανάπτυξη μιας υποδομής που θα φιλοξενήσει κλινικές οδηγίες (guidelines) διαμοιραζόμενες και προσβάσιμες από πολλούς φορείς υγείας. Με τον όρο κλινικές οδηγίες εννοούμε οδηγίες αντιμετώπισης ασθενειών και περιστατικών, γραμμένες σε μορφή διαγραμμάτων ροής χρησιμοποιώντας προκαθορισμένη ορολογία. Οι συνεργαζόμενοι φορείς θα πρέπει να είναι σε θέση να ανανεώνουν το σύστημα προσθέτοντας εναλλακτικές οδηγίες ή οδηγίες για ασθένειες που δεν είναι καταχωρημένες.



Εικόνα 4 - 5 Παράδειγμα Guideline

<sup>3</sup> Μη κερδοσκοπικός οργανισμός και ιατρική ερευνητική ομάδα που εδρεύει στις Η.Π.Α.

<sup>4</sup> Ο τομέας ιατρικής πληροφορικής του πανεπιστημίου Stanford



### ***General Motors Software Factory Enabler Project***

Αρχικά, δημιουργήθηκε από τη Sun ενώ κάποιοι από τους συνεργαζόμενους φορείς είναι οι General Motors, Sun Microsystems και IBM. Αποτελεί μία open source υλοποίηση του ebXML που καθιστά δυνατή τη δημιουργία και ανταλλαγή λογισμικού. Ουσιαστικά, επιτρέπει στις επιχειρήσεις που συμμετέχουν σε αυτό να αναζητήσουν συνεργάτες, να επικοινωνήσουν και να συνεργαστούν στην ανταλλαγή ή την υλοποίηση εφαρμογών. Επίσης, διαθέτει βάση δεδομένων με διαμοιραζόμενη πληροφορία, υλοποίηση η οποία θα ελαττώσει σημαντικά το κόστος της ανταλλαγής εγγράφων και δεδομένων. Πρωταρχικό ρόλο εδώ παίζουν οι web services. Ήδη η προσπέλαση του Registry/Repository γίνεται μέσω τέτοιας υπηρεσίας, ενώ η ανάπτυξη περισσότερων υπηρεσιών βρίσκεται στα σχέδια των ιδρυόντων.

## **4.5 Σύγκριση με UDDI**

Η βασική διαφορά των δύο frameworks είναι ότι το ebXML διαθέτει repository εκτός από registry, ενώ το UDDI διαθέτει μόνο registry για την ανακάλυψη των υπηρεσιών. Παρακάτω θα εξετάσουμε τα κοινά στοιχεία αλλά και τις διαφορές τους [13].

### ***Κοινά στοιχεία***

- Δυνατότητα δημοσίευσης και ανακάλυψης υπηρεσιών (WSDL αρχείων)
- Δυνατότητα εκτέλεσης προκαθορισμένων επερωτήσεων
- Δυνατότητα ταξινόμησης των δεδομένων σε προκαθορισμένες κατηγορίες
- Δυνατότητα ειδοποίησης γεγονότων
- Δυνατότητα αυθεντικοποίησης μέσω ψηφιακής υπογραφής

### ***Διαφορές / Χαρακτηριστικά που διαθέτει το ebXML***

- Δυνατότητα αποθήκευσης υπηρεσιών (WSDL αρχείων) στο repository
- Δυνατότητα χειρισμού του κύκλου ζωής των WSDL αρχείων
- Δυνατότητα δημιουργίας συσχετίσεων ανάμεσα στα δεδομένα
- Δυνατότητα ομαδοποίησης αντικειμένων δημιουργώντας πακέτα
- Δυνατότητα content validation and cataloging
- Δυνατότητα εκτέλεσης ad hoc επερωτήσεων

- Δυνατότητα ειδοποίησης γεγονότων βάσει των αλλαγών του περιεχομένου

Παρατηρούμε ότι όλα τα κύρια χαρακτηριστικά του ebXML είναι αυτά που αποτελούν τα σημεία υπεροχής του απέναντι στο UDDI. Το μοντέλο πληροφορίας του είναι επεκτάσιμο και διαθέτει όλα τα χαρακτηριστικά ώστε να μπορεί να χρησιμοποιηθεί και να αυτοματοποιήσει πολλές επιχειρηματικές και μη δραστηριότητες.

## 5

# Graphical User Interface

## (GUI)

Σε αυτό το κεφάλαιο θα ασχοληθούμε με τη γραφική διεπαφή χρήστη που δημιουργήθηκε για να υποστηρίξει την κλήση υπηρεσιών διαδικτύου.

### 5.1 Τεχνολογίες που χρησιμοποιήθηκαν

#### 5.1.1 *Content Management System(CMS) / Wordpress*

Για την υλοποίηση του γραφικού περιβάλλοντος αλληλεπίδρασης με το χρήστη επιλέχθηκε η χρήση ενός Content Management System καθώς διαθέτει όλα τα χαρακτηριστικά εκείνα που ικανοποιούν τις ανάγκες και τις απαιτήσεις μιας τέτοιας εφαρμογής. Διαθέτει την ικανότητα ορισμού ρόλων για τους χρήστες (όπως απλός χρήστης ή διαχειριστής) και την ανάθεση συγκεκριμένων δικαιωμάτων και προνομίων σε κάθε ομάδα και διευκολύνει την αποθήκευση και ανάκτηση δεδομένων και εγγράφων.

Από τα CMS επιλέχθηκε το Wordpress λόγω του εύκολου χειρισμού του αλλά και των δυνατοτήτων που προσφέρει. Το κύριο εργαλείο του διαχειριστή είναι το Dashboard το οποίο αποτελεί τον πίνακα ελέγχου από όπου αλλάζουν οι ρυθμίσεις της λειτουργίας του. Υποστηρίζει εύκολη προσθήκη νέων σελίδων, εύκολη μετατροπή των ήδη υπαρχόντων αλλά και αλλαγή του κώδικα php και css των θεμάτων καθώς πρόκειται για open source CMS. Προσφέρει ποικιλία έτοιμων plug-ins για την αύξηση των δυνατοτήτων, η εγκατάσταση των οποίων είναι αυτοματοποιημένη. Για παράδειγμα, στο συγκεκριμένο GUI χρησιμοποιήθηκε το Register Plus[14] plugin, σκοπός του οποίου είναι η ενίσχυση της φόρμας εγγραφής χρήστη προσθέτοντας νέα πεδία. Επίσης, υποστηρίζει τη λειτουργία blog και επιτρέπει την

άμεση πρόσβαση στους λογαριασμούς των χρηστών αλλά και δυνατότητα γνώσης και παρακολούθησης των πρόσφατων ενεργειών τους.

### **5.1.2 *Hyper Text Markup Language (HTML)***

Πρόκειται για την κυρίαρχη mark up γλώσσα του διαδικτύου η οποία χρησιμοποιείται για τη δημιουργία ιστοσελίδων. Στο εσωτερικό της μπορούν να ενσωματωθούν κείμενα, εικόνες αλλά και φόρμες. Επίσης, ενσωματώνοντας κώδικα γραμμένο σε CSS μπορεί να βελτιώσει τη μορφή των ιστοσελίδων που παράγει, ενώ σε συνδυασμό με τη Javascript<sup>5</sup> δημιουργεί διαδραστικές ιστοσελίδες οι οποίες αλληλεπιδρούν με το χρήστη.

### **5.1.3 *Cascading Style Sheets (CSS)***

Πρόκειται για μία style sheet γλώσσα που χρησιμοποιείται για να καθορίσει τη μορφή εγγράφων γραμμένων σε mark up γλώσσες όπως είναι η XML και η HTML. Συγκεκριμένα, μπορεί να ορίσει το χρώμα της σελίδας αλλά τα χαρακτηριστικά των επιμέρους στοιχείων της, όπως το μέγεθος και τη θέση τους.

### **5.1.4 *JQuery***

Πρόκειται για μία ελεύθερη και open source Javascript βιβλιοθήκη[15] η οποία δημιουργήθηκε για να διευκολύνει τη συγγραφή Javascript κώδικα και κυκλοφόρησε το 2006. Περιέχει ήδη υλοποιημένες συναρτήσεις Javascript και η χρήση της διευκολύνει και μειώνει τον απαιτούμενο χρόνο δημιουργίας προγραμμάτων. Οι δυνατότητες της αξιοποιήθηκαν από πολλούς προγραμματιστές και οδήγησαν στη δημιουργία διάφορων plug-ins<sup>6</sup> τα οποία αυτοματοποιούν πολλές χρονοβόρες και περίπλοκες διαδικασίες. Στη συγκεκριμένη υλοποίηση χρησιμοποιήθηκαν δύο plug-ins:

---

<sup>5</sup> Αποτελεί την κυρίαρχη scripting γλώσσα του διαδικτύου

<sup>6</sup> Πρόκειται για τμήματα λογισμικού τα οποία προσδίδουν συγκεκριμένες ιδιότητες στο λογισμικό στο οποίο ενσωματώνονται.

### ***Simple Modal Box (by Eric Martin [16])***

Πρόκειται για παράθυρο που εμφανίζεται ξαφνικά μόλις ο χρήστης εκτελέσει μια συγκεκριμένη ενέργεια (ονομάζονται και popup windows). Εμφανίζεται στην κεντρική οθόνη και σταματά την αλληλεπίδραση του χρήστη με την αρχική σελίδα. Χρησιμοποιείται σε διάφορες περιπτώσεις είτε για την απλή εμφάνιση πληροφοριών είτε για τη συμπλήρωση απαραίτητων στοιχείων του χρήστη (όπως όνομα χρήστη και κωδικός πρόσβασης) αλλά ακόμα και για την επιβεβαίωση πραγματοποίησης κάποιας ενέργειας (confirm box). Στη συγκεκριμένη υλοποίηση, χρησιμοποιήθηκε για την παρουσίαση απαιτούμενης πληροφορίας και συγκεκριμένα για την εμφάνιση της περιγραφής Lapack ρουτίνων και αρχείων WSDL.

### ***Tabbed Content Area [17]***

Πρόκειται για λειτουργία η οποία επιτρέπει στον προγραμματιστή να τοποθετήσει σε μία περιοχή της ιστοσελίδας περισσότερη πληροφορία από αυτή που θα μπορούσε αρχικά. Κάθε στιγμή ένα τμήμα αυτής της πληροφορίας είναι φανερό στο χρήστη, ενώ ο χρήστης είναι αυτός που επιλέγει τι θα βλέπει εκείνη τη στιγμή. Στη συγκεκριμένη υλοποίηση, χρησιμοποιήθηκε για την ενσωμάτωση δύο φορμών στην αρχική σελίδα με σκοπό τον καθορισμό επιλογών σχετικών με την είσοδο και την έξοδο αλλά και για την παροχή οδηγιών στο χρήστη σχετικά με την κλήση της υπηρεσίας.

## **5.2 Περιγραφή διεπαφής**

### **5.2.1 Απαιτήσεις**

Σκοπός της συγκεκριμένης διεπαφής είναι η κλήση υπηρεσιών μέσω αυτής. Η πραγματοποίηση της κλήσης προϋποθέτει μία σειρά από επιλογές που πρέπει να γίνουν.

#### ***Επιλογή της κατάλληλης ρουτίνας***

Υπάρχει ένα σύνολο από χαρακτηριστικά που πρέπει να καθοριστούν ώστε ο χρήστης να επιλέξει την κατάλληλη ρουτίνα επίλυσης γραμμικού συστήματος  $AX = B$ . Καθορίζοντας:

- Είδος αριθμητικής (πραγματικούς ή μιγαδικούς αριθμούς)
- Είδος ακρίβειας (απλή ή διπλή)

- Ιδιότητα του πίνακα
- Μορφή του πίνακα

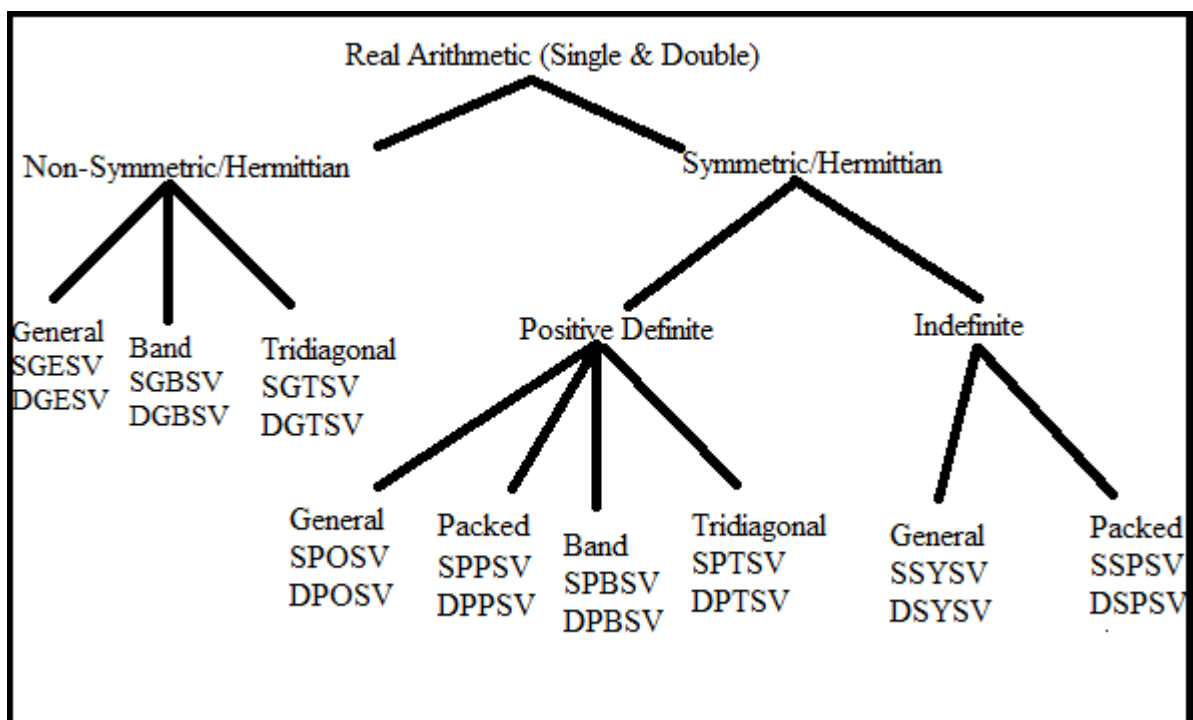
πρέπει να υπολογίζεται αυτόματα βάσει των τρεχουσών επιλογών του ποια είναι η αντίστοιχη ρουτίνα. Επίσης, πρέπει κάθε στιγμή να έχει τη δυνατότητα να δει και να διαβάσει την περιγραφή της ρουτίνας, που περιλαμβάνει τη λειτουργία της και περιγραφή των παραμέτρων της. Η κατηγοριοποίηση των ρουτινών που υποστηρίζονται παρουσιάζεται στις ακόλουθες εικόνες.

### ***Επιλογή του μέρους που θα τρέξει η υπηρεσία***

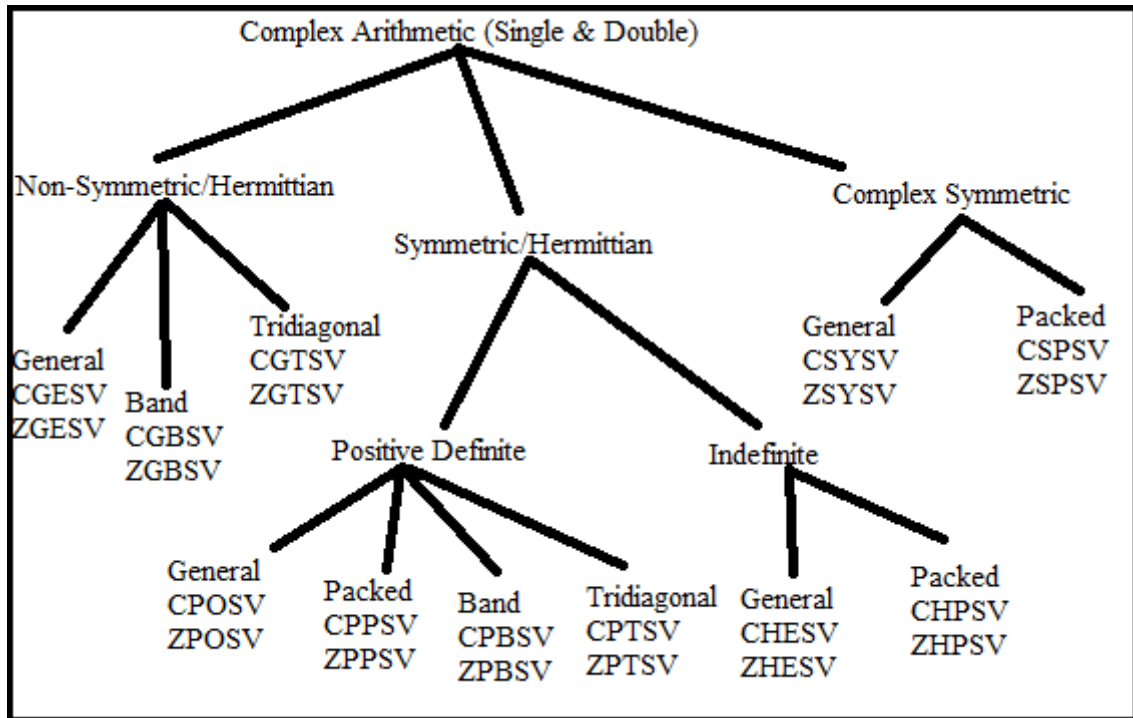
Έχοντας επιλέξει ο χρήστης ποια ρουτίνα ικανοποιεί τις ανάγκες του, ουσιαστικά έχει επιλέξει ποια υπηρεσία θέλει να καλέσει. Πλέον, η επόμενη επιλογή που πρέπει να γίνει είναι το πού θα τρέξει η υπηρεσία. Ο χρήστης έχει να επιλέξει ένα από τα ακόλουθα:

- Amazon
- Cluster
- Στον υπολογιστή του

Να σημειωθεί ότι από τις παραπάνω επιλογές, η πρώτη απαιτεί πληρωμή, ενώ η τελευταία κωδικούς πρόσβασης για τον υπολογιστή του χρήστη.



Εικόνα 5 - 1 Οι ρουτίνες πραγματικών αριθμών που υποστηρίζονται από τη διεπαφή



Εικόνα 5 - 2 Οι ρουτίνες μιγαδικών αριθμών που υποστηρίζονται από τη διεπαφή

### Καθορισμός εισόδου

Πλέον πρέπει να καθοριστούν οι παράμετροι κλήσης της υπηρεσίας, συνήθως οι πίνακες A και B και οι διαστάσεις τους, ενώ κατά περίπτωση απαιτούνται και άλλες παράμετροι . Δίνονται οι εξής επιλογές:

- **Να δοθούν οι πίνακες πληκτρολογώντας τις τιμές των στοιχείων τους**  
Σε αυτή την περίπτωση, η εφαρμογή πρέπει να θυμάται την επιλογή ρουτίνας που πραγματοποίησε ο χρήστης και να εμφανίζει την αντίστοιχη φόρμα. Μόλις ο χρήστης συμπληρώσει όλα τα πεδία η υπηρεσία καλείται.
- **Να δοθεί κώδικας ο οποίος παράγει τους πίνακες**  
Σε αυτή την περίπτωση, εμφανίζεται η αντίστοιχη φόρμα στην οποία δίνονται οι οδηγίες για τη δομή που πρέπει να έχει ο κώδικας αλλά και πεδίο συμπλήρωσης του.
- **Να δοθεί αρχείο το οποίο περιέχει τον πίνακα**  
Σε αυτή την περίπτωση, εμφανίζεται η αντίστοιχη φόρμα για το uploading του πίνακα.
- **Να δοθεί URL στο οποίο βρίσκεται ο πίνακας**  
Σε αυτή την περίπτωση, εμφανίζεται το αντίστοιχο πεδίο συμπλήρωσης της διεύθυνσης URL.

### ***Καθορισμός εξόδου***

Πρέπει να καθοριστεί η ποσότητα της πληροφορίας που επιθυμεί ο χρήστης να λάβει στην έξοδο. Διακρίνονται 3 επίπεδα:

- **Low (Χαμηλή)**  
Επιστρέφεται μόνο το αποτέλεσμα.
- **Medium (Ενδιάμεση)**  
Επιστρέφεται το αποτέλεσμα και κάποια στοιχεία του υπολογισμού του.
- **High (Υψηλή)**  
Επιστρέφεται λεπτομερής αναφορά υπολογισμού του αποτελέσματος. Περιλαμβάνει πληροφορίες όπως το ίδιο το αποτέλεσμα, ο χρόνος υπολογισμού του, η ακρίβεια αποτελέσματος και η cpu που απαιτήθηκε.

### ***Κλήση υπηρεσίας από τον ίδιο το χρήστη***

Εκτός από την επιλογή της εκτέλεσης μιας υπηρεσίας μέσω της διεπαφής, ο χρήστης πρέπει να έχει τη δυνατότητα να την καλέσει ο ίδιος. Σε αυτή την περίπτωση, διακρίνονται δύο επιλογές:

- Ο χρήστης αναπτύσσει μόνος του την εφαρμογή πελάτη που θα καλέσει την υπηρεσία και αφού ορίσει την πλατφόρμα χρησιμοποιεί επιλέγοντας από τις Net Beans, Visual Studio ή Eclipse, του εμφανίζονται οι αντίστοιχες οδηγίες.
- Ο χρήστης θέλει να καλέσει την υπηρεσία μέσα από δικό του κώδικα. Ορίζοντας τη γλώσσα προγραμματισμού στην οποία αναπτύσσει τον κώδικα, η διεπαφή του παρέχει την αντίστοιχη βιβλιοθήκη και τον απαραίτητο κώδικα που πρέπει να προστεθεί στο δικό του για να γίνει η κλήση. Σε αυτή την περίπτωση, η διεπαφή πρέπει να θυμάται την αρχική επιλογή ρουτίνας του χρήστη και να εμφανίσει την κατάλληλη βιβλιοθήκη και τον κατάλληλο κώδικα.

### ***Δημιουργία υπηρεσίας από τον ίδιο το χρήστη***

Εκτός από τη δυνατότητα κλήσης μιας ήδη υλοποιημένης υπηρεσίας, ο χρήστης πρέπει να έχει τη δυνατότητα δημιουργίας μια υπηρεσίας που καλεί μία ρουτίνα Lapack η οποία δεν προσφέρεται ήδη. Σε αυτή την περίπτωση, προσφέρονται οι οδηγίες για υλοποίηση σε Net Beans IDE.



## 5.2.2 Περιβάλλον χρήσης

Για την υλοποίηση των λειτουργιών που περιγράφηκαν νωρίτερα, χρησιμοποιήθηκαν φόρμες HTML σε συνδυασμό με JQuery plug-ins. Αυτές οι φόρμες ενσωματώθηκαν στις σελίδες του Wordpress μέσω του tag <iframe>, το οποίο εξυπηρετεί αυτό ακριβώς το σκοπό, δηλαδή την ενσωμάτωση ενός html εγγράφου μέσα σε ένα άλλο html έγγραφο.

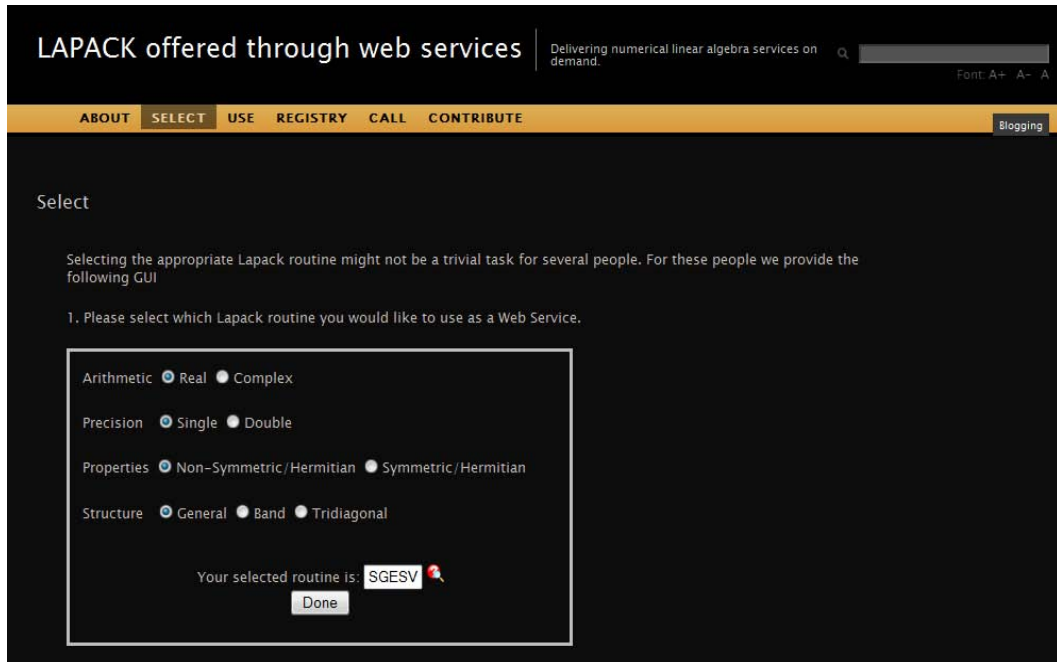
Οι λειτουργίες στις οποίες εξυπηρετεί η διεπαφή ταξινομήθηκαν σε διάφορες σελίδες ως εξής:

### **Σελίδα SELECT**

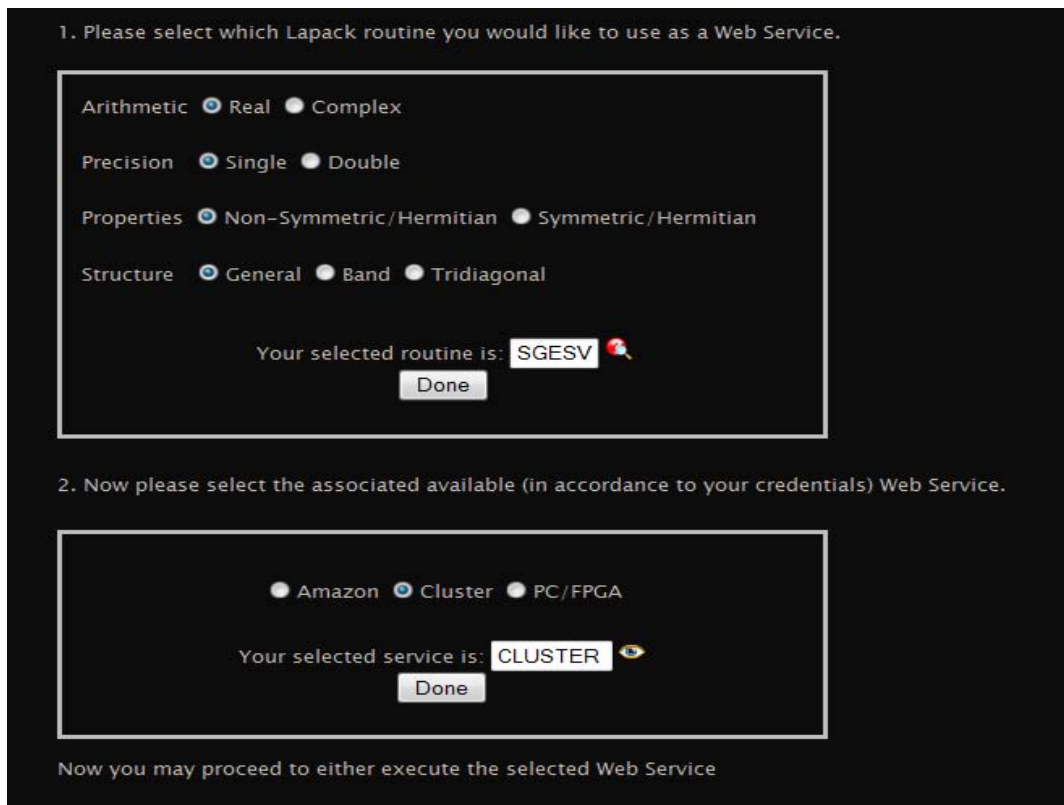
Εδώ υπάρχει η φόρμα επιλογής ρουτίνας που περιέχει radio buttons για την επιλογή των χαρακτηριστικών και ένα πεδίο όπου αναγράφεται το όνομα της αντίστοιχης ρουτίνας. Η προεπιλεγμένη ρουτίνα, σύμφωνα με τα προεπιλεγμένα χαρακτηριστικά είναι η SGESV. Κάθε φορά που κάποιο χαρακτηριστικό αλλάζει, αλλάζει και το όνομα της ρουτίνας. Μόλις πατηθεί το εικονίδιο δίπλα από το πεδίο, εμφανίζεται ένα popup window με την περιγραφή της ρουτίνας που αναγράφεται στο πεδίο. Ο χρήστης μπορεί να αλλάξει τα χαρακτηριστικά όσες φορές επιθυμεί, μέχρι να εντοπίσει τη ρουτίνα που ικανοποιεί τις ανάγκες του. Τότε, για να επικυρώσει την επιλογή του πρέπει να πατήσει το κουμπί Done. Το πάτημα του συγκεκριμένου κουμπιού έχει 2 επακόλουθα:

- Εμφανίζεται η φόρμα καθορισμού του μέρους όπου θα τρέξει η υπηρεσία.
- Δημιουργεί ένα session cookie με τιμή το όνομα της ρουτίνας. Η χρήση των cookies στη διεπαφή είναι απαραίτητη καθώς οι φόρμες που δημιουργήθηκαν για την υλοποίηση των λειτουργιών ανήκουν σε διαφορετικά HTML αρχεία και δεν μπορούν να επικοινωνήσουν. Όμως, για την ορθή λειτουργία, είναι απαραίτητο για τις φόρμες να γνωρίζουν την αρχική επιλογή του χρήστη. Η χρήση των cookies αποτελεί τη λύση του προβλήματος καθώς οι φόρμες διαβάζουν την τιμή του cookie και διαμορφώνουν τις ενέργειές τους ανάλογα.

Στη συνέχεια, στη φόρμα επιλογής του μέρους όπου θα τρέξει η υπηρεσία, ο χρήστης έχει τη δυνατότητα να αλλάξει την προεπιλεγμένη τιμή που είναι «Cluster», ενώ πατώντας το εικονίδιο δίπλα από το πεδίο μπορεί να δει το αρχείο WSDL της επιλεγμένης υπηρεσίας. Σε κάθε μία από τις 3 επιλογές του, Amazon, Cluster και PC αφήνοντας τον κέρσορα πάνω εμφανίζεται ένα πεδίο με την εξήγηση της συγκεκριμένης επιλογής. Μόλις πραγματοποιήσει την επιλογή, πρέπει να πατήσει το κουμπί Done το οποίο επικυρώνει την επιλογή του.



Εικόνα 5 - 3 Η φόρμα επιλογής ρουτίνας



Εικόνα 5 - 4 Οι φόρμες της σελίδας SELECT

### Σελίδα USE

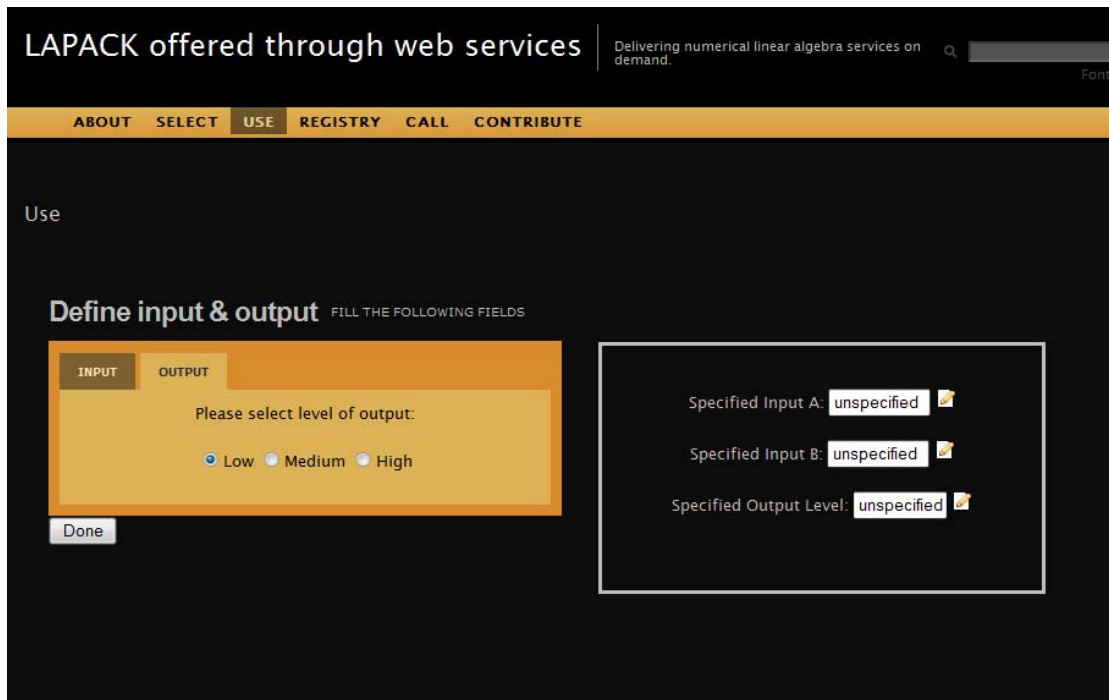
Σε αυτή τη σελίδα υπάρχουν τα τελευταία πεδία που πρέπει να συμπληρωθούν για την κλήση της υπηρεσίας. Πρόκειται για τις φόρμες καθορισμού εισόδου και εξόδου τα οποία

συνδυάζονται μαζί σε ένα tabbed content area plug-in. Στη φόρμα ορισμού της εισόδου (tab = INPUT) ο χρήστης πρέπει να συμπληρώσει στο πεδίο n την τάξη του πίνακα A και στο πεδίο nrhs τον αριθμό στηλών του πίνακα B (προεπιλεγμένη τιμή είναι το 1, αλλά έχει τη δυνατότητα να την αλλάξει). Στη συνέχεια, στη φόρμα ορισμού εξόδου (tab = OUTPUT) πρέπει να επιλέξει το είδος της εξόδου που επιθυμεί. Και στα 2 tabs, αφήνοντας τον κέρσορα πάνω σε κάθε μία από τις λέξεις

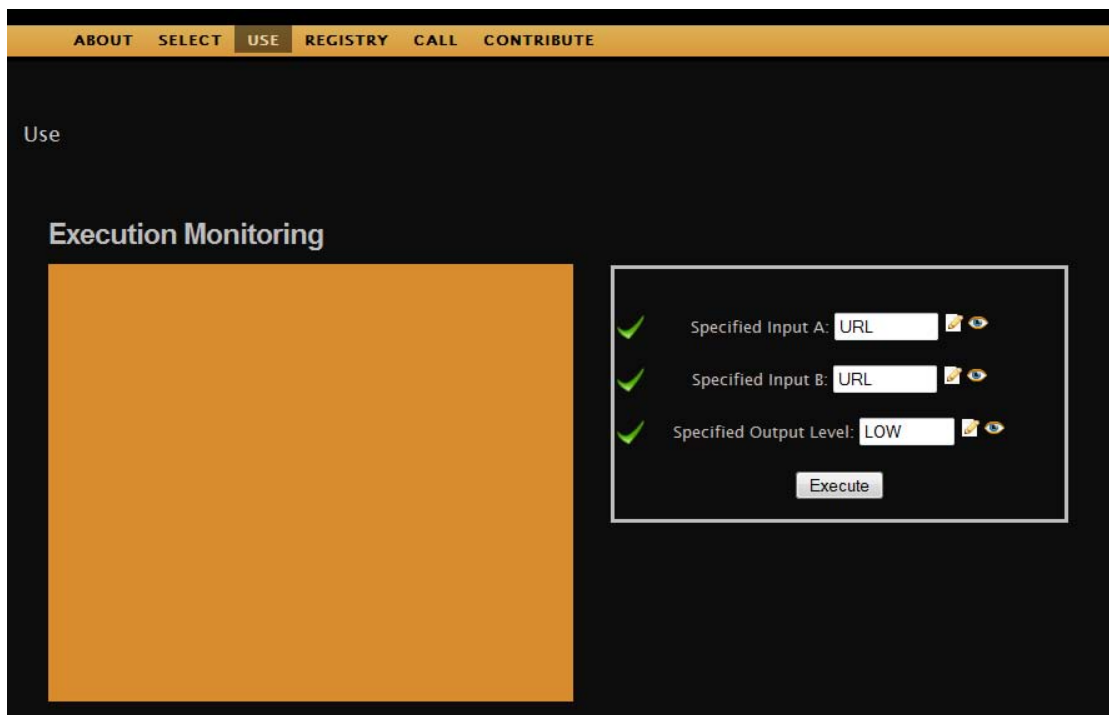
- Keyboard, Code, Uploading file, URL
- Low, Medium, High

εμφανίζεται ένα πλαίσιο πάνω από τη λέξη με την επεξήγηση της λειτουργίας το οποίο εξαφανίζεται μόλις ο κέρσορας μετακινηθεί. Όταν η επιλογή ολοκληρωθεί και πατηθεί το κουμπί Done, οι επιλογές επικυρώνονται, αντιγράφονται στη δεξιά φόρμα και το κουμπί Execute εμφανίζεται. Μόλις αυτό το κουμπί πατηθεί, καλείται η υπηρεσία και εξαφανίζεται η tabbed content area, ενώ στη θέση της εμφανίζεται μία φόρμα όπου θα καταγράφονται τα στάδια της εκτέλεσης. Επίσης, πλέον η δεξιά φόρμα διαθέτει εικονίδια Edit, τα οποία σταματάνε την εκτέλεση και επαναφέρουν τις φόρμες για επανακαθορισμό εισόδου – εξόδου αλλά και εικονίδια που εμφανίζουν popup παράθυρα και καταγράφουν τις λεπτομέρειες για την κλήση της υπηρεσίας.

Εικόνα 5 - 5 Σελίδα USE – Φόρμα καθορισμού εισόδου



Εικόνα 5 - 6 Σελίδα USE – Φόρμα καθορισμού εξόδου



Εικόνα 5 - 7 Σελίδα USE – Φόρμα παρακολούθησης εκτέλεσης

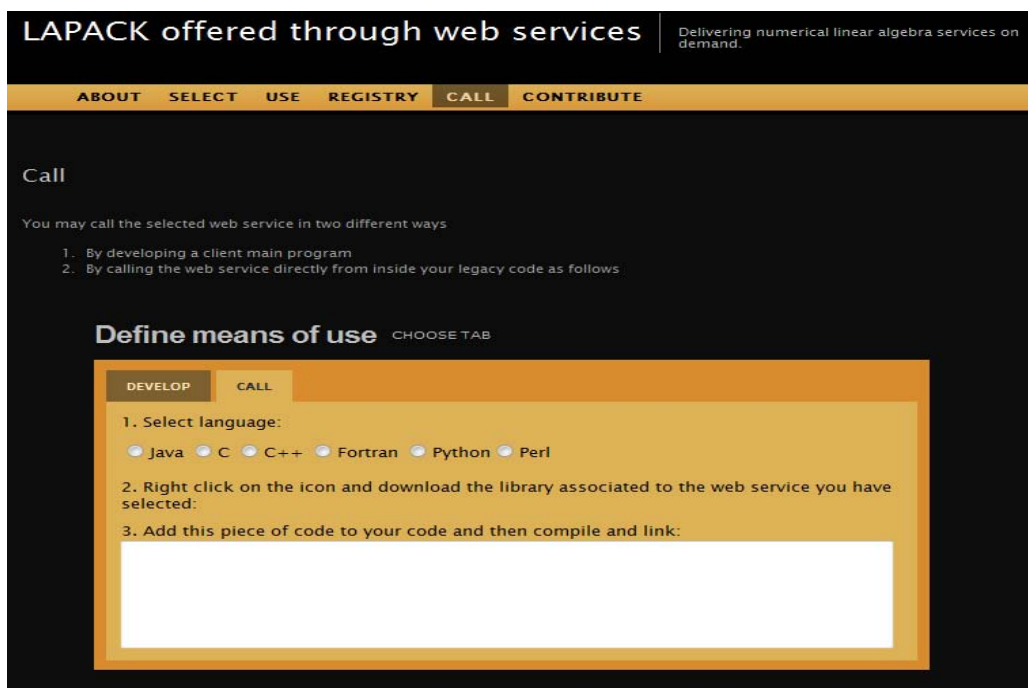
## Σελίδα CALL

Σε αυτή τη σελίδα ο χρήστης έχει τη δυνατότητα να επιλέξει ανάμεσα στο αν θα δημιουργήσει πελάτη που θα καλεί την υπηρεσία (tab= DEVELOP) ή αν σκοπεύει να ενσωματώσει την κλήση της υπηρεσίας σε κώδικα που έχει γράψει ο ίδιος (tab = CALL). Για την ενσωμάτωση αυτών των λειτουργιών σε μία φόρμα χρησιμοποιήθηκε πάλι το tabbed content area plugin.

Στην περίπτωση της δημιουργίας πελάτη, ο χρήστης πρέπει απλά να επιλέξει μέσω radio buttons την develop platform που χρησιμοποιεί, οπότε και εμφανίζονται οι αντίστοιχες οδηγίες.

Στην περίπτωση της κλήσης υπηρεσίας μέσα από κώδικα, πρέπει πρώτα να επιλέξει τη γλώσσα προγραμματισμού στην οποία είναι γραμμένος. Ουσιαστικά, η επιλογή της γλώσσας προκαλεί την ανάγνωση της τιμής του cookie ώστε να γίνει γνωστή η επιλογή του χρήστη. Κατά την επιλογή της γλώσσας διακρίνονται 3 περιπτώσεις:

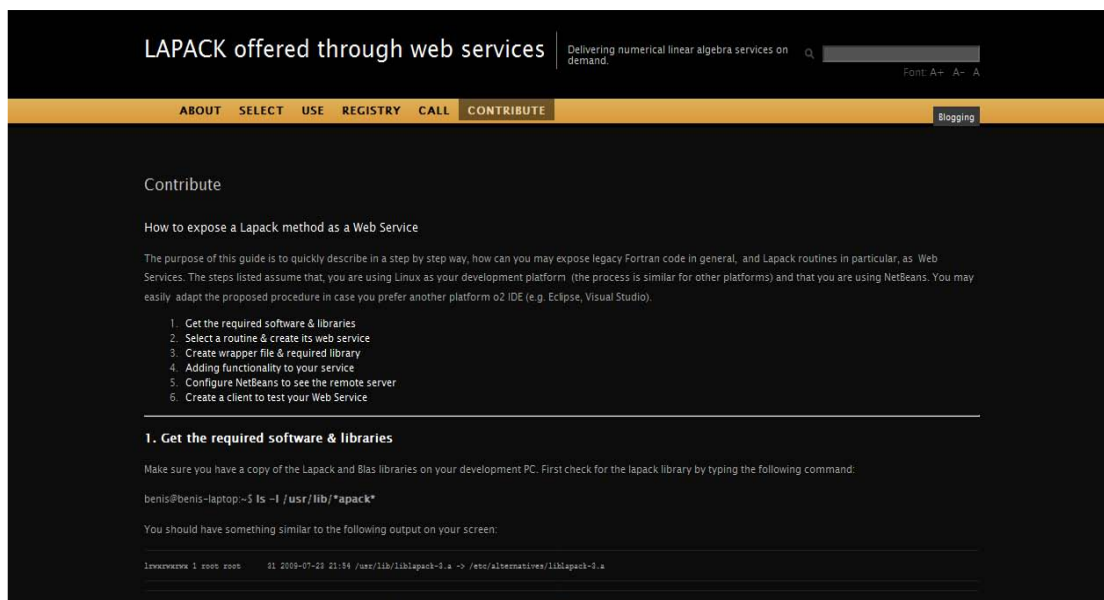
- Αν η γλώσσα που επιλέχθηκε, δεν υποστηρίζεται ακόμα από τη διεπαφή εμφανίζεται παράθυρο ειδοποίησης με σχετικό μήνυμα.
- Αν η γλώσσα που επιλέχθηκε υποστηρίζεται και η επιλεγμένη υπηρεσία έχει υλοποιηθεί, τότε εμφανίζεται το εικονίδιο download της βιβλιοθήκης και το ανάλογο τμήμα κώδικα.
- Αν η γλώσσα που επιλέχθηκε υποστηρίζεται αλλά η επιλεγμένη υπηρεσία δεν έχει υλοποιηθεί, τότε εμφανίζεται παράθυρο ειδοποίησης με σχετικό μήνυμα.



Εικόνα 5 - 8 Σελίδα CALL

## Σελίδα CONTRIBUTE

Σε αυτή τη σελίδα υπάρχουν οι οδηγίες υλοποίησης μιας υπηρεσίας χωρισμένες σε στάδια. Στην αρχή της σελίδας, υπάρχει ευρετήριο με τα βήματα που ουσιαστικά αποτελούν τα εσωτερικά bookmarks της σελίδας.



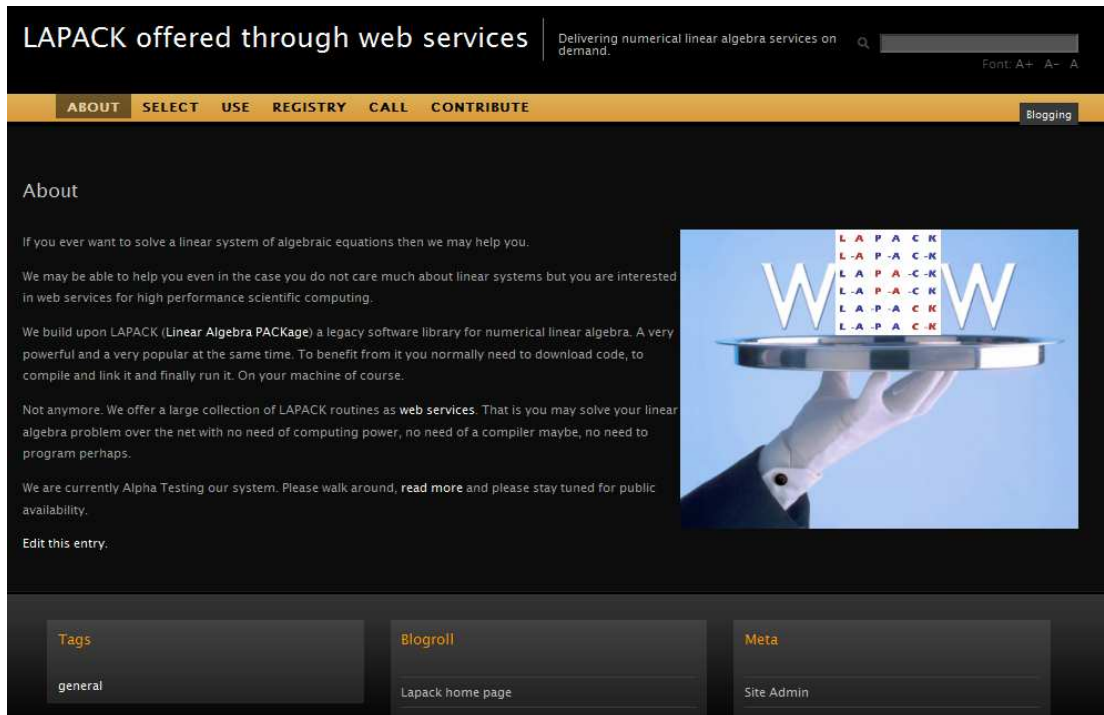
Εικόνα 5 - 9 Σελίδα Contribute

## 5.3 Σενάριο χρήσης

Έστω πως ο χρήστης επιθυμεί να λύσει γραμμικό σύστημα της μορφής  $AX = B$  όπου ο  $A$  είναι πραγματικός τριδιαγώνιος πίνακας με στοιχεία διπλής ακρίβειας. Οι πίνακες  $A$  και  $B$  είναι οι:

$$A = \begin{bmatrix} 3.0 & 2.1 & 0 & 0 & 0 \\ 3.4 & 2.3 & -1.0 & 0 & 0 \\ 0 & 3.6 & -5.0 & 1.9 & 0 \\ 0 & 0 & 7.0 & -0.9 & 8.0 \\ 0 & 0 & 0 & -6.0 & 7.1 \end{bmatrix} \quad B = \begin{bmatrix} 2.7 \\ -0.5 \\ 2.6 \\ 0.6 \\ 2.7 \end{bmatrix}$$

Πληκτρολογεί τη διεύθυνση <http://lapack.ws> και στον browser εμφανίζεται η αρχική σελίδα.



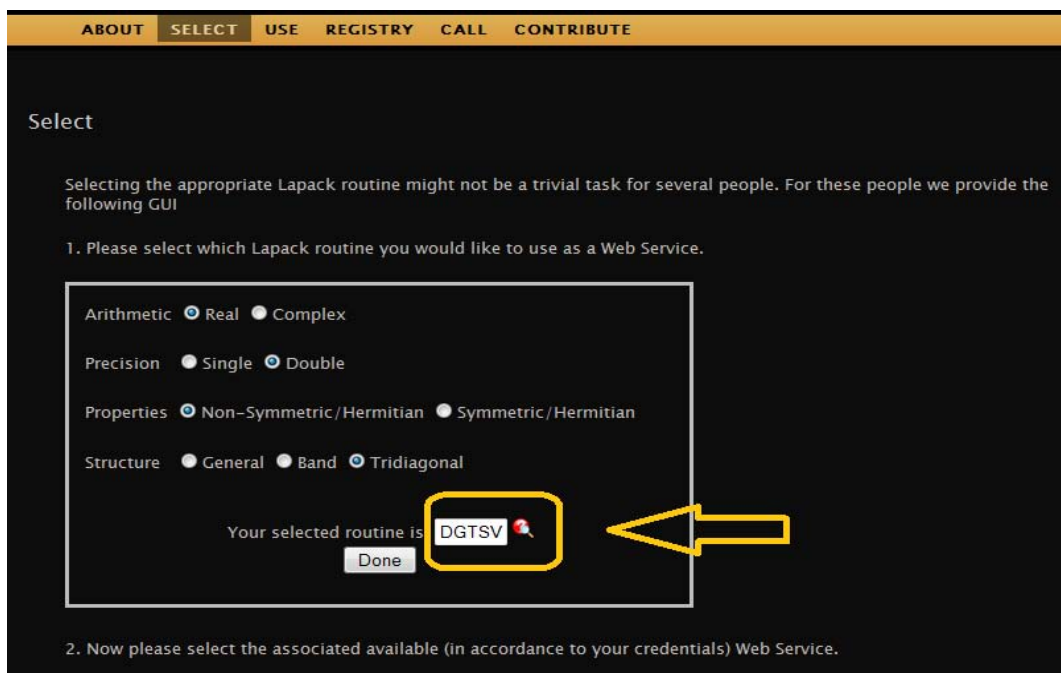
Εικόνα 5 - 10 Home Page

Επιλέγει το tab Select και ανοίγει η σελίδα Select όπου βλέπει τη φόρμα επιλογής.

Αλλάζει τα χαρακτηριστικά του πίνακα ως εξής:

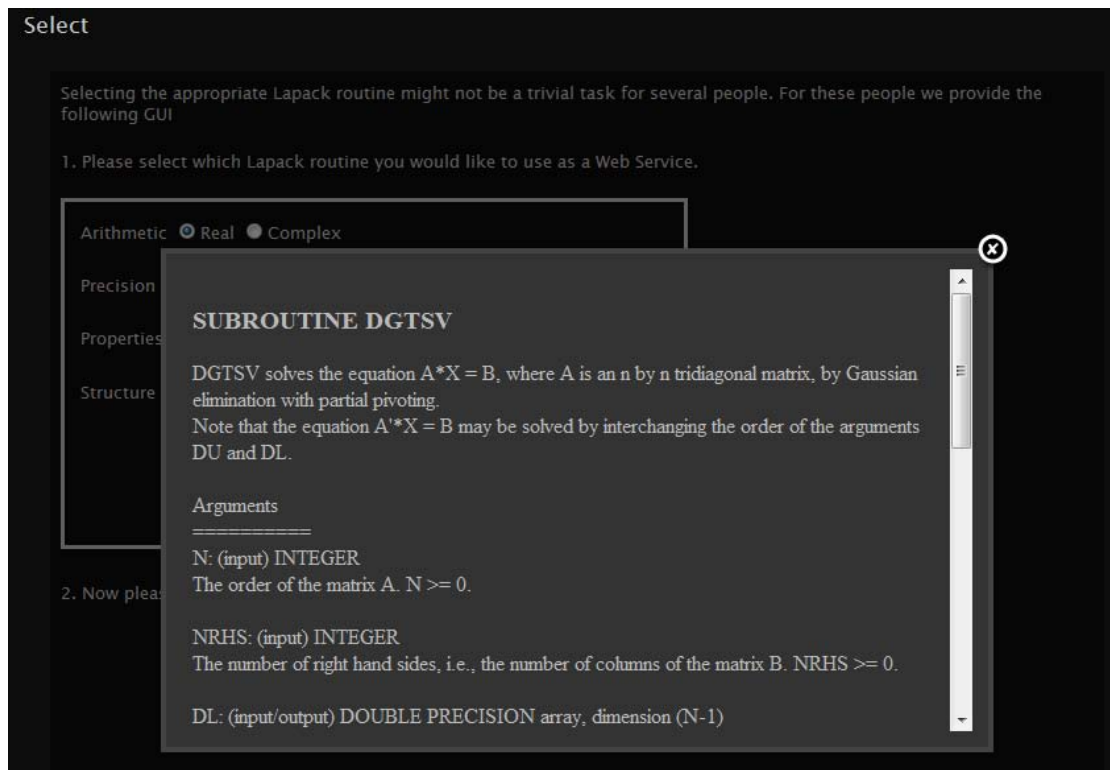
- Την αριθμητική ακρίβεια σε διπλή: Precision  $\longrightarrow$  Double
- Τη δομή του πίνακα σε τριδιαγώνιο: Structure  $\longrightarrow$  Tridiagonal

Πλέον το πεδίο της επιλεγμένης ρουτίνας γράφει : DGTSV .



Εικόνα 5 - 11 Ο χρήστης έχει επιλέξει τη ρουτίνα DGTSV

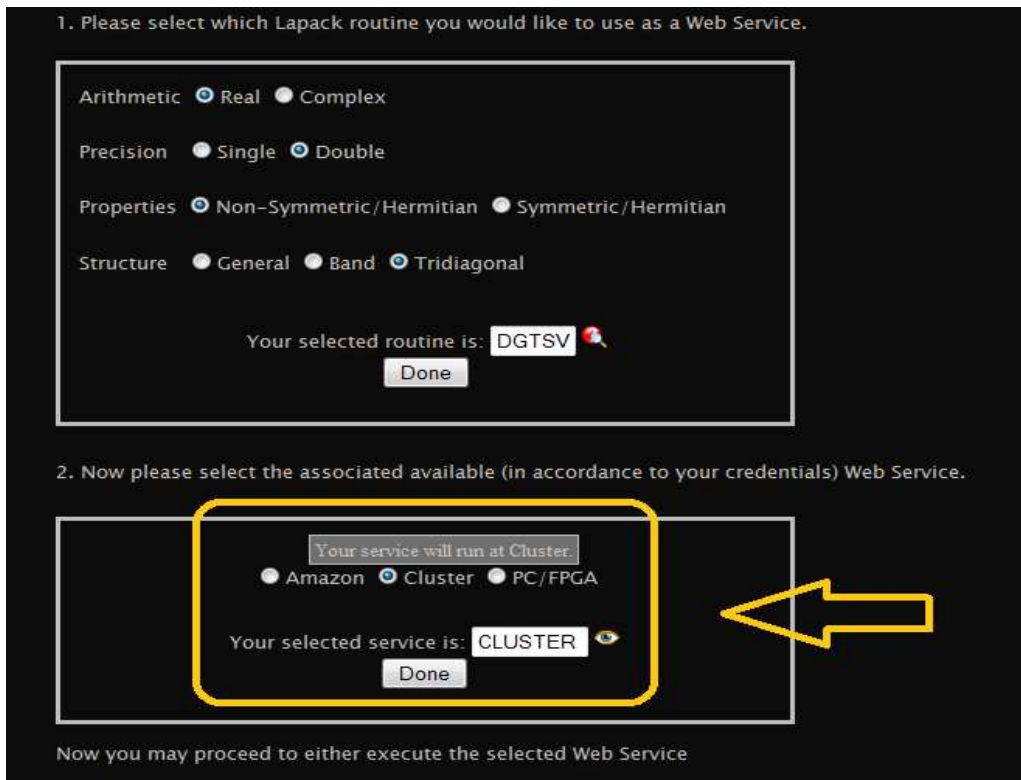
Για να δει την περιγραφή της πατάει το εικονίδιο που βρίσκεται δεξιά του πεδίου και εμφανίζεται ένα popup παράθυρο με την περιγραφή της ρουτίνας [18] .



Εικόνα 5 - 12 Ο χρήστης διαβάσει την περιγραφή της ρουτίνας στο popup παράθυρο

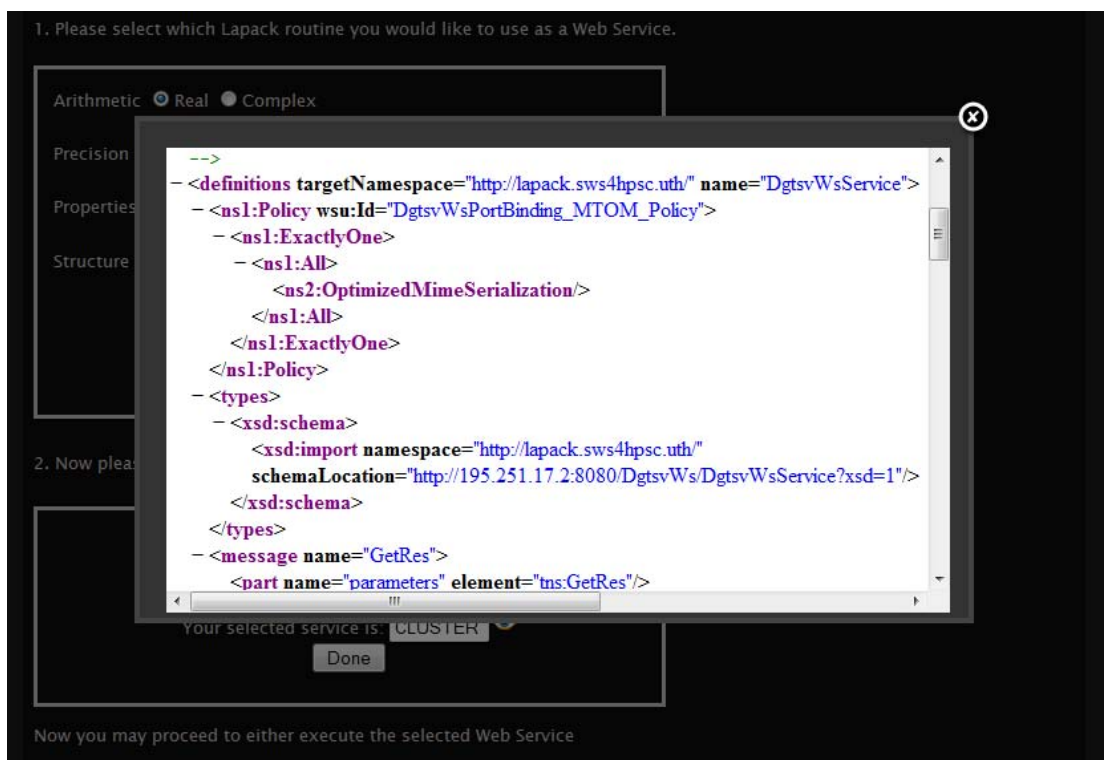
Αφού διαβάσει την περιγραφή, κλείνει το παράθυρο και επικυρώνει την επιλογή του πατώντας Done. Τότε εμφανίζεται η φόρμα καθορισμού του μέρους που θα τρέξει η υπηρεσία και υπάρχει ως προεπιλεγμένη επιλογή το Cluster. Αφήνοντας τον κέρσορα πάνω στη λέξη «Cluster» εμφανίζεται μέσα σε πλαίσιο η περιγραφή της επιλογής.





Εικόνα 5 - 13 Ο χρήστης διαβάζει την περιγραφή της επιλογής Cluster

Πατώντας το εικονίδιο που βρίσκεται δεξιά του πεδίου (eye icon), εμφανίζεται σε pop-up παράθυρο το WSDL αρχείο [19] της υπηρεσίας.



Εικόνα 5 - 14 Ο χρήστης διαβάζει το αρχείο WSDL της υπηρεσίας

Κλείνει το παράθυρο και αφού έχουν καθοριστεί οι επιλογές του, μεταβαίνει στη σελίδα Use για να καλέσει την υπηρεσία.

Εικόνα 5 - 15 Η φόρμα επιλογής εισόδου της σελίδας USE

Επιλέγει να δώσει τα στοιχεία των πινάκων μέσω πληκτρολογίου, οπότε πατώντας την επιλογή Keyboard (του πίνακα A ή του B), εμφανίζεται η φόρμα συμπλήρωσης των παραμέτρων. (Εικόνα 5 - 16)

Ο χρήστης συμπληρώνει τα πεδία (Εικόνα 5 - 17) και πατώντας το κουμπί Submit καλείται η υπηρεσία και επιστρέφεται το αποτέλεσμα. (Εικόνα 5 - 18)

(Να σημειωθεί ότι η υλοποίηση της υπηρεσίας DgtsvWs αλλά και του DemoApplication έχει γίνει από το Δημήτρη Μπένη)

## Dgtsv Demo Application

This application can be used to solve the equation  $A * X = B$ , where  $A$  is an  $n$  by  $n$  tridiagonal matrix, by Gaussian elimination, with partial pivoting.

It is just a demo, so  $n$  can be maximum 10 and nrhs 1.

n :

D :

DL:

DU:

B :

**NOTE: Use space between consecutive array entries!**

Εικόνα 5 - 16 Η φόρμα συμπλήρωσης παραμέτρων για την κλήση της DGTSV

## Dgtsv Demo Application

This application can be used to solve the equation  $A * X = B$ , where  $A$  is an  $n$  by  $n$  tridiagonal matrix, by Gaussian elimination, with partial pivoting.

It is just a demo, so  $n$  can be maximum 10 and nrhs 1.

n:

D:

DL:

DU:

B:

**NOTE: Use space between consecutive array entries!**

Εικόνα 5 - 17 Ο χρήστης συμπληρώνει τα πεδία των παραμέτρων



Εικόνα 5 - 18 Η υπηρεσία επιστρέφει το αποτέλεσμα

## Σύνοψη και μελλοντικές προοπτικές

Αρχικά, περιγράφηκε η βιβλιοθήκη της Lapack και μελετήθηκαν τα προβλήματα που προκύπτουν κατά τη χρήση της. Αναφερθήκαμε στην έννοια του επιστημονικού υπολογισμού και τις απαιτήσεις του σε υπολογιστική ισχύ. Στη συνέχεια, ασχοληθήκαμε με την εξέλιξη των διαδικτυακών εφαρμογών και περιγράψαμε τα συστατικά των υπηρεσιών διαδικτύου όπως και τη διαδικασία με την οποία μπορεί μία υπηρεσία διαδικτύου υλοποιημένη σε γλώσσα προγραμματισμού Java να καλεί ρουτίνα γραμμένη σε Fortran. Μελετήσαμε τα χαρακτηριστικά του ebXML Registry το οποίο και χρησιμοποιήθηκε για τη δήλωση των υπηρεσιών μας και αναλύσαμε τις δυνατότητες που προσφέρει η χρήση του.

Τέλος, υλοποιήσαμε μία γραφική διεπαφή χρήστη για την κλήση των υπηρεσιών που δημιουργήσαμε η οποία του προσφέρει επίσης τη δυνατότητα να ανακαλύψει την κατάλληλη για τους υπολογισμούς του ρουτίνα.

Υπάρχουν πολλές προοπτικές εξέλιξης και σημεία που επιδέχονται βελτίωσης. Παρακάτω αναφέρονται τομείς που πρέπει να βελτιωθούν.

1. Υλοποίηση περισσότερων υπηρεσιών της Lapack.
2. Υλοποίηση βοηθητικών υπηρεσιών της Blas.
3. Συνδυασμός υπηρεσιών για τη δημιουργία ροών εργασιών με τη χρήση της WS-BPEL.
4. Βελτίωση του GUI.
5. Οργάνωση των μεταδεδομένων στο ebXML Registry.
6. Δυνατότητα πρόβλεψης του χρόνου ολοκλήρωσης ενός υπολογισμού.

7. Εγκατάσταση των υπηρεσιών και σε άλλα υψηλής απόδοσης συστήματα όπως το Hellas Grid<sup>7</sup>

---

<sup>7</sup> Υποδομή που περιλαμβάνει 6 clusters με 90TB αποθηκευτικό χώρο και 768 CPUs και έχει πανελλαδική εμβέλεια.( <http://www.hellasgrid.gr/infrastructure/index.php?language=el> )

1. [http://en.wikipedia.org/wiki/High-performance\\_computing](http://en.wikipedia.org/wiki/High-performance_computing)
2. <http://www.mathcom.com/corpdire/techinfo.mdir/q115.html#q115.1>
3. “Web Services: Principles & Technology”, Michael P. Papazoglou
4. <http://en.wikipedia.org/wiki/UDDI>
5. <http://www.cs.colorado.edu/~jessup/lapack/individual.html>
6. “Developers Guide: How to expose a Lapack Method as a Web Service”, Δημήτρης Μπένης
7. <http://www.webservicesarchitect.com/content/articles/irani02.asp>
8. [http://www.ebxml.org/white\\_papers/whitepaper.htm](http://www.ebxml.org/white_papers/whitepaper.htm)
9. “Web Content Management Using the OASIS ebXML Registry Standard” , Farrukh Najmi
10. “ebXML Registry – A Tutorial” , Farrukh Najmi & Nikola Stojanovic
11. “ebXML Registry Tutorial: UML Class diagram to ebRIM mapping” , Farrukh Najmi & Nikola Stojanovic & Ivan Bedini
12. “ebXML Registry Repository at Open Forum 2003 on Metadata Registries” , Alan Kotok & Tony Weida & Monica J. Martin
13. [http://ebxmlrr.sourceforge.net/tmp/Registry\\_Capability\\_Matrix.html](http://ebxmlrr.sourceforge.net/tmp/Registry_Capability_Matrix.html)
14. <http://wordpress.org/extend/plugins/register-plus/>
15. <http://jquery.com/>
16. <http://www.ericmmartin.com/projects/simplemodal-demos/>
17. <http://net.tutsplus.com/html-css-techniques/how-to-create-a-slick-tabbed-content-area/>
18. <http://www.netlib.org/lapack/double/dgtsv.f>
19. <http://195.251.17.2:8080/DgtsvWs/DgtsvWsService?wsdl>
20. <http://www.w3.org/2002/ws/history.html>
21. <http://www.netlib.org/blas/dgemm.f>