

ΙΕΡΑΡΧΙΚΟ ΜΟΝΤΕΛΟ ΠΡΟΣΒΑΣΗΣ ΑΝΤΙΚΕΙΜΕΝΩΝ ΜΕ ΤΗ ΧΡΗΣΗ  
ΥΠΗΡΕΣΙΩΝ ΔΙΑΔΙΚΤΥΟΥ

Η ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ ΕΞΕΙΔΙΚΕΥΣΗΣ

Υποβάλλεται στην

ορισθείσα από τη Γενική Συνέλευση Ειδικής Σύνθεσης  
του Τμήματος Πληροφορικής  
Εξεταστική Επιτροπή

από την

Αναστασία Κώστα

ως μέρος των Υποχρεώσεων

για τη λήψη

του

ΜΕΤΑΠΤΥΧΙΑΚΟΥ ΔΙΠΛΩΜΑΤΟΣ ΣΤΗΝ ΠΛΗΡΟΦΟΡΙΚΗ  
ΜΕ ΕΞΕΙΔΙΚΕΥΣΗ ΣΤΟ ΛΟΓΙΣΜΙΚΟ

Ιούνιος, 2008

## **ΕΥΧΑΡΙΣΤΙΕΣ**

---

Το παρόν κείμενο αποτελεί το αποτέλεσμα μιας προσπάθειας, που ολοκληρώθηκε αν και συνάντησε αρκετές δυσκολίες. Θα ήθελα να ευχαριστήσω για την πολύτιμη βοήθειά του τον επιβλέποντα καθηγητή της μεταπτυχιακής εργασίας κ.Μανή. Επιπλέον, τους γονείς μου και την αδερφή μου, τους φίλους μου που με στήριξαν, ιδιαίτερα την Ευπραξία Τσανούσα, χωρίς τη βοήθεια και την κατανόηση των οποίων η μεταπτυχιακή εργασία δεν θα είχε περατωθεί.

## ΠΕΡΙΕΧΟΜΕΝΑ

---

	Σελ
ΕΥΧΑΡΙΣΤΙΕΣ	ii
ΠΕΡΙΕΧΟΜΕΝΑ	iii
ΕΥΡΕΤΗΡΙΟ ΠΙΝΑΚΩΝ	v
ΕΥΡΕΤΗΡΙΟ ΣΧΗΜΑΤΩΝ	vi
ΕΠΕΞΗΓΗΣΕΙΣ ΣΥΜΒΟΛΙΣΜΩΝ	vii
ΠΕΡΙΛΗΨΗ	viii
EXTENDED ABSTRACT IN ENGLISH	x
ΚΕΦΑΛΑΙΟ 1. ΕΙΣΑΓΩΓΗ	1
1.1 Εισαγωγικά στοιχεία	1
1.1.1 Οι ρόλοι στην Αρχιτεκτονική μιας Υπηρεσίας Διαδικτύου	2
1.1.2 Λειτουργίες στην Αρχιτεκτονική των Υπηρεσιών Διαδικτύου	2
1.1.3. Χαρακτηριστικά μιας Υπηρεσίας Διαδικτύου	3
1.1.4. Ανάλυση του Κύκλου Ζωής των Υπηρεσιών Διαδικτύου	4
1.1.5. Η Προδιαγραφή SOAP	5
1.1.6. Ασύγχρονη εναντίον Σύγχρονης Επικοινωνίας	6
1.1.7. Μέθοδος Κωδικοποίησης	7
1.1.8. Μέθοδος Δέσμευσης	8
1.2 Αντικείμενο της Μεταπτυχιακής Εργασίας	8
1.3 Οργάνωση του Τόμου	9
ΚΕΦΑΛΑΙΟ 2. ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΘΕΜΑΤΟΣ	10
2.1 Σχετικές εργασίες	10
2.2 Στόχος	12
ΚΕΦΑΛΑΙΟ 3. ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ	13
3.1 Περιγραφή του Συστήματος	13
3.2 Περιγραφή των Αντικειμένων	16
3.3 Ο Μηχανισμός Κληρονομικότητας	17
3.4 Λειτουργικά Στοιχεία του Συστήματος	20

ΚΕΦΑΛΑΙΟ 4. ΥΛΟΠΟΗΣΗ	22
4.1 Πλατφόρμες και Προγραμματιστικά Εργαλεία	22
4.2 Λεπτομέρειες Υλοποίησης	28
4.2.1 Περιγραφή του Είδους της Ιεραρχίας	28
4.2.2 Αναλυτική Περιγραφή των Εκτελούμενων Λειτουργιών	30
ΚΕΦΑΛΑΙΟ 5. ΕΛΕΓΧΟΣ ΚΑΙ ΜΕΤΡΗΣΕΙΣ	36
5.1 Πειραματικές Μετρήσεις	36
5.2 Διοικητική Οργάνωση Βιβλιοθήκης και Κέντρου Πληροφόρησης Πανεπιστημίου Ιωαννίνων	38
5.2.1. Η Μορφή των Αντικειμένων για την Εφαρμογή	41
5.2.2. Μετρήσεις για τη Συγκεκριμένη Εφαρμογή	42
ΚΕΦΑΛΑΙΟ 6. ΕΠΙΛΟΓΟΣ	43
6.1 Αξιολόγηση του Συστήματος	43
ΑΝΑΦΟΡΕΣ	45
ΠΑΡΑΡΤΗΜΑ	50
ΣΥΝΤΟΜΟ ΒΙΟΓΡΑΦΙΚΟ	56

## ΕΥΡΕΤΗΡΙΟ ΠΙΝΑΚΩΝ

---

Πίνακας	Σελ
Πίνακας 5.1	34
Πίνακας 5.2	35
Πίνακας 5.3	35
Πίνακας 5.4	35
Πίνακας 5.5	35
Πίνακας 5.6	40
Πίνακας 5.7	40

## ΕΥΡΕΤΗΡΙΟ ΣΧΗΜΑΤΩΝ

---

Σχήμα	Σελ
Σχήμα 1.1 Η αρχιτεκτονική μιας υπηρεσίας διαδικτύου.	2
Σχήμα 3.1 Η γενική μορφή των αντικειμένων	16
Σχήμα 3.2 Ο τρόπος επικοινωνίας των αντικειμένων δεδομένων	18
Σχήμα 3.3 Η δομή μιας εταιρείας	19
Σχήμα 4.1 Ο μηχανισμός της Axis	22
Σχήμα 4.2 Η επικοινωνία μεταξύ των υπηρεσιών	26
Σχήμα 4.3 Η ιεραρχία που δημιουργεί το σύστημα	27
Σχήμα 4.4 Τα περιεχόμενα της αίτησης ανάγνωσης	28
Σχήμα 4.5 Η κλήση της μεθόδου με δυναμικό τρόπο	29
Σχήμα 4.6 Τα περιεχόμενα της αίτησης εγγραφής	30
Σχήμα 4.7 Τα περιεχόμενα της αίτησης αντιγραφής αντικειμένου	30
Σχήμα 4.8 Τα περιεχόμενα της αίτησης μετακίνησης αντικειμένου	32
Σχήμα 5.1 Το οργανόγραμμα της Β.Κ.Π.-Π.Ι.	39

## **ΕΠΕΞΗΓΗΣΕΙΣ ΣΥΜΒΟΛΙΣΜΩΝ**

---

XML: Extensible Markup Language

SOAP: Simple Object Access Protocol

W3C: World Wide Web Consortium

WSDL: Web Services Description Language

HTTP: Hyper Text Transfer Protocol

RPC: Remote Procedure Call

MPI: Message Passing Interface

EJB: Enterprise JavaBeans

MDB: Message Driven Bean

JMS: Java Message Service

SMTP: Simple Mail Transfer Protocol

JWS: Java Web Start

JDK: Java Development Kit

XSD: XML Schema Definition

URL: Uniform Resource Locator

GRID: Grid computing, είναι χρησιμοποίηση των πόρων πολλών υπολογιστών σε ένα δίκτυο για την επίλυση ενός προβλήματος κάθε φορά.

CORBA: Common Object Request Broker Architecture

UDDI: Universal Description, Discovery και Integration

## ΠΕΡΙΛΗΨΗ

---

Αναστασία Κώστα του Στεφάνου και της Ευαγγελίας.

MSc, Τμήμα Πληροφορικής, Πανεπιστήμιο Ιωαννίνων

Ιούνιος, 2008

Ιεραρχικό Μοντέλο Πρόσβασης Αντικειμένων με τη χρήση Υπηρεσιών Διαδικτύου.

Επιβλέποντας: Γεώργιος Μανής.

Στο πλαίσιο αυτής της μελέτης εξετάζεται η δημιουργία ενός ιεραρχικού μοντέλου πρόσβασης αντικειμένων με τη χρήση υπηρεσιών διαδικτύου. Προτείνουμε μια αρχιτεκτονική οργάνωσης αντικειμένων-δεδομένων, που βασίζεται στην ιεράρχησή τους και σε σχέσεις κληρονομικότητας που τα διέπουν. Επειδή, η αρχιτεκτονική αυτή βρίσκει κυρίως εφαρμογή σε γεωγραφικά απομακρυσμένους πελάτες χρησιμοποιήσαμε, για την υλοποίηση και αξιολόγησή της, υπηρεσίες διαδικτύου. Κάθε πελάτης έχει τη δυνατότητα να απευθύνει κλήσεις ανάγνωσης και εγγραφής πληροφορίας ή αντιγραφής και μετακίνησης αντικειμένου μεταξύ των υπηρεσιών.

Αν η κλήση αφορά σε πληροφορία, η οποία βρίσκεται σε αντικείμενο υπηρεσίας που απευθύνει την κλήση ο πελάτης, η αίτηση μπορεί να απαντηθεί άμεσα. Αν η πληροφορία δε βρίσκεται εκεί ή βρίσκεται μερικώς εκεί, ενεργοποιείται ο μηχανισμός της κληρονομικότητας. Στην περίπτωση αυτή, η αίτηση προωθείται, μέσω του συγκεκριμένου αντικειμένου, σε αντικείμενα άλλων υπηρεσιών, όπως ορίζει η ιεραρχία. Κατά την αντιγραφή ή μετακίνηση αντικειμένου εκμεταλλευόμαστε πάλι τους μηχανισμούς κληρονομικότητας και προώθησης αιτήσεων.

Η υλοποίηση έγινε με τη χρήση της γλώσσας προγραμματισμού Java και της πλατφόρμας ενδιάμεσου λογισμικού Axis με τη χρήση του εξυπηρετητή Tomcat Apache.



Η δομή της εργασίας έχει ως εξής: αρχικά, παρέχονται εισαγωγικά στοιχεία και αναλύονται όροι που είναι απαραίτητοι για τη μελέτη του θέματος. Προσδιορίζεται το αντικείμενο της μεταπτυχιακής εργασίας, που είναι η δημιουργία ενός ιεραρχικού μοντέλου πρόσβασης αντικειμένων με τη χρήση υπηρεσιών διαδικτύου. Παρατίθενται ορισμένα συστήματα τα οποία έχουν δημιουργηθεί στο παρελθόν και κυμαίνονται στο ίδιο πεδίο έρευνας και δίνονται τα σημεία που κάνουν το προτεινόμενο σύστημα διαφορετικό από τα άλλα συστήματα. Περιγράφονται οι λειτουργίες που παρέχει και η χαρακτηριστική ιεραρχική οργάνωση των αντικειμένων. Αναλύονται τα χαρακτηριστικά της συγκεκριμένης υλοποίησης, όπως η πλατφόρμα ανάπτυξης και εκτέλεσης, τα προγραμματιστικά εργαλεία, οι απαιτήσεις της εφαρμογής σε υλικό, γίνεται αναφορά στη δομή του προγράμματος και στον τρόπο με τον οποίο έχει σχεδιαστεί η υλοποίηση. Αναφέρονται εφαρμογές του συστήματος σε συγκεκριμένα παραδείγματα και γίνεται σχολιασμός του συστήματος για την απόδοσή του και τα πλεονεκτήματα της εφαρμογής του.

## ABSTRACT IN ENGLISH

---

Anastasia Kosta, Stefanos, Evagelia.

MSc, Computer Science Department, University of Ioannina, Greece.

June, 2008

Hierarchical Model of Accessing Objects Using Web Services.

Thesis Supervisor: Georgios Manis

During this study we create and analyze the Hierarchical Model of Accessing Objects Using Web Services. We suggest an architecture of organizing data objects, based on their hierarchical structure and in their heredity relationships. Based on the fact, that the architecture is mostly implemented in geographically distant clients, we used for its implementation and evaluation Web Services. Each client can send read and write information requests or copy and migrate object requests, between services.

If the request is about information, which is located in an object of the service that the client sends its request, then the call can be answered instantly. If the information isn't located there, or is located partially there, then inheritance mechanism is activated. In this case, from this particular object, the request is being forwarded to other service objects, as the hierarchy defines. During the copy or the migration of an object we use the inheritance mechanisms to forward the requests.

For the implementation we used the programming language Java and the Axis middleware with the use of the Tomcat Apache server.

The structure of this study is as follows: initially we give some introduction information and we analyze expressions that are necessary in understanding the study. We define the main subject of this study, which is the creation of a Hierarchical Model of Accessing Objects Using Web Services. We mention some systems that

have been created in the past and they oscillate in the same field of research and we underline the differences with the suggested system. We describe the functions that that the hierarchical object management provides. We analyze the characteristics of this particular implementation, such as the required software and we describe the study's structure and how it has been designed. We analyze system's implementations and we discuss the outputs and the advantages of its implementation.

# ΚΕΦΑΛΑΙΟ 1. ΕΙΣΑΓΩΓΗ

---

## 1.1. Εισαγωγικά Στοιχεία

- 1.1.1 Οι ρόλοι στην Αρχιτεκτονική μιας Υπηρεσίας Διαδικτύου
- 1.1.2 Λειτουργίες στην Αρχιτεκτονική των Υπηρεσιών Διαδικτύου
- 1.1.3 Χαρακτηριστικά μιας Υπηρεσίας Διαδικτύου
- 1.1.4 Ανάλυση του Κύκλου Ζωής των Υπηρεσιών Διαδικτύου
- 1.1.5 Η Προδιαγραφή SOAP
- 1.1.6 Ασύγχρονη εναντίον Σύγχρονης Επικοινωνίας
- 1.1.7 Μέθοδος Κωδικοποίησης
- 1.1.8 Μέθοδος Δέσμευσης

## 1.2. Αντικείμενο της Μεταπτυχιακής Εργασίας

## 1.3. Οργάνωση του Τόμου

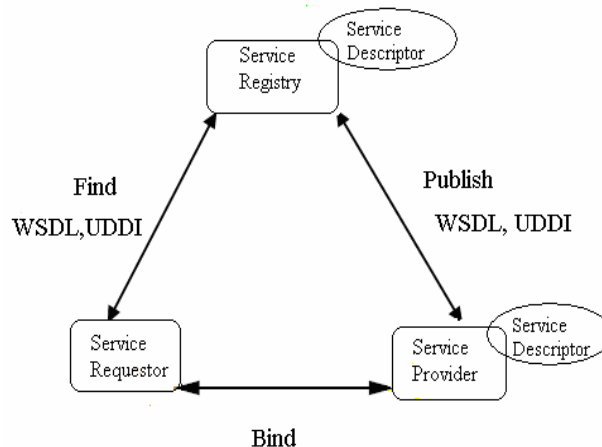
---

### **1.1. Εισαγωγικά Στοιχεία**

Μια υπηρεσία διαδικτύου είναι μια διεπαφή που περιγράφει μια συλλογή υπηρεσιών που είναι προσβάσιμες μέσω του διαδικτύου με καθορισμένα XML μηνύματα. Περιγράφεται από μια XML περιγραφή, που λέγεται περιγραφή της υπηρεσίας. Περιλαμβάνει όλες τις λεπτομέρειες που είναι απαραίτητες για την αλληλεπίδραση με την υπηρεσία, όπως η μορφή των μηνυμάτων και τα πρωτόκολλα μεταφοράς. Η διεπαφή κρύβει τις λεπτομέρειες της εφαρμογής, γεγονός που επιτρέπει να χρησιμοποιηθεί ανεξάρτητα από την πλατφόρμα λογισμικού ή υλικού στην οποία εφαρμόζεται και ανεξάρτητα από τη γλώσσα προγραμματισμού που έχει χρησιμοποιηθεί.

Η αρχιτεκτονική των υπηρεσιών διαδικτύου βασίζεται στην αλληλεπίδραση μεταξύ τριών οντοτήτων με τους εξής ρόλους: παροχή της υπηρεσίας (Service Provider),

αποθήκευση της υπηρεσίας (Service Registry) και αίτηση για παροχή της υπηρεσίας (Service Requestor). Οι συνδιαλλαγές περιλαμβάνουν τη δημοσίευση, την εύρεση και τη δέσμευση υπηρεσίας, όπως φαίνεται στην εικόνα που ακολουθεί.



Σχήμα 1.1 Η αρχιτεκτονική μιας υπηρεσίας διαδικτύου [IBM Software Group]

#### 1.1.1. Οι ρόλοι στην Αρχιτεκτονική μιας Υπηρεσίας Διαδικτύου

- Παροχή υπηρεσίας: Είναι η πλατφόρμα που φιλοξενεί την πρόσβαση στην υπηρεσία.
- Αίτηση για παροχή υπηρεσίας: Είναι η εφαρμογή που αναζητά την υπηρεσία και αρχικοποιεί την συνδιαλλαγή με την υπηρεσία. Τον ρόλο αυτό μπορεί να τον έχει μια άλλη υπηρεσία (υπηρεσία διαδικτύου).
- Αποθήκευση της υπηρεσίας: Είναι ένας χώρος αποθήκευσης, όπου οι παροχείς υπηρεσιών αποθηκεύουν τις περιγραφές των υπηρεσιών τους. Οι αιτήσεις για παροχή υπηρεσίας απευθύνονται στη συγκεκριμένη οντότητα για να πάρουν πληροφορίες για την περιγραφή της υπηρεσίας.

#### 1.1.2. Λειτουργίες στην Αρχιτεκτονική των Υπηρεσιών Διαδικτύου

Για να επωφεληθεί μια εφαρμογή από μια υπηρεσία διαδικτύου πρέπει να συμβούν τρεις συμπεριφορές: δημοσίευση της συμπεριφοράς της υπηρεσίας, αναζήτηση ή εύρεση της περιγραφής της υπηρεσίας, και δέσμευση ή επίκληση των υπηρεσιών που βασίζονται στην περιγραφή της υπηρεσίας. Αυτές οι συμπεριφορές μπορούν να

συμβούν κάθε μια απλά ή επαναληπτικά. Πιο συγκεκριμένα οι λειτουργίες είναι οι εξής:

- Δημοσίευση: Για να είναι προσβάσιμη, η περιγραφή μιας υπηρεσίας θα πρέπει να δημοσιευτεί, έτσι ώστε ο αιτών της υπηρεσίας να μπορεί να την βρει. Το που μπορεί να δημοσιευτεί μια υπηρεσία ποικίλει και εξαρτάται από τις απαιτήσεις της εφαρμογής.
- Εύρεση: Στην λειτουργία αναζήτησης ο αιτών της υπηρεσίας ανακτά την περιγραφή της απευθείας ή ρωτά την καταχώρηση της υπηρεσίας για τον τύπο της υπηρεσίας που χρειάζεται. Η λειτουργία της αναζήτησης μπορεί να εμπλακεί σε δυο διαφορετικές φάσεις του κύκλου ζωής για τον αιτών της υπηρεσίας: Στο χρόνο σχεδιασμού για να ανακτηθεί η διεπαφή της περιγραφής υπηρεσίας για την ανάπτυξη του προγράμματος και στον χρόνο εκτέλεσης για να ανακτηθεί η δέσμευση και η τοποθεσία της υπηρεσίας για επίκληση.
- Δέσμευση: Τελικά μια υπηρεσία απαιτείται να εμπλακεί. Στην λειτουργία της δέσμευσης ο αιτών της υπηρεσίας εμπλέκει ή αρχικοποιεί μια αλληλεπίδραση με την υπηρεσία στο χρόνο εκτέλεσης χρησιμοποιώντας τις λεπτομέρειες της «δέσμευσης» για να εντοπίσει, να επικοινωνήσει και να επικαλεστεί την υπηρεσία.

### *1.1.3. Χαρακτηριστικά μιας Υπηρεσίας Διαδικτύου*

- Υπηρεσία. Μια υπηρεσία διαδικτύου είναι μια διεπαφή που περιγράφεται από μια περιγραφή της υπηρεσίας και η υλοποίησή της είναι η υπηρεσία. Μια υπηρεσία είναι μια ενότητα λογισμικού, που εφαρμόζεται σε πλατφόρμες προσβάσιμες μέσω δικτύου και παρέχεται από τον παροχέα υπηρεσιών. Μπορεί να κληθεί ή να αλληλεπιδράσει με τον αντίστοιχο καλούντα της υπηρεσίας. Μπορεί να λειτουργήσει και ως καλών χρησιμοποιώντας άλλες υπηρεσίες διαδικτύου στην υλοποίηση της.
- Περιγραφή της υπηρεσίας. Η περιγραφή της υπηρεσίας περιέχει τις λεπτομέρειες της διεπαφής και της υλοποίησης της υπηρεσίας. Αυτό περιλαμβάνει τους τύπους δεδομένων, τις λειτουργίες, τις πληροφορίες για τη δέσμευση της υπηρεσίας και την τοποθεσία του δικτύου. Μπορεί επίσης, να περιλαμβάνει κατηγοριοποίηση και άλλα

μεταδεδομένα για να φιλοξενηθεί, ανακαλυφθεί και να χρησιμοποιηθεί από τους καλούντες της υπηρεσίας. Η περιγραφή της υπηρεσίας μπορεί να δημοσιευθεί σε έναν καλούντα της υπηρεσίας ή σε έναν καταχωρητή υπηρεσιών. Η αρχιτεκτονική των υπηρεσιών διαδικτύου εξηγεί πως μπορεί να γίνει η αρχικοποίηση των στοιχείων και να εφαρμοστούν οι λειτουργίες με διαλειτουργικό τρόπο.

#### *1.1.4. Ανάλυση του Κύκλου Ζωής των Υπηρεσιών Διαδικτύου*

Η ανάλυση του κύκλου ζωής των υπηρεσιών διαδικτύου περιλαμβάνει το σχεδιασμό, την εφαρμογή και τις απαιτήσεις κατά το χρόνο εκτέλεσης για κάθε έναν από τους παρακάτω ρόλους: τον καταχωρητή υπηρεσιών, τον παροχέα υπηρεσιών και τον καλούντα των υπηρεσιών. Κάθε ρόλος έχει συγκεκριμένες απαιτήσεις για κάθε στοιχείο του κύκλου ζωής της ανάπτυξης. Η ανάπτυξη και η εφαρμογή του καταχωρητή υπηρεσιών περιλαμβάνεται σε αυτή τη μελέτη.

Η ανάπτυξη του κύκλου ζωής μπορεί να έχει 4 φάσεις:

##### 1. Κατασκευή

Η φάση της κατασκευής του κύκλου ζωής περιλαμβάνει την εφαρμογή και τον έλεγχο της υλοποίησης μιας υπηρεσίας διαδικτύου, τον ορισμό της περιγραφής της διεπαφής της υπηρεσίας και τον ορισμό της περιγραφής της υλοποίησης της υπηρεσίας. Η υλοποίηση των υπηρεσιών διαδικτύου μπορεί να γίνει με τη δημιουργία νέων υπηρεσιών διαδικτύου, μετατρέποντας προϋπάρχουσες εφαρμογές σε υπηρεσίες διαδικτύου, και συνθέτοντας νέες υπηρεσίες διαδικτύου από άλλες υπηρεσίες διαδικτύου και εφαρμογές.

##### 2. Εφαρμογή

Η φάση της εφαρμογής περιλαμβάνει τη δημοσίευση της διεπαφής της υπηρεσίας και τον ορισμό της εφαρμογής σε έναν καλούντα της υπηρεσίας ή σε έναν καταχωρητή υπηρεσιών και την εφαρμογή των εκτελέσιμων σε ένα περιβάλλον εκτέλεσης(τυπικά, έναν εξυπηρετητή εφαρμογών διαδικτύου).

##### 3. Εκτέλεση

Κατά την φάση της εκτέλεσης, η υπηρεσία διαδικτύου είναι διαθέσιμη για κλήση. Σε αυτό το σημείο, η υπηρεσία διαδικτύου είναι πλήρως εφαρμόσιμη, λειτουργική και

προσβάσιμη μέσω δικτύου από τον παροχέα υπηρεσιών. Τώρα, ο παροχέας υπηρεσιών μπορεί να εκτελέσει τις λειτουργίες εύρεσης και δέσμευσης της υπηρεσίας.

#### 4. Διαχείριση

Η φάση της διαχείρισης καλύπτει την τρέχουσα διαχείριση και διοίκηση της εφαρμογής της υπηρεσίας διαδικτύου. Η ασφάλεια, η διαθεσιμότητα, η απόδοση, η ποιότητα της υπηρεσίας πρέπει να διευθετηθούν σε αυτή τη φάση.

##### *1.1.5. Η Προδιαγραφή SOAP*

Οι XML υπηρεσίες διαδικτύου στηρίζονται στο Simple Object Access Protocol (SOAP). Η προδιαγραφή SOAP κατατέθηκε στο W3C και προσδιορίζει ένα πρωτόκολλο βασισμένο σε XML για να διευκολύνει την ανταλλαγή μηνυμάτων μεταξύ αποκεντρωμένων και κατακεντρωμένων συστημάτων.

Ο σχεδιασμός της διεπαφής των υπηρεσιών διαδικτύου απαιτεί την λεπτομερή περιγραφή των χρησιμοποιούμενων τεχνολογιών και των σχετικών συναλλαγών που λαμβάνουν χώρα.

Η προδιαγραφή SOAP δημιουργήθηκε για να επιτραπεί στις εφαρμογές να λειτουργούν με σύγχρονο τρόπο, δηλαδή να αποστέλλουν σύγχρονες απομακρυσμένου τύπου κλήσεις προς μια υπηρεσία κατά μήκος του δικτύου με τεχνολογικά ουδέτερο τρόπο. Η προδιαγραφή SOAP στην πραγματικότητα προσδιορίζει ένα ξεχωριστό μήνυμα αίτησης και απάντησης για κάθε κλήση προς μια υπηρεσία (όπως ορίζεται στην WSDL περιγραφή της υπηρεσίας). Το SOAP αναπτύχθηκε βασισμένο στη λογική των μηνυμάτων αλλά η πλειοψηφία των εφαρμογών των υπηρεσιών διαδικτύου χρησιμοποιούν το HTTP ως το πρωτόκολλο μεταφοράς. Το HTTP είναι μια φυσική επιλογή διότι είναι το πιο συχνά χρησιμοποιούμενο πρωτόκολλο στο Διαδίκτυο και μπορεί να περάσει από τις ζώνες ασφάλειας. Ωστόσο, το πρωτόκολλο HTTP είναι το "Hyper Text Transfer Protocol" – σχεδιάστηκε για να επιτρέπει στους χρήστες που διαθέτουν φυλλομετρητές να ανακτούν έγγραφα μέσω του Διαδικτύου, και όχι για να επικοινωνούν μεταξύ τους οι εφαρμογές. Το πρωτόκολλο HTTP είναι εγγενώς αναξιόπιστος και σχεδιάστηκε για



τη σύγχρονη ανάκτηση εγγράφων – η εφαρμογή πελάτη χρησιμοποιεί την ίδια σύνδεση για να στείλει την αίτηση στον εξυπηρετητή και για να λάβει την απάντηση. Για το λόγο αυτό, οι υπηρεσίες διαδικτύου που χρησιμοποιούν το πρωτόκολλο HTTP χρησιμοποιούν και το σύγχρονο μήνυμα αιτήματος/απάντησης, διότι η ασύγχρονη επικοινωνία πάνω σε ένα αναξιόπιστο κανάλι είναι τόσο χρήσιμη σαν μια σταγόνα στον ωκεανό.

#### *1.1.6. Ασύγχρονη εναντίον Σύγχρονης επικοινωνίας*

Σε μια σύγχρονη υλοποίηση μιας υπηρεσίας διαδικτύου, η σύνδεση του πελάτη παραμένει ανοιχτή από τη στιγμή που η αίτηση στάλθηκε στον εξυπηρετητή. Ο πελάτης θα περιμένει μέχρι ο εξυπηρετητής να στείλει το απαντητικό μήνυμα. Το πλεονέκτημα της χρήσης της σύγχρονης επικοινωνίας είναι ότι η εφαρμογή του πελάτη γνωρίζει την κατάσταση λειτουργίας της υπηρεσίας διαδικτύου σε πολύ μικρό χρονικό διάστημα(είτε λαμβάνει απάντηση είτε τελειώνει ο χρόνος αναμονής). Ένας σοβαρός περιορισμός που προκύπτει από τη χρήση της σύγχρονης λειτουργίας, είναι ότι ο εξυπηρετητής μπορεί να έχει να διαχειριστεί ένα μεγάλο αριθμό ταυτόχρονων συνδέσεων γιατί ο κάθε πελάτης διατηρεί ανοιχτή τη σύνδεση καθώς περιμένει τα αποτελέσματα. Αυτό οδηγεί την εφαρμογή του εξυπηρετητή να γίνει εξαιρετικά πολύπλοκη. Αν μια από τις κλήσεις, σε έναν σύγχρονο παροχέα υπηρεσιών αποτύχει, η εφαρμογή του εξυπηρετητή πρέπει να παρέχει έναν μηχανισμό έτσι ώστε να παγιδεύσει την αποτυχία, να ανακάμψει και να επαναδρομολογήσει την διαδικασία ή να σημάνει λάθος πριν συνεχίσει με άλλες σύγχρονες κλήσεις.

Προς το παρόν τα περισσότερα πακέτα λογισμικού παρέχουν εξ'ορισμού σύγχρονη επικοινωνία. Ωστόσο με τη χρήση προτύπων και εργαλείων που ήδη υπάρχουν μπορεί να επιτευχθεί και ασύγχρονη επικοινωνία για υπηρεσίες διαδικτύου. Αρκετοί οργανισμοί και εταιρείες έχουν αναγνωρίσει την ανάγκη να υπάρχει ασύγχρονη λειτουργία και έχουν κινηθεί προς αυτή την κατεύθυνση με τον καθορισμό νέων προτύπων (π.χ. WS-ReliableMessaging).

### 1.1.7. Μέθοδος Κωδικοποίησης

Η έννοια της κωδικοποίησης του SOAP έχει οδηγήσει σε δικαιολογημένες συζητήσεις και σύγχυση. Το πρότυπο SOAP καθορίζει μια μέθοδο που ονομάζεται "μέθοδος κωδικοποίησης" και προσδιορίζεται από την ιδιότητα `encodingStyle`. Αυτή η ιδιότητα μπορεί να πάρει δύο τιμές, "encoded" και "literal" (ορίζοντας μιας διαφορετική (ή όχι) τιμή της ιδιότητας). Αυτός ο τρόπος καθορίζει πως τα αντικείμενα μιας εφαρμογής και οι παράμετροι που έχει αναπαρίστανται σε XML. Το χαρακτηριστικό "Encoded" (αναφέρεται επίσης κωδικοποίηση SOAP), και καθορίζει έναν πρωταρχικό μηχανισμό για την αναπαράσταση των τύπων μιας γλώσσας προγραμματισμού σε XML. Το χαρακτηριστικό "Literal" (αναφέρεται επίσης ως "doc/literal") είναι αποτρεπτικός όρος. Αντίθετα, ο τύπος της πληροφορίας παρέχεται από ένα εξωτερικό μηχανισμό, πιθανότατα όχι από μια WSDL (Web Services Description Language) περιγραφή που χρησιμοποιεί έννοιες XML για να προσδιορίσει τους τύπους που χρησιμοποιούνται στο μήνυμα SOAP.

Ο λόγος που προέκυψε αυτό είναι, διότι η προδιαγραφή SOAP ορίστηκε πριν την υιοθέτηση του προτύπου XML (XSD). Επιπλέον, το αρχικό πρότυπο SOAP έπρεπε να παρέχει έναν τρόπο κωδικοποίησης της πληροφορίας μαζί με τις παραμέτρους που έπρεπε να σταλούν για τις κλήσεις των μεθόδων, διότι δεν υπήρχε γενικά αποδεκτός τρόπος. Η ανάγκη αυτή υπεισέρχεται στην περίπτωση ιδιαίτερα πολύπλοκων τύπων, όπως είναι οι πίνακες. Το πρότυπο SOAP ορίζει έναν συγκεκριμένο μηχανισμό για να αναπαρασταθούν οι πίνακες μιας γλώσσας προγραμματισμού σε XML (SOAPEnc:Array schema type).

Ωστόσο, από τη στιγμή της υιοθέτησης του προτύπου XML, οι περισσότερες γλώσσες έχουν εστιάσει στη χρήση των δικών τους αντιστοιχίσεων (ή κανόνων αλληλουχίας) από το πρότυπο XML στους τύπους της γλώσσας προγραμματισμού. Για παράδειγμα το πρότυπο JAX-RPC ορίζει μοναδικά, πως οι τύποι της γλώσσας προγραμματισμού Java αντιστοιχίζονται σε στοιχεία XML και το αντίστροφο. Αυτό προλαμβάνει την ανάγκη για επιπλέον πληροφορία στην XML. Σαν αποτέλεσμα η κωδικοποίηση SOAP δεν προτιμάται πλέον, και έχει αντικατασταθεί από την literal κωδικοποίηση με την αντιστοίχιση να καθορίζεται εξωτερικά από ένα πρότυπο έγγραφο XML συνήθως με τη μορφή ενός εγγράφου WSDL.

### 1.1.8. Μέθοδος Δέσμευσης

Η προδιαγραφή WSDL προσδιορίζει δύο διαφορετικούς τρόπους δέσμευσης στο πρωτόκολλο SOAP. Οι τιμές του τρόπου δέσμευσης είναι "RPC" και "Document". Αυτό σημαίνει ότι αν μια WSDL περιγραφή θέτει μια λειτουργία έχει τύπο δέσμευσης "RPC", τότε ο παραλήπτης πρέπει να αναλύσει το μήνυμα χρησιμοποιώντας τους αντίστοιχους κανόνες του προτύπου SOAP. Αυτό σημαίνει για παράδειγμα, ότι το XML στοιχείο στο σώμα του SOAP (ονομάζεται " συσκευασμένο στοιχείο") πρέπει να έχει ένα όνομα όμοιο με αυτό της λειτουργίας της γλώσσας προγραμματισμού που θα επικαλεσθεί, και κάθε τμήμα του μηνύματος μέσα σε αυτό το στοιχείο πρέπει να απευθύνεται ακριβώς (κατά όνομα και κατά σειρά) σε μια παράμετρο της αντίστοιχης λειτουργίας. Επιπλέον, πρέπει να επιστρέφεται ένα και μοναδικό στοιχείο (που πρέπει να ονομάζεται XXXResponse, όπου XXX είναι το όνομα της αντίστοιχης λειτουργίας στη γλώσσα), το οποίο περιέχει ακριβώς ένα στοιχείο, που είναι η τιμή επιστροφής της λειτουργίας.

Ο τύπος δέσμευσης «Document» δεν είναι τόσο αυστηρός. Ένα μήνυμα μέσα στον τύπο δέσμευσης «Document» μπορεί να φτιαχτεί απλά από μια XML περιγραφή. Εξαρτάται από τη λειτουργία SOAP που το λαμβάνει πως θα το ερμηνεύσει. Σύμφωνα με όσα ειπώθηκαν παραπάνω, είναι αρκετά συχνό (για παράδειγμα στα εργαλεία της Microsoft) να χρησιμοποιείται ο τύπος δέσμευσης "Document" και η κωδικοποίηση "Literal" για την αναπαράσταση κλήσεων RPC.

## 1.2. Αντικείμενο της Μεταπτυχιακής Εργασίας

Στην μελέτη που παρουσιάζεται προτείνεται μια αρχιτεκτονική και η υλοποίησή της που στοχεύει στην κατανομημένη αποθήκευση δεδομένων. Τα δεδομένα μεταφέρονται μέσω υπηρεσιών διαδικτύου από απομακρυσμένους πελάτες. Το σύστημα αποτελείται από πελάτες, που μπορούν να απευθύνουν κλήσεις ανάγνωσης και εγγραφής πληροφορίας, που αποτελείται από απλούς και σύνθετους τύπους δεδομένων, και αντιγραφής και μετακίνησης αντικειμένου μεταξύ των υπηρεσιών.

### 1.3. Οργάνωση του Τόμου

Στο πρώτο κεφάλαιο παρέχονται εισαγωγικά στοιχεία και αναλύονται όροι που είναι απαραίτητοι για τη μελέτη του θέματος. Επιπλέον, προσδιορίζεται το αντικείμενο της μεταπτυχιακής εργασίας, που είναι η δημιουργία ενός ιεραρχικού μοντέλου πρόσβασης αντικειμένων με τη χρήση υπηρεσιών διαδικτύου.

Στο δεύτερο κεφάλαιο παραθέτονται συστήματα τα οποία έχουν δημιουργηθεί στο παρελθόν και κυμαίνονται στο ίδιο πεδίο έρευνας. Επισημαίνονται οι διαφορές με το προτεινόμενο σύστημα και αναλύεται ο στόχος της μεταπτυχιακής εργασίας.

Στο τρίτο κεφάλαιο γίνεται περιγραφή του συστήματος και των λειτουργιών που παρέχει. Επιπλέον, δίνεται η γενική μορφή των αντικειμένων που χρησιμοποιούνται για την αποθήκευση των πληροφοριών και αναδεικνύεται ο σημαντικός ρόλος που διαδραματίζουν ως συστατικά στοιχεία του συστήματος.

Στο τέταρτο κεφάλαιο περιγράφονται τα χαρακτηριστικά της συγκεκριμένης υλοποίησης, όπως η πλατφόρμα ανάπτυξης και εκτέλεσης, τα προγραμματιστικά εργαλεία και οι απαιτήσεις της εφαρμογής σε υλικό. Επίσης, παρέχεται αναλυτική περιγραφή της υλοποίησης όλων των χαρακτηριστικών του συστήματος, όπως είναι ο μηχανισμός κληρονομικότητας και οι διαθέσιμες λειτουργίες.

Στο πέμπτο κεφάλαιο γίνεται εκτενής έλεγχος των λειτουργιών και παραθέτονται εφαρμογές του συστήματος σε συγκεκριμένα παραδείγματα. Ο έλεγχος περιλαμβάνει μετρήσεις, που αφορούν στο χρόνο που μεσολαβεί μεταξύ αποστολής αίτησης και λήψης απάντησης.

Στο έκτο κεφάλαιο γίνεται ο τελικός σχολιασμός του συστήματος και εξάγονται χρήσιμα συμπεράσματα.

## ΚΕΦΑΛΑΙΟ 2. ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΘΕΜΑΤΟΣ

---

### 2.1. Σχετικές Εργασίες

### 2.2. Στόχος

---

### 2.1. Σχετικές Εργασίες

Τα τελευταία χρόνια η ανάγκη για παράλληλο και καταναμημένο υπολογισμό έχει αυξηθεί σημαντικά. Αυτό αφορά τόσο σε υπολογισμούς σε ομάδες που χρησιμοποιούν MPI, όσο και σε αρχιτεκτονικές Grid και τεχνολογίες διαδικτύου όπως J2EE, servlets, JSP κτλ. Για την διαχείριση λοιπόν τεράστιων ποσοτήτων δεδομένων είναι απαραίτητο να βρεθούν επαρκή μέσα.

Υπάρχουν πολλά συστήματα που στοχεύουν στην παράλληλη και καταναμημένη επεξεργασία δεδομένων. Ωστόσο, φαίνονται να δίνουν λύσεις που είτε είναι επικεντρωμένες σε παράλληλους υπολογισμούς ανάμεσα σε ζεύγη διεργασιών όπως MPI-IO[1], είτε είναι πολύ γενικές, όπως συστήματα που βασίζονται στο WWW πρότυπο για παράδειγμα Metacat[2] ή OceanStore[3].

Τα τελευταία είναι γενικές λύσεις που ενσωματώνουν απομακρυσμένο έλεγχο εργασιών, διαχείριση δεδομένων και αρκετές άλλες υπηρεσίες.

Το MPI-I/O ([6]) είναι ένα σύνολο λειτουργιών παράλληλης πρόσβασης σε αρχεία που ορίζεται στο MPI-2 και περιορίζεται στο MPI. Υπάρχουν συστήματα που υποστηρίζουν διαλειτουργικότητα και ελεγχόμενο διαμοιρασμό αρχείων για εφαρμογές σε διαφορετικούς Μαζικούς Παράλληλους Επεξεργαστές (MPPs). Αυτό επιτρέπει διαφορετικές υλοποιήσεις MPI να λειτουργούν όπως στο MPI Connect ([13]). Το MPI Conn IO API επιτρέπει την πρόσβαση σε αρχεία από παράλληλες εφαρμογές που εκτελούνται σε διαφορετικούς παράλληλους υπολογιστές. Το MPI

Conn IO API χρησιμοποιείται για να ανοίξει ένα αρχείο καθολικά και να το καταναίμει σε διάφορα παράλληλα σημεία, και στη συνέχεια πιθανότατα να ανοιχθεί με τη χρήση MPI-2 MPI File open() και να ενημερωθεί. Το [14] προτείνει Stampi-I/O – μια κατανεμημένη παράλληλη βιβλιοθήκη E/E, που είναι μια υλοποίηση του MPI-I/O, η οποία υποστηρίζει παράλληλες λειτουργίες ανάγνωσης/εγγραφής σε αρχεία. Ένα ακόμα παρόμοιο σύστημα είναι το PACX-MPI PIO ([13]) στο οποίο πολλοί πελάτες μπορούν να έχουν πρόσβαση σε πολλούς παράλληλους εξυπηρετητές αρχείων. Το PIOUS ([15]) είναι επίσης ένα παρόμοιο σύστημα για PVM, το οποίο υλοποιεί ένα εικονικό σύστημα αρχείων σε ένα περιβάλλον PVM και επιπλέον είναι ένα παράλληλο και όχι ένα κατανεμημένο σύστημα.

Τα συστήματα που ακολουθούν υποστηρίζουν κατανεμημένη αποθήκευση αρχείων. Το [16] παρουσιάζει την αρχιτεκτονική του WebFS – ένα κατανεμημένο σύστημα αρχείων για μη τροποποιημένες εφαρμογές που χρησιμοποιούν το πρωτόκολλο HTTP για να γράψουν και να ανακτήσουν αρχεία.

Το UFO ([17]) είναι μια ακόμα υλοποίηση ενός κατανεμημένου συστήματος αρχείων, στο οποίο τα αρχεία μπορούν να χρησιμοποιηθούν σαν να είναι τοπικά και να η πρόσβαση να γίνεται με τη χρήση των πρωτοκόλλων

Τα παρακάτω δύο συστήματα είναι επίσης πολύ γενικά για να χρησιμοποιηθούν σε εφαρμογές RPC. Για παράδειγμα, το OceanStore ([8]) έχει αναπτυχθεί με τη λογική ότι θα χρησιμοποιηθεί από χιλιάδες χρήστες, που διαθέτουν τεράστιους όγκους δεδομένων. Είναι διασκορπισμένο στο Διαδίκτυο και αποτελείται από κατανεμημένους σωρούς αποθηκευτικών συστημάτων, με καθένα να αποτελείται από συγκεκριμένους εξυπηρετητές. Το [7] παρουσιάζει το Metacat – ένα πλαίσιο για κατανεμημένη αποθήκευση δεδομένων, που είναι φυσικά κατανεμημένα κατά μήκος του Διαδικτύου, πιθανότατα ετερογενή και βασισμένα σε κάποια κωδικοποίηση. Τα δεδομένα αποθηκεύονται σε XML και σε βάσεις δεδομένων συμβατές με SQL.

Εκτιμώντας τις τελευταίες εξελίξεις, υπάρχουν συστήματα - που βασίζονται σε πλέγματα όπως τα EU-DataGrid ([9]) and GridLab ([11]). Τα συστήματα επικεντρώνονται σε υψηλού επιπέδου κατανεμημένο έλεγχο για μεγάλες εφαρμογές. Το Global Access to Secondary Storage system (GASS, [18]), που είναι τμήμα του

Globus toolkit ([10]) και χρησιμοποιείται σε συστήματα βασισμένα σε πλέγματα, είναι μια υπηρεσία για πρόσβαση σε πληροφορίες που χρησιμοποιεί URLs για να υλοποιήσει ένα χώρο για αρχεία καθολικής πρόσβασης. Παρέχει κρυφές τεχνικές για λειτουργίες εγγραφής και ανάγνωσης.

Τέλος, το DOOM [27] είναι ένα σύστημα που βασίζεται στην ανταλλαγή μηνυμάτων μεταξύ διεργασιών, που έχουν ρόλους πελάτη και εξυπηρετητή και χρησιμοποιεί για την εγκαθίδρυση της επικοινωνίας το MPI. Το σύστημα αυτό παρουσιάζει ομοιότητες στην αρχιτεκτονική του με το προτεινόμενο μοντέλο της παρούσας μελέτης.

## **2.2. Στόχος**

Ο κύριος στόχος της μελέτης αυτής είναι η δημιουργία ενός συστήματος, που θα εκμεταλλεύεται τα χαρακτηριστικά των υπηρεσιών διαδικτύου για να εγκαθιδρύσει την επικοινωνία μεταξύ απομακρυσμένων υπηρεσιών.

Κύριο μέλημά μας είναι να υλοποιήσουμε όλες τις ιδιότητες της προτεινόμενης αρχιτεκτονικής, αποσκοπώντας στην κατανεμημένη επεξεργασία δεδομένων, και να μελετήσουμε τη βέλτιστη εφαρμογή τους, κυρίως σε μικρής εμβέλειας εφαρμογές, που δεν απαιτούν τη διαχείριση τεράστιων όγκων δεδομένων. Στην προηγούμενη ενότητα περιγράφηκαν συστήματα, που ο ρόλος τους είναι να διαχειρίζονται πολύ μεγάλες ποσότητες πληροφοριών καθώς και να υλοποιούν μια πληθώρα άλλων λειτουργιών.

## ΚΕΦΑΛΑΙΟ 3. ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΘΕΜΑΤΟΣ

---

- 3.1. Περιγραφή του Συστήματος
  - 3.2. Περιγραφή των Αντικειμένων
  - 3.3. Ο Μηχανισμός Κληρονομικότητας
  - 3.4 Λειτουργικά Στοιχεία του Συστήματος
- 

### 3.1. Περιγραφή του Συστήματος

Η προτεινόμενη αρχιτεκτονική στηρίζεται στην εγκαθίδρυση επικοινωνίας μεταξύ υπηρεσιών διαδικτύου. Ο συγκεκριμένος σχεδιασμός προήλθε από την ανάγκη της μεταφοράς ενός προηγούμενου υπολογιστικού μοντέλου που ονομάζεται DOOM[27], σε διαφορετική πλατφόρμα υλοποίησης για να εξυπηρετηθούν διαφορετικής φύσης υπηρεσίες.

Γενικά, ο αντικειμενοστραφής προγραμματισμός περιγράφεται από τη χρήση αντικειμένων, κλάσεων και κληρονομικότητας. Τα αντικείμενα αυτά περιέχουν δεδομένα και μεθόδους. Οι κλάσεις παρέχουν τους αναγκαίους μηχανισμούς για την κληρονομικότητα. Σύμφωνα με την παραδοσιακή προσέγγιση, οι κλάσεις μοιάζουν αναντικατάστατες στον αντικειμενοστραφή προγραμματισμό, ειδικά επειδή σχετίζονται άμεσα με την κληρονομικότητα. Σε μια εναλλακτική προσέγγιση που παρουσιάζεται στη βιβλιογραφία ως «delegation», τα αντικείμενα δεν εντάσσονται σε κλάσεις αλλά δημιουργούνται σαν κλώνοι υπαρχόντων αντικειμένων. Η κληρονομικότητα εισάγεται δημιουργώντας ένα αντικείμενο και όχι μέσα από την ιεραρχία των κλάσεων.



Ένα αντικείμενο μπορεί να κληρονομεί δεδομένα και μεθόδους από ένα άλλο αντικείμενο, προωθώντας τις κλήσεις που δεν μπορεί να εξυπηρετήσει σε κάποιο άλλο αντικείμενο. Το αντικείμενο-πατέρας μπορεί να προωθήσει την αίτηση πιο ψηλά στο ιεραρχικό δένδρο, αν αυτό είναι απαραίτητο, μέχρι ένα αντικείμενο να μπορεί να απαντήσει ή να μην υπάρχει άλλο αντικείμενο-πατέρας στο ιεραρχικό δένδρο. Στην περίπτωση αυτή η αίτηση αποτυγχάνει.

Το προτεινόμενο σύστημα περιγράφεται περισσότερο σαν αντικειμενοστραφές όσον αφορά τα δεδομένα και όχι τα αντικείμενα. Τα αντικείμενα δεδομένων είναι δομές στις οποίες έχουν πρόσβαση διαφορετικές υπηρεσίες. Σύμφωνα με την προηγούμενη ιδέα, η ιεραρχία που βασίζεται στα αντικείμενα δεδομένων χαρακτηρίζεται από το να επιτρέπεται στις αιτήσεις να προωθούνται στα αντικείμενα απογόνους, όταν το συγκεκριμένο αντικείμενο δεν περιέχει τα ζητούμενα δεδομένα.

Τα δεδομένα μπορούν προσπελαστούν μέσα από έναν αριθμό από λειτουργίες όπως, ανάγνωση, τροποποίηση, μετακίνηση και αντιγραφή. Τα αντικείμενα δεδομένων είναι αποθηκευμένα στα αρχεία των υπηρεσιών. Όλες οι λειτουργίες που υποστηρίζονται πραγματοποιούνται σε κάθε υπηρεσία εξαλείφοντας την ανάγκη να κληρονομηθεί οποιαδήποτε συμπεριφορά.

Στη συνέχεια θα αναλυθεί με ποιο τρόπο περιγράφεται η κληρονομικότητα, που θα βρίσκεται αυτή η πληροφορία και πως θα τη διαχειριστούμε. Μια λύση είναι όλη η πληροφορία να βρίσκεται μέσα στα αντικείμενα. Με αυτόν τον τρόπο οι υπηρεσίες θα έχουν τη δυνατότητα να χειρίζονται τις αιτήσεις που δέχονται μόνοι τους, χωρίς να είναι απαραίτητη η ανάμιξη άλλων υπηρεσιών. Ο μηχανισμός για την προώθηση της αίτησης στο αντικείμενο γονέα περιέχεται στη λειτουργικότητα της υπηρεσίας και όλες οι αποφάσεις λαμβάνονται τοπικά.

Το προτεινόμενο σύστημα υποστηρίζει πρόσβαση σε δεδομένα μέσα από λειτουργίες ανάγνωσης και τροποποίησης, μετακίνησης αντικειμένων και αντιγραφής. Οι λειτουργίες αυτές περιγράφονται παρακάτω:

### **Αίτηση για ανάγνωση δεδομένων**

Έστω ότι ο πελάτης χρειάζεται κάποια πληροφορία. Για να αποκτήσει την συγκεκριμένη πληροφορία αποστέλλει αίτηση, βάσει της οποίας αναζητά τα δεδομένα. Η αίτηση (μήνυμα) που στέλνει περιλαμβάνει το όνομα της υπηρεσίας που διαθέτει το αντικείμενο, το όνομα του αντικειμένου, τις ομαδοποιημένες πληροφορίες που θα διαβαστούν, το είδος της λειτουργίας που θα εκτελεστεί και το όνομα της μεθόδου που θα αναλάβει τον αρχικό έλεγχο για την ύπαρξη ή όχι του αντικειμένου καθώς και τις ενέργειες που πρέπει να ακολουθηθούν σε κάθε περίπτωση.

### **Αίτηση για εγγραφή δεδομένων**

Η αίτηση για εγγραφή πληροφορίας περιλαμβάνει την αποστολή με όμοιο τρόπο μιας αίτησης-μηνύματος, που στόχο έχει την εύρεση της πληροφορίας και την τροποποίηση της. Συγκεκριμένα, το πακέτο της αίτησης αποτελείται από το όνομα της υπηρεσίας που έχει το ζητούμενο αντικείμενο, το όνομα του αντικειμένου, τις πληροφορίες που πρέπει αρχικά να βρεθούν και στη συνέχεια να αντικατασταθούν με νέες οποίες αποστέλλονται με την αίτηση, το είδος της είδος της λειτουργίας και το όνομα της μεθόδου που ελέγχει τη διαθεσιμότητα των αντικειμένων στη συγκεκριμένη υπηρεσία και λειτουργεί ανάλογα με τα αποτελέσματα.

### **Αίτηση για αντιγραφή αντικειμένου**

Η διαδικασία της αντιγραφής ενός αντικειμένου ξεκινά με την αντίστοιχη αίτηση προς την υπηρεσία που το περιέχει. Συγκεκριμένα, αποστέλλεται ξανά ένα μήνυμα-πακέτο όπου περιλαμβάνει το όνομα της υπηρεσίας-πηγή, δηλαδή της υπηρεσίας που έχει το ζητούμενο αντικείμενο, το όνομα του αντικειμένου, την υπηρεσία-προορισμό όπου θα αντιγραφεί το αντικείμενο, το είδος της λειτουργίας και την μέθοδο που αναλαμβάνει τον έλεγχο της διαδικασίας. Η αντιγραφή αντικειμένου είναι μια διαδικασία που απαιτεί ελέγχους προκειμένου τα νέα αντικείμενα να ενημερώνουν το αρχικό για αλλαγές που υφίστανται, κατά την ικανοποίηση των αιτήσεων του πελάτη. Ο τρόπος που υλοποιούνται αυτοί οι έλεγχοι περιγράφεται στο τέταρτο κεφάλαιο.

### Αίτηση για μετακίνηση αντικειμένου

Η αίτηση για μετακίνηση αντικειμένου ξεκινά από την κλήση του πελάτη προς μια υπηρεσία. Η κλήση περιλαμβάνει πληροφορίες, που σχετίζονται με το όνομα της υπηρεσίας που καλείται και είναι η υπηρεσία-πηγή, το όνομα του αντικειμένου που θα μετακινηθεί, το όνομα της υπηρεσίας στην οποία θα μεταφερθεί, το είδος της λειτουργίας που εκτελείται και τη μέθοδο που ελέγχει τη διαδικασία. Η μετακίνηση αντικειμένου, όπως και η αντιγραφή, απαιτεί ελέγχους προκειμένου οι αιτήσεις να μην κατευθύνονται σε λάθος αντικείμενα ή ακόμα και να χάνονται. Ο τρόπος που υλοποιούνται οι έλεγχοι περιγράφεται στο τέταρτο κεφάλαιο.

### 3.2. Περιγραφή των Αντικειμένων

Οι αιτήσεις που απευθύνει ο πελάτης, τα είδη των οποίων αναφέρθηκαν στην προηγούμενη ενότητα, στοχεύουν στην προσπέλαση πληροφοριών που βρίσκονται αποθηκεύονται σε αντικείμενα δεδομένων. Τα αντικείμενα δεδομένων αποτελούν βασικά στοιχεία του συστήματος και σε αυτά στηρίζεται η ιεραρχία που εγκαθιδρύεται και χαρακτηρίζει το σύστημα μας. Η γενική μορφή των αντικειμένων φαίνεται στο Σχήμα 3.1

Πεδίο 1:	όνομα αντικειμένου
Πεδίο 2:	όνομα αντικειμένου και υπηρεσίας από όπου κληρονομεί ιδιότητες
Πεδίο 3:	πληροφορίες: <ul style="list-style-type: none"> <li>▪ Ακέραιος αριθμός (τύπος όνομα τιμή)</li> <li>▪ Πραγματικός αριθμός (τύπος όνομα τιμή)</li> <li>▪ Αλφαριθμητικό (τύπος όνομα τιμή)</li> <li>▪ Πίνακας Στοιχείων (τύπος όνομα_διαστάσεις στοιχεία)</li> </ul>
Πεδίο 4:	αντίγραφα αντικειμένου(όνομα αντικειμένου, όνομα υπηρεσίας)

Σχήμα 3.1 Η γενική μορφή των αντικειμένων

Οι πληροφορίες που περιέχονται στα αντικείμενα είναι απαραίτητες για λειτουργία του συστήματος. Συγκεκριμένα, το πεδίο 1 είναι απαραίτητο για την αναζήτηση του

αντικειμένου και το πεδίο 2 απαιτείται για τη στήριξη του μηχανισμού κληρονομικότητας, που ενεργοποιείται όταν δε βρεθεί η απαιτούμενη πληροφορία. Το πεδίο 3 περιέχει όλες τις δυνατές πληροφορίες, που μπορεί να αναζητήσει ο πελάτης.

Οι πληροφορίες μπορεί να είναι, είτε ένας ακέραιος αριθμός, του οποίου η αναζήτηση γίνεται βάσει του ονόματός του, είτε ένας πραγματικός αριθμός, του οποίου η αναζήτηση γίνεται επίσης με βάση το όνομα του, είτε ένα αλφαριθμητικό, είτε τέλος ένας πίνακας στοιχείων.

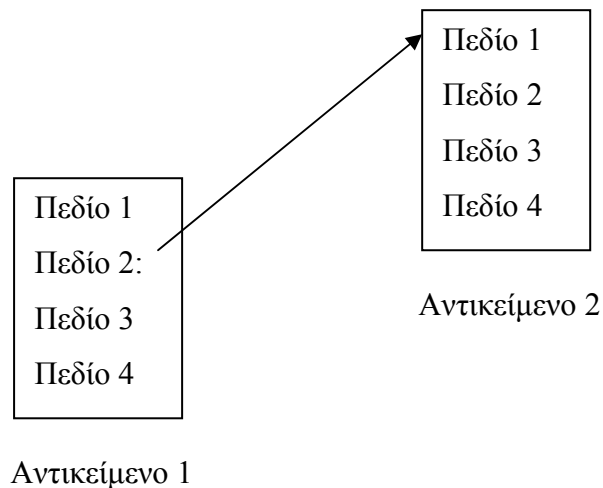
Στην περίπτωση του πίνακα στοιχείων, πρέπει να περιέχονται και οι διαστάσεις του. Ο λόγος είναι η χρησιμοποίησή τους στον έλεγχο των στοιχείων της αίτησης του πελάτη. Ο έλεγχος αυτός θα καθορίσει, αν η απάντηση μπορεί να σταλεί απευθείας ή αν χρειάζεται να συλλεχθούν οι πληροφορίες που υπολείπονται και από άλλα αντικείμενα.

Το πεδίο 4 απαιτείται για την υποστήριξη της λειτουργίας της δημιουργίας αντιγράφων αντικειμένου. Μετά την εκτέλεση αυτής της λειτουργίας, θα πρέπει το πεδίο αυτό να ενημερωθεί για τη δημιουργία των αντιγράφων. Συγκεκριμένα, συμπληρώνεται το όνομα του αντικειμένου και η υπηρεσία όπου δημιουργήθηκε το αντίγραφο του.

Τέλος, η αρχή κάθε αντικειμένου ορίζεται με το χαρακτηριστικό όνομα «object» και το τέλος του με το «end\_object».

### **3.3. Ο Μηχανισμός Κληρονομικότητας**

Έχει αναφερθεί ότι χαρακτηριστικό του προτεινόμενου συστήματος είναι η ιεραρχική οργάνωση των αντικειμένων-δεδομένων και οι αρχές κληρονομικότητας που τα διέπουν. Στο Σχήμα 3.2 φαίνεται ο τρόπος με τον οποίο ένα αντικείμενο επικοινωνεί με κάποιο άλλο.



Σχήμα 3.2 Ο τρόπος επικοινωνίας των αντικειμένων-δεδομένων

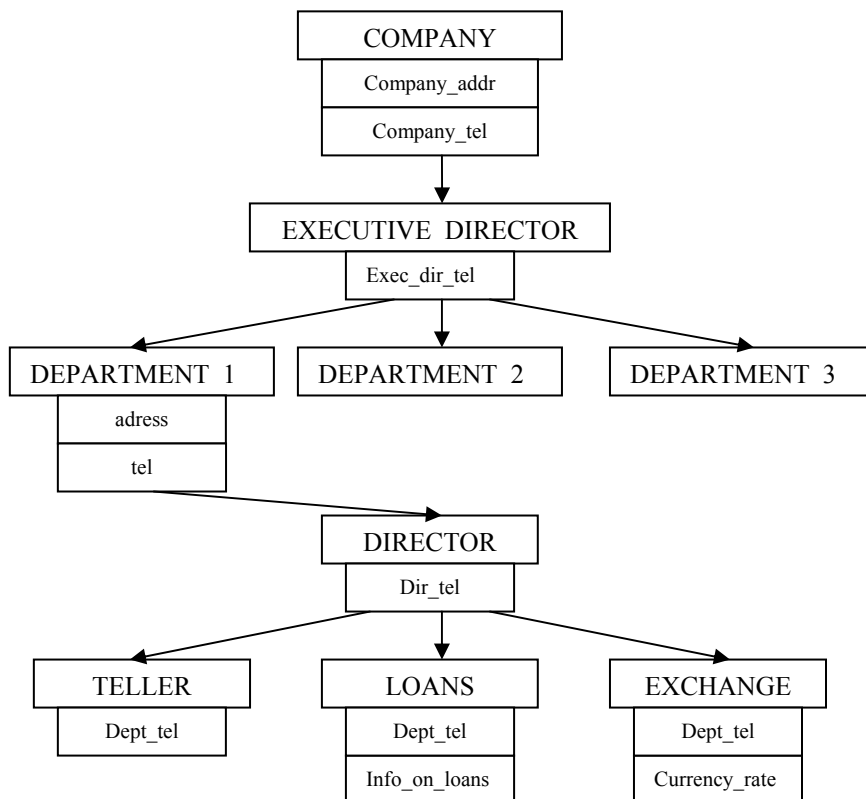
Αναλυτικά, ο μηχανισμός κληρονομικότητας ενεργοποιείται όταν ύστερα από την αίτηση ενός πελάτη προς μια υπηρεσία δεν εντοπιστεί η απαραίτητη πληροφορία. Εν γένει, μια αίτηση-μήνυμα περιλαμβάνει, μαζί με άλλη πληροφορία, το όνομα του αντικειμένου και το όνομα της πληροφορίας, που πρέπει να βρεθεί. Αν βρεθεί το ζητούμενο αντικείμενο αλλά όχι η πληροφορία που αποθηκεύεται στο πεδίο 3 του αντικειμένου, τότε η αίτηση προωθείται στο αντικείμενο-πρόγονο(δηλαδή στο αντικείμενο 2 του Σχήματος 3.2 ). Το πεδίο 2 του αντικειμένου περιέχει το όνομα του αντικειμένου-προγόνου και το όνομα της υπηρεσίας που το περιέχει.

Επιπλέον, ο ίδιος μηχανισμός ενεργοποιείται στην περίπτωση που η πληροφορία εντοπιστεί, αλλά μερικώς. Αυτό συμβαίνει είτε όταν η πληροφορία είναι πίνακας στοιχείων και η ζητούμενη πληροφορία υπερβαίνει τις διαστάσεις του πίνακα, είτε όταν ζητούνται περισσότερα από ένα είδη πληροφορίας σε μια αίτηση. Στην περίπτωση του πίνακα στοιχείων, η προώθηση της αίτησης έχει σκοπό την εύρεση της πληροφορίας(γραμμές και στήλες) που απομένει. Τελικά, και στις δύο περιπτώσεις ο πελάτης λαμβάνει συνολική απάντηση.

Ως παράδειγμα εφαρμογής του προτεινόμενου συστήματος παραθέτουμε τη λειτουργία μια εταιρείας. Συγκεκριμένα, υποθέτουμε ότι τα αντικείμενα δεδομένων περιέχουν χρήσιμες πληροφορίες για τη λειτουργία μιας τράπεζας. Μέρος του γραφήματος της ιεραρχίας φαίνεται στο Σχήμα 3.3, όπου φαίνονται επίσης μερικά μέρη των αντικειμένων δεδομένων. Όλα τα αντικείμενα μπορούν να απαντήσουν σε

κλήσεις για πρόσβαση σε δεδομένα που βρίσκονται μέσα στα αντικείμενα ή σε αντικείμενα προγόνους. Το αντικείμενο που περιγράφει τον «TELLER» μπορεί να απαντήσει στην ερώτηση «telephone», όπως επίσης και στην ερώτηση «address», παρόλο που αυτή η πληροφορία δεν ανήκει στο συγκεκριμένο αντικείμενο.

Έστω ότι στέλνεται μια αίτηση στον «TELLER», που αφορά τον αριθμό τηλεφώνου του τμήματος. Στην περίπτωση αυτή ο «TELLER» μπορεί να απαντήσει άμεσα αφού έχει αυτή την πληροφορία. Όμως, αν η αίτηση-ερώτηση αφορά στην διεύθυνση τότε θα πρέπει να προωθήσει την αίτηση πιο ψηλά στην ιεραρχία μέχρι κάποιο αντικείμενο να απαντήσει. Σε άλλα συστήματα για να εξυπηρετηθεί η αίτηση θα έπρεπε να αντιγραφεί η πληροφορία στο συγκεκριμένο αντικείμενο ή να γίνει μεταφορά του αντικειμένου. Η διαφορά του προτεινόμενου συστήματος είναι ότι δεν χρειάζεται να γίνει αυτό, αντίθετα η πληροφορία λαμβάνεται με τη χρήση της ιεραρχικής οργάνωσης των αντικειμένων πληροφοριών και του μηχανισμού κληρονομικότητας.



Σχήμα 3.3. Η δομή μιας εταιρείας

### 3.4. Λειτουργικά Στοιχεία του Συστήματος

Στις προηγούμενες παραγράφους περιγράφηκαν τα στοιχεία που συνθέτουν το προτεινόμενο σύστημα. Ακολουθεί μια συνολική παράθεση των λειτουργιών, των δυνατοτήτων του και των απαιτήσεων του αναφορικά με την εφαρμογή του παραδείγματος με τη δομή μιας εταιρείας (Σχήμα 3.3).

Τα αντικείμενα βρίσκονται εκεί όπου ανήκουν γεωγραφικά, δηλαδή στην αντίστοιχη υπηρεσία. Στο παράδειγμα με τη δομή της εταιρείας το αντικείμενο «TELLER» βρίσκεται στην υπηρεσία διαδικτύου που το προσδιορίζει γεωγραφικά.

Οι κλήσεις γίνονται εκεί όπου είναι λογικό να γίνουν ανεξάρτητα από το που βρίσκεται η πληροφορία που αναζητείται. Η κλήση για τη διεύθυνση ενός τμήματος είναι λογικό να γίνει στο αντικείμενο «TELLER» δεδομένου του ρόλου που έχει αυτό το αντικείμενο, χωρίς αυτό να σημαίνει ότι περιέχει απαραίτητα την πληροφορία.

Η πληροφορία είναι κατανεμημένη, αλλά στο χρήστη εμφανίζεται ενιαία. Μια αίτηση μπορεί να απαιτεί τη συγκέντρωση πληροφορίας από διάφορα αντικείμενα, τελικά όμως ο χρήστης θα λάβει συνολική απάντηση. Για παράδειγμα μπορεί να ζητηθεί το τηλέφωνο από το αντικείμενο «TELLER» και το τηλέφωνο από το αντικείμενο «DIRECTOR» και να επιστραφεί συνολική απάντηση και με τα δύο τηλέφωνα, χωρίς ο χρήστης να αντιληφθεί την αναζήτηση της πληροφορίας σε παραπάνω από ένα αντικείμενα.

Κάθε αίτηση αντιλαμβάνεται διαφορετικά την πληροφορία ανάλογα με το αντικείμενο στο οποίο απευθύνεται. Για παράδειγμα μια αίτηση για την εύρεση του τηλεφώνου στο αντικείμενο «DEPARTMENT\_1» θα επιστρέψει διαφορετική απάντηση από μια αντίστοιχη αίτηση στο αντικείμενο «TELLER» παρόλο που η πληροφορία έχει τη ίδια ταυτότητα.

Οι σχέσεις των αντικειμένων αντανακλούν τον πραγματικό κόσμο και τις σχέσεις που εκεί τα διέπουν. Στο Σχήμα 3.3 οι σχέσεις κληρονομικότητας των αντικειμένων(εργαζομένων) αντιπροσωπεύουν την ιεραρχία που πραγματικά υπάρχει σε μια οργανωμένη εταιρεία.

Η πληροφορία μπορεί να επικαλύπτεται. Αυτό σημαίνει ότι η ίδια πληροφορία για τη διεύθυνση υπάρχει σε διαφορετικά αντικείμενα, αλλά έχει διαφορετικές τιμές σε κάθε αντικείμενο.

Η πληροφορία μπορεί να μοιράζεται, δηλαδή να έχουν πρόσβαση σε αυτή διάφορα αντικείμενα χρησιμοποιώντας τις σχέσεις κληρονομικότητας που τα διέπουν.

Η ιεραρχία μπορεί να αλλάζει ακόμα και κατά την εκτέλεση χωρίς να επηρεάζεται η λειτουργία του συστήματος, για παράδειγμα κατά τη μετακίνηση αντικειμένου από μια υπηρεσία διαδικτύου σε κάποια άλλη.

Η ιεραρχία μπορεί να χρησιμοποιηθεί κατά τη δημιουργία των αντιγράφων αντικειμένων και κατά τη μετακίνηση αντικειμένων. Συγκεκριμένα, εκμεταλλευόμαστε τους μηχανισμούς της κληρονομικότητας για την υλοποίηση αυτών των λειτουργιών.



## ΚΕΦΑΛΑΙΟ 4. ΥΛΟΠΟΙΗΣΗ

---

4.1. Πλατφόρμες και Προγραμματιστικά Εργαλεία

4.2. Λεπτομέρειες Υλοποίησης

4.2.1. Περιγραφή του Είδους της Ιεραρχίας

4.2.2. Αναλυτική Περιγραφή των Εκτελούμενων Λειτουργιών

---

### 4.1. Πλατφόρμες και Προγραμματιστικά Εργαλεία

Για την ανάπτυξη των υπηρεσιών χρησιμοποιήθηκαν τα ακόλουθα πακέτα λογισμικού:

- jakarta-tomcat (έκδοση 4.1.31)

Ο Apache Tomcat είναι μια υλοποίηση των τεχνολογιών Java Servlet και JavaServer Pages. Οι προδιαγραφές Java Servlet και Java Server Pages αναπτύσσονται κάτω από την επεξεργασία της κοινότητας της Java.

Ο Apache Tomcat έχει αναπτυχθεί σε ένα ανοιχτό και πολυσυμμετοχικό περιβάλλον και κυκλοφορεί κάτω από την επίβλεψη του λογισμικού Apache. Ο Apache Tomcat προορίζεται να είναι μια συνεργασία των καλύτερων υπεύθυνων για την ανάπτυξη λογισμικού στον κόσμο.

Διαθέτει πολυάριθμες και μεγάλης κλίμακας, σημαντικές εφαρμογές διαδικτύου σε διάφορες βιομηχανίες και οργανισμούς.

- Η πλατφόρμα Axis (έκδοση 1.1)

Ακολουθεί μια σύντομη περιγραφή της αρχιτεκτονικής της Axis που θα βοηθήσει να αναδειχθούν τα σημαντικά σημεία στο σχεδιασμό μας.

Το σημείο κατάληξης ενός μηνύματος ορίζεται σαν ο μηχανισμός, που χρησιμοποιεί μια εφαρμογή για να συνδεθεί σε ένα κανάλι μηνύματος προκειμένου να στείλει και να λάβει μηνύματα. Στις εφαρμογές μας το πλαίσιο Axis αναπαριστά το κανάλι μηνύματος και η κύρια λειτουργία του είναι να διαχειρίζεται την προώθηση των μηνυμάτων εκ μέρους του χρήστη.

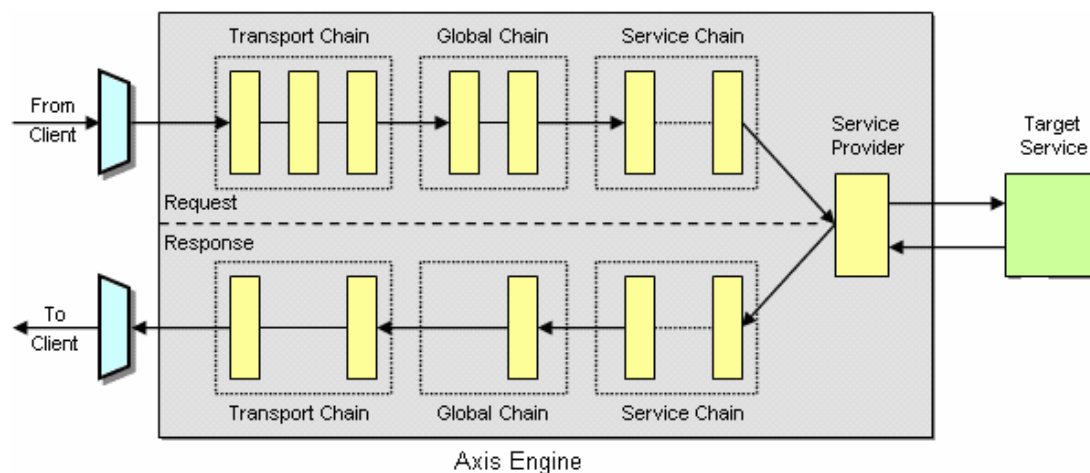
Ο εξυπηρετητής Axis υλοποιεί το πρότυπο της υπηρεσίας ενεργοποίησης. Μια υπηρεσία ενεργοποίησης μπορεί να συνδέσει ένα κανάλι μηνύματος με μια σύγχρονη υπηρεσία σε μια εφαρμογή, έτσι ώστε όταν ληφθεί ένα μήνυμα να κληθεί η υπηρεσία.

Ο προσαρμογέας μηνύματος υλοποιείται μέσα στον εξυπηρετητή Axis έτσι ώστε ο υπεύθυνος για την ανάπτυξη να μην χρειάζεται να υλοποιήσει αυτή τη λειτουργία. Για το λόγο αυτό, ο κώδικας της εφαρμογής περιέχει μόνο επιχειρησιακή λογική για την εφαρμογή και ο εξυπηρετητής Axis αναλαμβάνει τις υπηρεσίες που έχουν να κάνουν με τη διαχείριση μηνυμάτων.

Το προγραμματιστικό μοντέλο του πελάτη της Axis παρέχει τα στοιχεία για την εφαρμογή πελάτη προκειμένου να κληθεί ένα σημείο κατάληξης URL και στη συνέχεια να λάβει το απαντητικό μήνυμα από τον εξυπηρετητή. Οι πελάτες στις εφαρμογές μας βασίζονται σε σύγχρονη λειτουργία που χρησιμοποιεί το προγραμματιστικό μοντέλο του πελάτη της Axis. Μέσα στην εξυπηρετητή βρίσκεται ένας ελεγκτής για κάθε πρωτόκολλο μεταφοράς που υποστηρίζεται από τον εξυπηρετητή. Όταν ο πελάτης στέλνει ένα μήνυμα στο σημείο κατάληξης, ένας ελεγκτής μεταφοράς μέσα στο πλαίσιο υλοποίησης της Axis δημιουργεί ένα αντικείμενο περιεχομένου και το περνά μέσα από μια αλυσίδα αιτήσεων στο πλαίσιο υλοποίησης της Axis. Το περιεχόμενο μηνύματος περιέχει το πραγματικό μήνυμα που έχει ληφθεί από τον πελάτη, μαζί με άλλες ιδιότητες που προστέθηκαν κατά τη μεταφορά.

Το πλαίσιο υλοποίησης της Axis αποτελείται από ένα σύνολο χειριστών, που καλούνται με μια συγκεκριμένη σειρά εξαρτώμενη από τη διαμόρφωση υλοποίησης και από το αν ο πελάτης ή ο εξυπηρετητής χρησιμοποιεί το πλαίσιο

υλοποίησης. Οι χειριστές είναι του υποσυστήματος για τη ροή μηνυμάτων και ομαδοποιούνται σε αλυσίδες. Τα μηνύματα αιτήσεων προωθούνται μέσα από μια σειρά χειριστών αιτήσεων σε μια αλυσίδα. Οποιαδήποτε απαντητικά μηνύματα αποστέλλονται στη συγκεκριμένη αλυσίδα απάντησης διαμέσου μιας ακολουθίας από χειριστές απαντήσεων.



Σχήμα 4.1 Ο μηχανισμός της Axis

Η παραπάνω εικόνα δείχνει μια αναπαράσταση υψηλού επιπέδου των εσωτερικών διεργασιών της πλατφόρμας της Axis.

Η Axis αποτελείται από αρκετά υποσυστήματα που συνεργάζονται για να παρέχουν τη λειτουργικότητα του καναλιού μηνυμάτων. Αυτό το πλαίσιο είναι διαθέσιμο να χρησιμοποιηθεί τόσο από την εφαρμογή πελάτη όσο και από την εφαρμογή εξυπηρετητή.

Τα σχετικά υποσυστήματα για τις εφαρμογές μας είναι τα παρακάτω:

- Το υποσύστημα Μοντέλο Μηνύματος που προσδιορίζει την XML σύνταξη των μηνυμάτων SOAP.
- Το υποσύστημα Ροής μηνυμάτων που προσδιορίζει τους χειριστές και τις αλυσίδες για το πέρασμα των μηνυμάτων.
- Το υποσύστημα Υπηρεσία που προσδιορίζει το χειριστή της υπηρεσίας (SOAP, XML-RPC).

- Το υποσύστημα Μεταφοράς που παρέχει εναλλακτικές λύσεις για τη μεταφορά των μηνυμάτων (για παράδειγμα., HTTP, JMS, SMTP).
- Το υποσύστημα Παροχέας που προσδιορίζει τους παροχείς για διαφορετικούς τύπους κλάσεων (για παράδειγμα., java:RPC, EJB, MDB).

Όπως αναφέρθηκε νωρίτερα, ο υπεύθυνος για την ανάπτυξη λογισμικού πρέπει να εστιάσει στη δημιουργία μιας εφαρμογής η οποία εμπεριέχει την επιχειρησιακή λογική, η οποία στη συνέχεια μπορεί να εφαρμοστεί στον εξυπηρετητή της Axis. Υπάρχουν τρεις τεχνικές για την εφαρμογή μιας κλάσης σε γλώσσα προγραμματισμού java σε υπηρεσία διαδικτύου και στη συνέχεια να είναι διαθέσιμη να κληθεί και είναι οι παρακάτω:

- Αυτόματη εφαρμογή.
- Χρησιμοποιώντας έναν Περιγραφέα Υλοποίησης υπηρεσιών διαδικτύου.
- Δημιουργώντας proxies από ένα υπάρχον WSDL έγγραφο.

Ο πρώτος και πιο απλός τρόπος είναι να γραφτεί η κλάση που περιέχει την επιχειρησιακή λογική ως ένα αρχείο σε Java υπηρεσία διαδικτύου ( αυτό είναι ένα αρχείο με τον κώδικα σε Java και με κατάληξη \*.jws). Οι μέθοδοι αυτής της κλάσης περιέχουν την επιχειρησιακή λογική. Το αρχείο JWS δε χρειάζεται να μεταγλωττιστεί και μπορεί απευθείας να εφαρμοστεί αντιγράφοντας το αρχείο στον φάκελο webapps στον εξυπηρετητή. Κάθε δημόσια μέθοδος είναι από το σημείο αυτό και μετά προσβάσιμη ως υπηρεσία διαδικτύου. Το όνομα αυτού του JWS αρχείου αποτελεί τμήμα του τελικού σημείου που ο εξυπηρετητής Axis αποκαλύπτει ως υπηρεσία διαδικτύου, όπως φαίνεται παρακάτω:

<http://hostname:portnumber/axis/exampleservice.jws>

Η Axis 1.1 δημιουργεί αυτόματα το WSDL έγγραφο(Web Services Description Language) από τις υπηρεσίες που υλοποιούνται μέσα στον εξυπηρετητή. Η περιγραφή WSDL είναι ένα XML σχήμα που περιγράφει τη δημόσια διεπαφή μιας υπηρεσίας διαδικτύου (π.χ. τις μεθόδους που περιγράφονται μέσα από τη διεπαφή), όπως επίσης και την τοποθεσία της υπηρεσίας (π.χ. το URL). Η αυτόματη δημιουργία της WSDL περιγραφής επιτρέπει σε άλλες εφαρμογές να περιεργαστούν τη δημόσια και απομακρυσμένη διεπαφή που παρέχεται από την

κλάση της υπηρεσίας διαδικτύου. Μπορεί επίσης, να χρησιμοποιηθεί για την αυτόματη δημιουργία των κλάσεων του πελάτη, που εμπεριέχουν την κλήση σε υπηρεσίες διαδικτύου μέσα σε μια συνηθισμένη Java κλάση. Το μειονέκτημα αυτής της μεθόδου είναι ότι ο υπεύθυνος για την ανάπτυξη δεν μπορεί να ελέγξει τις παραμέτρους της υλοποίησης.

Η δεύτερη τεχνική υλοποιεί μια μεταγλωττισμένη κλάση χρησιμοποιώντας έναν WSDD (Περιγραφέα Ανάπτυξης υπηρεσιών διαδικτύου), ο οποίος επιτρέπει στον υπεύθυνο ανάπτυξης να ελέγξει τις παραμέτρους ανάπτυξης, π.χ. την εμβέλεια της κλάσης. Εξ'ορισμού μια κλάση υλοποιείται στην εμβέλεια που έχει ζητηθεί π.χ. ένα καινούριο στιγμιότυπο της κλάσης αρχικοποιείται για κάθε αίτηση που λαμβάνεται. Όταν τελειώσει η διαδικασία το στιγμιότυπο καταστρέφεται. Αν η κλάση χρειάζεται να παραμείνει για να εξυπηρετήσει πολλαπλές αιτήσεις από τον ίδιο πελάτη για κάποιο διάστημα, τότε πρέπει να οριστεί η κλάση για την εμβέλεια αυτού του διαστήματος.

Η τελευταία τεχνική είναι πιο περίπλοκη από τις δύο προηγούμενες τεχνικές, αλλά επιτρέπει τη δημιουργία proxies και skeletons από ένα υπάρχον WSDL έγγραφο (με τη χρήση του εργαλείου wsdl2java). Οι proxies and skeletons εσωκλείουν ολόκληρο τον κώδικα που σχετίζεται με το SOAP έτσι ώστε ο υπεύθυνος για την ανάπτυξη του λογισμικού να μην χρειάζεται να γράψει κάποιο κώδικα που σχετίζεται με SOAP (ή Axis), αλλά μπορεί να μπορεί να ενσωματώσει την επιχειρησιακή λογική απευθείας μέσα στο σώμα των μεθόδων που δημιουργούνται.

Επιλέξαμε να υλοποιήσουμε όλες τις υπηρεσίες διαδικτύου χρησιμοποιώντας την τεχνική της αυτόματης υλοποίησης με τα αρχεία JWS, προκειμένου να κρατήσουμε το σχεδιασμό και τις απαιτήσεις της υλοποίησης όσο πιο απλές γίνεται. Από την πλευρά του πελάτη είναι πιο εύκολο να γραφτεί άμεσα ο κώδικας για τις κλήσεις προς τις υπηρεσίες. Θα μπορούσε να χρησιμοποιηθεί και το εργαλείο wsdl2java για τη δημιουργία των stubs, που θα μπορούσαν στη συνέχεια να κληθούν από τον πελάτη, αλλά προσπαθήσαμε να μετριάσουμε την αυτόματη δημιουργία κώδικα διότι θα μπορούσε να κάνει πιο δύσκολη τη δημιουργία των παραδειγμάτων

## Ανακάλυψη υπηρεσιών

Προκειμένου να κληθεί μια υπηρεσία που υλοποιείται σε έναν εξυπηρετητή, κάθε εφαρμογή πελάτη πρέπει να γνωρίζει το τελικό σημείο URL. Στο μοντέλο των υπηρεσιών διαδικτύου, μια εφαρμογή πελάτη θα αναζητήσει την τοποθεσία μιας υπηρεσίας διαδικτύου που χρειάζεται να καλέσει σε ένα κοινό καταχωρητή υπηρεσιών.

Το UDDI είναι ένα τέτοιο πρότυπο που χρησιμοποιείται για καταχώρηση. Στις εφαρμογές μας γράφουμε το τελικό σημείο στον κώδικα του πελάτη. Ωστόσο, για να διευκολύνουμε την υλοποίηση δημιουργούμε αρχεία ιδιοτήτων για τις εφαρμογές του εξυπηρετητή. Τα αρχεία ιδιοτήτων περιέχουν τα ονόματα των υπηρεσιών που καλούνται κατά περίπτωση. Αυτό προσφέρει κάποια ευελιξία στην υλοποίηση των εφαρμογών και σε άλλο περιβάλλον.

Σε οποιοδήποτε κατανεμημένο υπολογιστικό πλαίσιο υλοποίησης είτε είναι Java RMI, CORBA ή SOAP υπηρεσίες διαδικτύου, χρειάζεται να οριστούν οι παράμετροι των μεθόδων που καλούνται σε απομακρυσμένα αντικείμενα. Αυτές οι παράμετροι είναι οι ιδιότητες των μηνυμάτων-αντικειμένων που αποστέλλονται από τον πελάτη στον εξυπηρετητή. Γενικά, για να κρατήσουμε τη λύση απλή στις εφαρμογές, χρησιμοποιούμε ένα αλφαριθμητικό για να επιστρέψουμε την απάντηση στον πελάτη από τον εξυπηρετητή. Αν οι παράμετροι με τις οποίες γίνονται οι κλήσεις είναι καθορισμένοι τύποι δεδομένων, (π.χ. int, double, κ.τ.λ..) πρέπει υλοποιηθούν με αντίστοιχα προκαθορισμένα αντικείμενα (για παράδειγμα Integer, Double, κ.τ.λ.).

- Java JDK (έκδοση jdk1.5.0\_06)

Είναι ένα λογισμικό ανάπτυξης για την παραγωγή προγραμμάτων σε Java. Έχει αναπτυχθεί από τη Sun Microsystem's JavaSoft.

Τα παραπάνω πακέτα λογισμικού εγκαταστάθηκαν σε λειτουργικό σύστημα Windows XP Professional, όπου και αναπτύχθηκαν οι αντίστοιχες εφαρμογές.

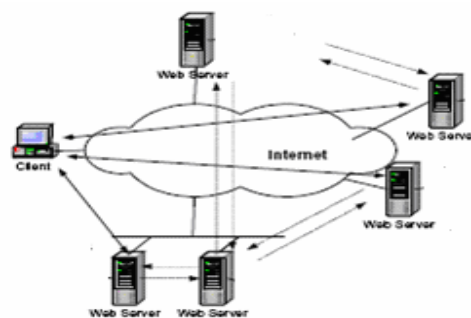
## 4.2. Λεπτομέρειες Υλοποίησης

Στο προτεινόμενο σύστημα, η ιεραρχία καθορίζεται από τα αντικείμενα των υπηρεσιών. Ένας χρήστης μπορεί να αρχικοποιήσει μια αίτηση από την αντίστοιχη υπηρεσία-πελάτη. Όταν η υπηρεσία δεν μπορεί να παρέχει την λειτουργία που έχει ζητηθεί, αποστέλλει την αίτηση σε μια πιο υψηλού επιπέδου υπηρεσία, που καθορίζεται από την ιεραρχία που παρέχει η κληρονομικότητα των αντικειμένων, προκειμένου να απαντηθεί επιτυχώς η αίτηση.

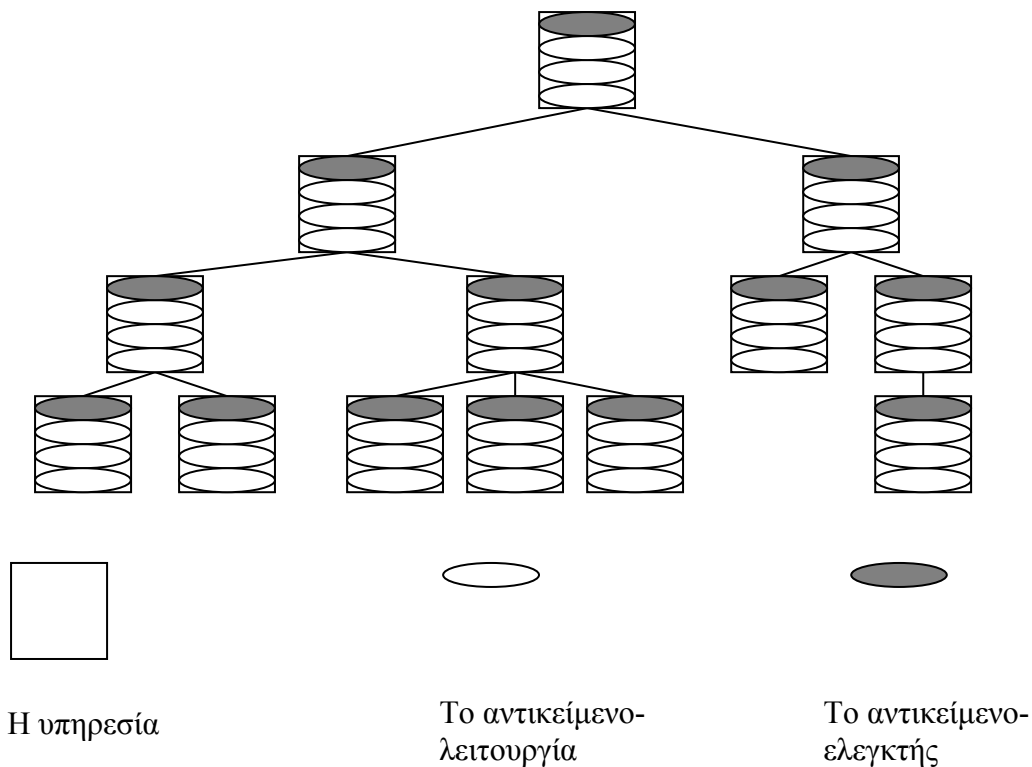
### 4.2.1 Περιγραφή του Είδους της Ιεραρχίας

Υλοποιούμε μια ιεραρχία που υλοποιείται από τα αντικείμενα (Object Level Hierarchy). Αναλυτικά, κάθε υπηρεσία μπορεί να περιέχει μεθόδους-αντικείμενα κάθε μια από τις υλοποιεί κάποια συγκεκριμένη λειτουργία. Η λειτουργία αυτή μπορεί να είναι μια από τις τέσσερις βασικές λειτουργίες, που υποστηρίζει το σύστημα μας και είναι η ανάγνωση πληροφορίας, η εγγραφή πληροφορίας, η παραγωγή αντίγραφου αντικειμένου και η μετακίνηση αντικειμένου σε μια απομακρυσμένη υπηρεσία. Οι πληροφορίες που περιέχουν τα αντικείμενα καθορίζονται από αρχεία εισόδου. Στην προσπέλαση αυτών των αρχείων και κατά συνέπεια και των πληροφοριών που περιέχουν, εστιάζεται η όλη εφαρμογή των παρεχόμενων λειτουργιών του συστήματος.

Στο Σχήμα 4.2 και στο Σχήμα 4.3 που ακολουθούν φαίνεται ο τρόπος επικοινωνίας των υπηρεσιών.



Σχήμα 4.2 Η επικοινωνία μεταξύ των υπηρεσιών



Σχήμα 4.3 Η ιεραρχία που δημιουργεί το σύστημα

Στο σημείο αυτό απαιτείται περαιτέρω ανάλυση του τρόπου, που υλοποιούμε την ιεραρχία, που περιγράφηκε παραπάνω. Έχει αναφερθεί ότι η βασική λειτουργία του συστήματος μας βασίζεται στη αποστολή αιτήσεων και τη λήψη απαντήσεων.

Συγκεκριμένα, όταν η αίτηση προς μια υπηρεσία είναι επιτυχημένη, δηλαδή όταν έχει βρεθεί ζητούμενο αντικείμενο, συνεχίζεται η αναζήτηση της συγκεκριμένης πληροφορίας μέσα στο αντικείμενο. Αν η πληροφορία βρεθεί τότε αποστέλλεται η απάντηση στον πελάτη που τη ζήτησε. Αν η πληροφορία δε βρεθεί, ενεργοποιείται ο μηχανισμός κληρονομικότητας, που περιλαμβάνει την ανάγνωση της πληροφορίας που βρίσκεται στο πεδίο Inherits του αντικειμένου και την αποστολή της αίτησης στα περιεχόμενα του πεδίου. Τα περιεχόμενα του πεδίου περιλαμβάνουν το όνομα του αντικειμένου και το όνομα της υπηρεσίας που θα αποσταλεί η αίτηση .

Αναλύθηκε η διαδικασία που ακολουθείται όταν η αίτηση είναι επιτυχημένη, όμως όταν η αίτηση προς μια υπηρεσία αποτύχει, δηλαδή δεν βρεθεί το ζητούμενο αντικείμενο, γίνονται τα εξής: καλείται η υπηρεσία Location\_Server, που περιέχει την



τοποθεσία όλων των αντικειμένων. Η υπηρεσία αυτή αποστέλλει απευθείας την αίτηση, στην υπηρεσία που βρέθηκε ότι ανήκει το αντικείμενο που ζητήθηκε.

#### 4.2.2 Αναλυτική Περιγραφή των Εκτελούμενων Λειτουργιών

Μια αίτηση για **ανάγνωση** δεδομένων αποστέλλεται από τον πελάτη. Η αίτηση μπορεί να αφορά στην ανάγνωση των εξής τύπων δεδομένων: ακέραιοι, πραγματικοί, αλφαριθμητικά και σύνθετες δομές, όπως είναι οι πίνακες στοιχείων. Η αίτηση αποστέλλεται σαν ένα αντικείμενο δεδομένων `String ret = (String) call.invoke(new Object[] {s1, s2, s3, s4, s5})` και η απάντηση επιστρέφεται με τη μορφή αλφαριθμητικού. Τα στοιχεία που συνθέτουν το αντικείμενο της αίτησης, δηλαδή τα `s1, s2, s3, s4, s5` είναι οι πληροφορίες για τη διαδρομή που θα ακολουθήσει η αίτηση προκειμένου να συμπληρωθεί η απάντηση (Σχήμα 4.4):

<code>String s1 = new String(args[0]);</code>	Το όνομα της υπηρεσίας
<code>String s2 = new String(args[1]);</code>	Το όνομα του αντικειμένου
<code>String s3 = new String(args[2]);</code>	Οι πληροφορίες ανάγνωσης
<code>String s4 = new String(args[3]);</code>	Το είδος της λειτουργίας
<code>String s5 = new String(args[4]);</code>	Το αντικείμενο ελεγκτής
<code>String ret = (String) call.invoke (new Object[] {s1, s2, s3, s4, s5});</code>	

Σχήμα 4.4 Τα περιεχόμενα της αίτησης ανάγνωσης

Το αλφαριθμητικό, που εκφράζει τις πληροφορίες, που πρέπει να διαβαστούν, περιέχει συγκεκριμένα, το όνομα της πληροφορίας ή το όνομα της πληροφορίας και τις συντεταγμένες των στοιχείων του πίνακα, στην περίπτωση που πρέπει να διαβαστούν στοιχεία πίνακα. Υπάρχει η δυνατότητα να σταλούν σε ένα μήνυμα και δύο πληροφορίες που πρέπει να διαβαστούν. Μετά την εύρεση της πρώτης συνεχίζεται η αναζήτηση, χρησιμοποιώντας το μηχανισμό κληρονομικότητας, για την εύρεση και της επόμενης. Τελικά, το μήνυμα-απάντηση συμπληρώνεται συνολικά και αποστέλλεται στον πελάτη. Στόχος αυτής της δυνατότητας είναι να εντοπίζονται με μια αίτηση δεδομένα που είναι καταναμημένα σε διαφορετικές υπηρεσίες.

Αναφέρθηκε παραπάνω, ότι ανάμεσα στις μεθόδους-αντικείμενα που διαδραματίζουν κάποιο ρόλο στη λειτουργία του συστήματος είναι και η μέθοδος ελεγκτής. Η μέθοδος αυτή είναι η πρώτη μέθοδος, η οποία καλείται στην πλευρά της υπηρεσίας σε όλες τις αιτήσεις. Αναλαμβάνει τον έλεγχο της ύπαρξης του ζητούμενου αντικειμένου στο αρχείο των αντικειμένων, που διαθέτει η συγκεκριμένη υπηρεσία. Αν βρεθεί το ζητούμενο αντικείμενο, καλείται η αντίστοιχη μέθοδος που το υλοποιεί. Η κλήση της μεθόδου γίνεται δυναμικά, χρησιμοποιώντας τις δυνατότητες, που παρέχει η Java, για τη δυναμική κλήση μεθόδων, όπως φαίνεται στο Σχήμα 4.5:

```
Class c = this.getClass();
String methodCall = ObjectName;
Method method = c.getMethod(methodCall, new Class[] {String.class,
String.class, String.class, String.class, String.class} );
ret_str = (String)method.invoke(this, new Object[] {ServiceName, ObjectName,
ReadInfo, Operation, CheckObject } );
```

Σχήμα 4.5 Η κλήση της μεθόδου με δυναμικό τρόπο

Στην περίπτωση που ο έλεγχος για την ύπαρξη του αντικειμένου αποτύχει, καλείται η υπηρεσία Location\_Server, που περιέχει όλα τα αντικείμενα και τις υπηρεσίες όπου ανήκουν. Όταν βρεθεί το αντικείμενο στο αρχείο της υπηρεσίας αυτής, η αίτηση δρομολογείται ξανά, αλλά από την υπηρεσία Location\_Server.

Πρέπει να αναφερθεί, ότι κατά την ανάγνωση στοιχείων πινάκων, αν οι δοθείσες διαστάσεις είναι μεγαλύτερες από τις διαθέσιμες στο συγκεκριμένο αντικείμενο, η αναζήτηση συνεχίζεται και σε άλλα αντικείμενα, χρησιμοποιώντας το μηχανισμό κληρονομικότητας.

Μια αίτηση για **εγγραφή** δεδομένων αποστέλλεται ομοίως από τον πελάτη. Η αίτηση έχει τη μορφή αντικειμένου, αφού περιλαμβάνει ένα σύνολο από πληροφορίες, που απαιτούνται για την αποστολή της, την παραλαβή από την κατάλληλη υπηρεσία, την αναζήτηση αρχικά του αντικειμένου και στη συνέχεια της απαραίτητης πληροφορίας και την τροποποίησή της (Σχήμα 4.6).

String s1 = new String(args[0]);	Το όνομα της υπηρεσίας
String s2 = new String(args[1]);	Το όνομα του αντικειμένου
String s3 = new String(args[2]);	Οι πληροφορίες ανάγνωσης-εγγραφής
String s4 = new String(args[3]);	Το είδος της λειτουργίας
String s5 = new String(args[4]);	Το αντικείμενο ελεγκτής
String ret = (String) call.invoke (new Object[] {s1, s2, s3, s4, s5});	

Σχήμα 4.6 Τα περιεχόμενα της αίτησης εγγραφής

Στο αλφαριθμητικό s3 υπάρχει η κωδικοποίηση της πληροφορίας που πρέπει να διαβαστεί και μετά να τροποποιηθεί και η αποκωδικοποίηση γίνεται από την υπηρεσία, όταν την παραλάβει. Οι υπόλοιπες διαδικασίες εκτελούνται κατά όμοιο τρόπο με τη λειτουργία ανάγνωσης.

Η αίτηση για **αντιγραφή** αντικειμένου περιλαμβάνει την παρακάτω απαιτούμενη πληροφορία (Σχήμα 4.7).

String s1 = new String(args[0]);	Το όνομα της υπηρεσίας
String s2 = new String(args[1]);	Το όνομα του αντικειμένου
String s3 = new String(args[2]);	Το όνομα της υπηρεσίας-προορισμός
String s4 = new String(args[3]);	Το είδος της λειτουργίας
String s5 = new String(args[4]);	Το αντικείμενο ελεγκτής
String ret = (String) call.invoke (new Object[] {s1, s2, s3, s4, s5});	

Σχήμα 4.7 Τα περιεχόμενα της αίτησης αντιγραφής αντικειμένου

Το διαφορετικό στοιχείο είναι η ύπαρξη της πληροφορίας για την υπηρεσία που θα δημιουργηθεί το αντίγραφο του αντικειμένου, το όνομα του οποίου όπως και η υπηρεσία που ανήκει, αποστέλλονται με την αίτηση.

Μετά την εύρεση του αντικειμένου, συλλέγονται οι πληροφορίες που το αποτελούν και αποθηκεύονται σε ένα αλφαριθμητικό. Το αλφαριθμητικό αποστέλλεται στην υπηρεσία προορισμός, όπου και συνεχίζεται η ολοκλήρωση της διαδικασίας. Επιπλέον, ενημερώνεται το αρχείο των αντικειμένων(στο πεδίο replica) για την ύπαρξη αντίγραφου του αντικειμένου σε κάποια άλλη υπηρεσία

Για την εφαρμογή αυτής της λειτουργίας δημιουργήθηκε η μέθοδος public String CopyObject( ), η οποία καλείται από την υπηρεσία-πηγή, αλλά υλοποιείται στην υπηρεσία-προορισμός. Η μέθοδος αυτή αναλαμβάνει να ενημερώσει τα αντίστοιχα αρχεία(δηλαδή, το αρχείο με τις συνολικές πληροφορίες των αντικειμένων και το αρχείο που χρησιμοποιεί η μέθοδος ελεγκτής για την αντιστοίχιση υπηρεσιών και αντικειμένων), που αντιστοιχούν στην υπηρεσία προορισμός, για το νέο αντικείμενο που προστέθηκε.

Πρέπει να σημειωθεί, ότι σε περίπτωση που το αντικείμενο δε βρεθεί, καλείται η υπηρεσία Location\_Server και συνεχίζεται η διαδικασία. Αν το αντικείμενο δε βρεθεί στο συνολικό αρχείο αντικειμένων, τότε αποστέλλεται αρνητική απάντηση στον πελάτη.

Εκτός από τη δημιουργία των αντιγράφων-αντικειμένων πρέπει να καθοριστεί και η συμπεριφορά τους στις αιτήσεις του πελάτη. Συγκεκριμένα, σε αιτήσεις ανάγνωσης το αντικείμενο-αντίγραφο απαντά απευθείας, σε αντίθεση με τις αιτήσεις εγγραφής όπου η αίτηση προωθείται στο αρχικό αντικείμενο, που αναλαμβάνει την εγγραφή της πληροφορίας του και ενημερώνει όλα τα αντίγραφα-αντικείμενα για την τροποποίηση που ζητήθηκε από την υπηρεσία πελάτη. Στην περίπτωση αυτή γίνεται ειδικός χειρισμός προκειμένου να διαχωριστούν οι αιτήσεις ενημέρωσης από το αντικείμενο-πρόγONO, από τις αιτήσεις που αποστέλλουν οι υπηρεσίες-πελάτες και πρέπει να απαντηθούν.

Για την υλοποίηση της λειτουργίας αυτής χρησιμοποιείται όμοιος μηχανισμός με αυτόν της προώθησης κλήσεων. Το αρχικό αντικείμενο εντοπίζεται από το αντίγραφο μέσω ειδικού πεδίου πληροφορίας που δημιουργείται για το σκοπό αυτό.

Η αίτηση για **μετακίνηση** αντικειμένου περιλαμβάνει όμοιες πληροφορίες με αυτές που απαιτούνται για τη δημιουργία αντίγραφου αντικειμένου (Σχήμα 4.8):

String s1 = new String(args[0]);	Το όνομα της υπηρεσίας
String s2 = new String(args[1]);	Το όνομα του αντικειμένου
String s3 = new String(args[2]);	Το όνομα της υπηρεσίας-προορισμός
String s4 = new String(args[3]);	Το είδος της λειτουργίας
String s5 = new String(args[4]);	Το αντικείμενο ελεγκτής
String ret = (String) call.invoke (new Object[] {s1, s2, s3, s4, s5});	

Σχήμα 4.8 Τα περιεχόμενα της αίτησης μετακίνησης αντικειμένου

Αναλυτικά, κατά την μετακίνηση του αντικειμένου απαιτούνται πολλές ενημερώσεις αρχείων. Από την πλευρά της υπηρεσίας-πηγή θα πρέπει να ενημερωθεί το αρχείο αντικειμένων και το αρχείο που χρησιμοποιεί η μέθοδος-ελεγκτής. Από την πλευρά της υπηρεσίας-προορισμός θα πρέπει να ενημερωθεί ομοίως το αρχείο αντικειμένων και το αρχείο που χρησιμοποιεί η μέθοδος-ελεγκτής. Από την πλευρά της υπηρεσίας Location\_Server θα πρέπει να ενημερωθεί το αρχείο αντιστοίχισης αντικειμένων και υπηρεσιών.

Αρχικά, από την πλευρά της υπηρεσίας-πηγή, αφού βρεθεί το ζητούμενο αντικείμενο, διαβάζονται οι πληροφορίες που περιέχει και αποθηκεύονται σε ένα αλφαριθμητικό, το οποίο αποστέλλεται στην υπηρεσία-προορισμός. Η υπηρεσία-προορισμός όταν λάβει το αλφαριθμητικό το γράφει στο αρχείο αντικειμένων της. Τα αρχεία που χρησιμοποιούνται είναι τύπου τυχαίας προσπέλασης για να επιτρέπουν ταυτόχρονα τόσο την ανάγνωση πληροφορίας, όσο και την εγγραφή πληροφορίας (RandomAccessFile inout = new RandomAccessFile (filename, "rw")).

Για την υλοποίηση της συγκεκριμένης λειτουργίας δημιουργήθηκε επιπλέον, η μέθοδος public String MigrateObject( ), της οποίας ο ρόλος είναι να κάνει τις απαραίτητες ενημερώσεις που αναφέρθηκαν, στα αρχεία της υπηρεσίας-προορισμός.

Η υλοποίηση της λειτουργίας αυτής δε δημιουργεί προβλήματα στη διαχείριση των αιτήσεων. Συγκεκριμένα, αν αποσταλούν αιτήσεις στην υπηρεσία από την οποία μετακινήθηκε το αντικείμενο, αυτές προωθούνται στην υπηρεσία Location\_Server, που έχει ενημερωθεί για τη μετακίνηση, και η διαδικασία συνεχίζεται όπως περιγράφηκε παραπάνω.

Επιπλέον, το αντικείμενο που μετακινείται αφήνει στην αρχική του θέση ένα αντικείμενο που δεν περιέχει πληροφορία και η λειτουργία του αφορά στην προώθηση των κλήσεων που δέχεται στη νέα θέση του αντικειμένου. Η ταυτότητα του αντικειμένου είναι όμοια με του αρχικού αντικειμένου που μετακινήθηκε.

Οι παραπάνω προσθήκες έχουν στόχο την αποφυγή του ενδεχόμενου απώλειας αιτήσεων, που μπορεί να αποσταλούν κατά τη μετακίνηση του αντικειμένου. Ο στόχος επιτυγχάνεται αφού όλες οι αιτήσεις αντιμετωπίζονται με την προώθησή τους στη νέα θέση του αντικειμένου. Η διαδικασία αυτή παύει να χρησιμοποιείται όταν οι πελάτες λάβουν γνώση για τη μετακίνηση και αποστέλλουν τις αιτήσεις στην υπηρεσία που πρέπει.

## ΚΕΦΑΛΑΙΟ 5.ΕΛΕΓΧΟΣ ΚΑΙ ΜΕΤΡΗΣΕΙΣ

---

### 5.1.Πειραματικές Μετρήσεις

#### 5.2.Διοικητική Οργάνωση Βιβλιοθήκης και Κέντρου Πληροφόρησης Πανεπιστημίου Ιωαννίνων

##### 5.2.1. Η Μορφή των Αντικειμένων για την Εφαρμογή

##### 5.2.2. Μετρήσεις για τη Συγκεκριμένη Εφαρμογή

---

### 5.1. Πειραματικές Μετρήσεις

Πριν από την ανάλυση συγκεκριμένων εφαρμογών, πραγματοποιήθηκαν οι μετρήσεις που εμπεριέχονται στους πίνακες που ακολουθούν. Οι μετρήσεις αφορούν σε όλες τις λειτουργίες που μπορούν να εφαρμοστούν στο προτεινόμενο ιεραρχικό μοντέλο πρόσβασης αντικειμένων μέσω υπηρεσιών διαδικτύου.

Ο στόχος των παρακάτω μετρήσεων είναι κυρίως, ο έλεγχος όλων των παρεχομένων λειτουργιών του προτεινόμενου συστήματος.

Πίνακας 5.1 Μετρήσεις χωρίς προώθηση της αίτησης

Είδη αιτήσεων/Μέσος χρόνος(secs)	ακέραιος	πραγματικός	αλφαριθμητικό	πίνακας στοιχείων[]
Ανάγνωση πληροφορίας	1.067	1.068	1.068	1.068
Εγγραφή πληροφορίας	1.068	1.068	1.068	1.069

Πίνακας 5.2 Μετρήσεις με προώθηση της αίτησης σε 1 βήμα

Είδη αιτήσεων/Μέσος χρόνος(secs)	ακέραιος	πραγματικός	αλφαριθμητικό	πίνακας στοιχείων[]
Ανάγνωση πληροφορίας	1.093	1.094	1.093	1.094
Εγγραφή πληροφορίας	1.094	1.094	1.094	1.094

Πίνακας 5.3 Μετρήσεις με προώθηση της αίτησης σε 2 βήματα

Είδη αιτήσεων/Μέσος χρόνος(secs)	ακέραιος	πραγματικός	αλφαριθμητικό	πίνακας στοιχείων[]
Ανάγνωση πληροφορίας	1.124	1.124	1.125	1.124
Εγγραφή πληροφορίας	1.124	1.126	1.124	1.126

Πίνακας 5.4 Μετρήσεις για αντιγραφή και μετακίνηση αντικειμένου

Είδη αιτήσεων	Μέσος χρόνος(secs)
Αντιγραφή αντικειμένου	1.101
Μετακίνηση αντικειμένου	1.105

Πίνακας 5.5 Μετρήσεις με χρήση της υπηρεσίας Location\_Server

Είδη αιτήσεων/Μέσος χρόνος(secs)	ακέραιος	πραγματικός	αλφαριθμητικό	πίνακας στοιχείων[]
Ανάγνωση πληροφορίας	1.099	1.098	1.099	1.099
Εγγραφή πληροφορίας	1.099	1.099	1.098	1.101

Στους παραπάνω πίνακες έγιναν μετρήσεις χρόνου αρχικά χωρίς να χρειάζεται η χρήση του μηχανισμού κληρονομικότητας και στη συνέχεια με τη χρήση του. Συγκεκριμένα, στον Πίνακα 5.1 η πληροφορία βρέθηκε απευθείας στην υπηρεσία που κάλεσε ο πελάτης, στον Πίνακα 5.2 η πληροφορία βρέθηκε ύστερα από μια προώθηση σε αντικείμενο άλλης υπηρεσίας, από όπου κληρονομεί, και στον Πίνακα 5.3 η πληροφορία βρέθηκε ύστερα από δύο προωθήσεις σε αντικείμενα υπηρεσιών που κληρονομεί το ένα από το άλλο. Στον πίνακα 5.4 μετρήθηκε ο χρόνος που χρειάζεται για την αντιγραφή και τη μετακίνηση ενός αντικειμένου σε μια άλλη



υπηρεσία και στον Πίνακα 5.5 ο χρόνος για την εύρεση της πληροφορίας, αν το αντικείμενο δε βρεθεί απευθείας αλλά χρειάζεται για να βρεθεί, η κλήση της υπηρεσίας Location\_Server. Όλες οι παραπάνω μετρήσεις έγιναν σε έναν επεξεργαστή και σε λειτουργικό σύστημα Windows XP Professional.

## **5.2. Διοικητική Οργάνωση Βιβλιοθήκης και Κέντρου Πληροφόρησης Πανεπιστημίου Ιωαννίνων**

Η Διεύθυνση Βιβλιοθήκης και Κέντρου Πληροφόρησης του Πανεπιστημίου Ιωαννίνων συγκροτείται από τα ακόλουθα Τμήματα:

1. Τμήμα Διοίκησης και Οικονομικής Διαχείρισης
2. Τμήμα Εξυπηρέτησης και Συλλογών
3. Τμήμα Προσκτήσεων
4. Τμήμα Βιβλιοθηκονομικής Επεξεργασίας
5. Τμήμα Ηλεκτρονικών Υπηρεσιών

Οι αρμοδιότητες των παραπάνω τμημάτων είναι οι ακόλουθες:

### 1. Τμήμα Διοίκησης και Οικονομικής Διαχείρισης

Το τμήμα είναι υπεύθυνο για την παρακολούθηση της παρουσίας του προσωπικού και τον ορισμό του προσωπικού που θα λειτουργεί τα αναγνωστήρια και τις άλλες υπηρεσίες της Βιβλιοθήκης κατά τις απογευματινές ώρες και τις αργίες. Είναι υπεύθυνο για την οργάνωση σεμιναρίων επιμόρφωσης και συνεχιζόμενης κατάρτισης του προσωπικού της Β.Κ.Π. και συντάσσει το αναγκαίο για τον σκοπό αυτό πληροφοριακό υλικό. Το Τμήμα είναι υπεύθυνο για το σχεδιασμό, την τυποποίηση και την παραγωγή εντύπων κάθε μορφής καθώς και τον σχεδιασμό των εσωτερικών διαδικασιών λειτουργίας.

Το τμήμα είναι αρμόδιο για την παραλαβή, τον χαρακτηρισμό και την πρωτοκόλληση της εισερχόμενης αλληλογραφίας και τη διανομή της στις καθ' ύλην αρμόδιες υπηρεσίες και υπαλλήλους.

Στις αρμοδιότητες του Τμήματος ανήκουν επίσης :

- Η μέριμνα για την αποστολή του υλικού κάθε μορφής (βιβλία, τεύχη περιοδικών, οπτικοακουστικό υλικό κ.α. ) που αποστέλλεται ταχυδρομικώς.
- Η μέριμνα για τη διεκπεραίωση της εξερχόμενης αλληλογραφίας κάθε μορφής.
- Η παρακολούθηση των πληρωμών προς τις ταχυδρομικές υπηρεσίες κάθε μορφής.
- Η μέριμνα για την τήρηση αρχείων εγγράφων, εγκυκλίων κ.λπ.
- Η μέριμνα για την επιμελητεία, την καθαριότητα και την ασφάλεια των κτηρίων της Β.Κ.Π.-Π.Ι.

## 2. Τμήμα Εξυπηρέτησης και Συλλογών

Το Τμήμα καθιστά εφικτή και διασφαλίζει την πρόσβαση και τη χρήση των συλλογών της Β.Κ.Π.-Π.Ι. Λειτουργεί τις υπηρεσίες εξυπηρέτησης των αναγνωστών της Βιβλιοθήκης τόσο στις εγκαταστάσεις της όσο και εκτός αυτής.

Είναι υπεύθυνο για την παροχή υπηρεσιών πληροφόρησης και τεκμηρίωσης της Β.Κ.Π. Εκτελεί κάθε εργασία που σχετίζεται με την αποτελεσματική χρήση των συλλογών της από τους αναγνώστες και μεριμνά, μέσω συνεργασιών με άλλες Βιβλιοθήκες στην Ελλάδα και το εξωτερικό, για την ικανοποίηση των αναγκών τους στην περίπτωση που οι συλλογές της Β.Κ.Π.-Π.Ι. αδυνατούν να τις ικανοποιήσουν.

Παρέχει βιβλιογραφική και κάθε άλλης μορφής πληροφόρηση και εκτελεί εργασίες διαδανεισμού μέσω δικτύου Βιβλιοθηκών σύμφωνα με τα προβλεπόμενα στον εσωτερικό κανονισμό λειτουργίας της Β.Κ.Π.-Π.Ι. Συντάσσει πληροφοριακό υλικό και έχει την ευθύνη για την πληροφόρηση των επισκεπτών της.

Το τμήμα είναι υπεύθυνο για τη λειτουργία των αναγνωστηρίων της Β.Κ.Π.-Π.Ι. τον δανεισμό υλικού, την ευθύνη λειτουργίας του σχετικού εξοπλισμού χρήσης των συλλογών, όπως φωτοτυπικά μηχανήματα, αναγνώστες μικροταινιών, μικροϋπολογιστές κ.α.

### 3. Τμήμα Προσκτήσεων

Το Τμήμα είναι υπεύθυνο για την παραλαβή του βιβλιακού υλικού και άλλου υλικού που εισέρχεται στη Β.Κ.Π.-Π.Ι. με οποιονδήποτε τρόπο. Παρακολουθεί τα βιβλιογραφικά δελτία των εκδοτών, τις βιβλιοπαρουσιάσεις κάθε μορφής και τις σχετικές πηγές του διαδικτύου για την παρακολούθηση της εθνικής και διεθνούς εκδοτικής παραγωγής και κάνει κάθε ενέργεια για τον εμπλουτισμό των συλλογών. Εκτελεί όλες τις διαδικασίες παραγγελίας, παραλαβής, καταγραφής στο κτηματολόγιο και προώθησης για καταλογογράφηση του υλικού κάθε μορφής που εισέρχεται στη Β.Κ.Π.-Π.Ι. Παραλαμβάνει το υλικό που έχει παραγγελθεί και ελέγχει αν είναι όντως αυτό που είχε παραγγελθεί. Αντιπαραβάλλει τα τιμολόγια των προμηθευτών με τις τιμές των καταλόγων των εκδοτικών οίκων για έλεγχο της πραγματικής αξίας του παραγγελθέντος υλικού και προωθεί την πληρωμή των σχετικών τιμολογίων

### 4. Τμήμα Βιβλιοθηκονομικής Επεξεργασίας

Το Τμήμα είναι υπεύθυνο για τη βιβλιοθηκονομική επεξεργασία του υλικού κάθε μορφής –εκτός των περιοδικών εκδόσεων- και την καταγραφή του στο μηχανογραφικό σύστημα της Β.Κ.Π.-Π.Ι. Είναι υπεύθυνο για την προετοιμασία του βιβλιακού και άλλου υλικού για ένταξή του σε συγκεκριμένες συλλογές.

Είναι υπεύθυνο για την ανταλλαγή βιβλιογραφικών εγγραφών της Β.Κ.Π.-Π.Ι. με άλλες Βιβλιοθήκες της Ελλάδας και του εξωτερικού και την παραγωγή σχετικών υποστηρικτικών εργαλείων, όπως αρχείων καθιερωμένων όρων, θησαυρών όρων κ.α.

Συνεργάζεται με άλλες Βιβλιοθήκες για τη δημιουργία συλλογικών καταλόγων Βιβλιοθηκών

### 5. Τμήμα Ηλεκτρονικών υπηρεσιών

Έχει την ευθύνη της μηχανοργάνωσης της Β.Κ.Π.-Π.Ι. Ως εκ τούτου, εγκαθιστά και λειτουργεί τα μηχανογραφικά συστήματα που εξασφαλίζουν πρόσβαση στους καταλόγους της Β.Κ.Π.-Π.Ι. Είναι αρμόδιο για τη λειτουργία συστημάτων αποθήκευσης και χρήσης των ψηφιοποιημένων συλλογών κάθε μορφής, καθώς και των συστημάτων μηχανοργάνωσης των λειτουργιών της Β.Κ.Π.-Π.Ι. Είναι υπεύθυνο για τακτική λήψη αντιγράφων ασφαλείας των βάσεων δεδομένων που δημιουργούνται στη Β.Κ.Π.-Π.Ι.

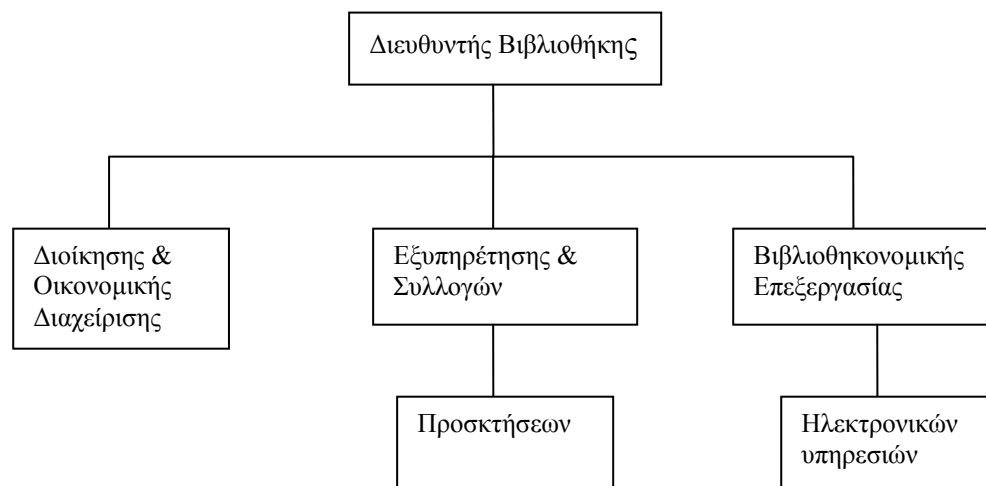
Τέλος, για τα τμήματα που συγκροτούν την Β.Κ.Π.-Π.Ι είναι υπεύθυνος ο Διευθυντής της Βιβλιοθήκης.

Πιο αναλυτική περιγραφή των τμημάτων και των αρμοδιοτήτων τους βρίσκεται στο παράρτημα.

### 5.2.1 Η Μορφή των Αντικειμένων για την Εφαρμογή

Στην ανάλυση που προηγήθηκε, κάθε τμήμα που περιγράφηκε υλοποιείται ως μια υπηρεσία διαδικτύου και οι αρμοδιότητες του κάθε τμήματος αποτελούν τις αντίστοιχες μεθόδους-αντικείμενα της κλάσης.

Η ιεραρχία των αντικειμένων που είναι χαρακτηριστικό του συστήματός μας βρίσκεται άμεση εφαρμογή στο οργανόγραμμα της Β.Κ.Π.-Π.Ι. που φαίνεται στο Σχήμα 5.1. Συγκεκριμένα, αναπαρίστανται μόνο οι υπηρεσίες διαδικτύου και όχι οι αρμοδιότητες κάθε τμήματος, που περιγράφηκαν στην προηγούμενη παράγραφο.



Σχήμα 5.1 Το οργανόγραμμα της Β.Κ.Π.-Π.Ι.

### 5.2.2 Μετρήσεις για τη Συγκεκριμένη Εφαρμογή

Πραγματοποιήθηκαν μετρήσεις που αφορούν στο χρόνο που μεσολαβεί από τη στιγμή που ο πελάτης απευθύνει μια κλήση προς κάποια υπηρεσία και έως ότου λάβει την απάντηση από την υπηρεσία που ικανοποίησε την αίτησή του.

Για την εφαρμογή δημιουργήθηκαν τόσες υπηρεσίες διαδικτύου, όσες και τα τμήματα της Β.Κ.Π.-Π.Ι. που αναλύθηκαν στην προηγούμενη παράγραφο και όμοια αρχεία αντικειμένων. Κάθε αρχείο που αντιστοιχεί στην ανάλογη υπηρεσία διαδικτύου περιλαμβάνει τόσα αντικείμενα, όσες και λειτουργίες που μπορεί να εκτελέσει κάθε τμήμα.

Οι μετρήσεις που εμπεριέχονται στον Πίνακα 5.6 και στον Πίνακα 5.7, έγιναν σε τοπικό δίκτυο με τη χρήση τόσων επεξεργαστών όσες και οι υπηρεσίες που λαμβάνουν μέρος στη διαδικασία και περιλαμβάνουν και αναζήτηση της πληροφορίας σε άλλη υπηρεσία. Οι μετρήσεις έγιναν σε υπολογιστές που διαθέτουν λειτουργικό σύστημα Windows XP Professional.

Πίνακας 5.6 Ανάγνωση- εγγραφή απομακρυσμένης εφαρμογής

Αίτηση/χρόνος(secs)	Απευθείας εύρεση	Με 1 προώθηση	Με 2 προωθήσεις
Ανάγνωση πληροφορίας	1.072	1.515	3.626
Εγγραφή πληροφορίας	1.072	1.515	3.625

Πίνακας 5.7 Αντιγραφή-μετακίνηση απομακρυσμένης εφαρμογής

Είδη αιτήσεων	Μέσος χρόνος(secs)
Αντιγραφή αντικειμένου	1.454
Μετακίνηση αντικειμένου	1.460

## ΚΕΦΑΛΑΙΟ 6. ΑΞΙΟΛΟΓΗΣΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

---

### 6.1 Αξιολόγηση του Συστήματος

---

#### 6.1 Αξιολόγηση του Συστήματος

Προτείναμε και υλοποιήσαμε ένα σύστημα για απομακρυσμένη πρόσβαση αντικειμένων, που βασίζεται στις υπηρεσίες διαδικτύου σε πρώτο επίπεδο και χρησιμοποιεί την Axis σαν την πλατφόρμα ενδιάμεσου λογισμικού.

Χαρακτηριστικό στο προτεινόμενο σύστημα είναι η ιεραρχική οργάνωση των αντικειμένων. Ο στόχος του είναι να παρέχει έναν εναλλακτικό σχεδιασμό για σχεδιασμούς οργάνωσης επικοινωνίας μεταξύ υπηρεσιών διαδικτύου. Είναι περισσότερο ενδιαφέρον να επικεντρωθεί η διαδικασία αξιολόγησης στα πλεονεκτήματα της χρήσης μιας ιεραρχικής οργάνωσης των κοινών αντικειμένων για την ανάπτυξη της εφαρμογής. Έχουν δοθεί παραδείγματα τέτοιων εφαρμογών και έχουν περιγραφεί τα πλεονεκτήματα μιας τέτοιας προσέγγισης.

Ωστόσο, η απόδοση είναι ένας κρίσιμος παράγοντας στην ανάπτυξη λογισμικού. Η πλατφόρμα του ενδιάμεσου λογισμικού εισάγει πάντα επιπρόσθετο χρόνο για την εφαρμογή των υποστηριζόμενων λειτουργιών (εγκαταστάσεων). Οι εφαρμογές που αναπτύσσονται σε αντικειμενοστραφές περιβάλλον είναι πάντα πιο αργές από αυτές που ακολουθούν πιο παραδοσιακούς συναρτησιακούς σχεδιασμούς. Όμως, ο επιπρόσθετος χρόνος που εισάγει το σύστημα είναι ασήμαντος εξαιτίας της απλότητας στη φιλοσοφία του και στην εφαρμογή του.

Συμπερασματικά, σε αυτή την εργασία παρουσιάστηκε το ιεραρχικό μοντέλο πρόσβασης αντικειμένων με τη χρήση υπηρεσιών διαδικτύου. Ακολουθεί την τεχνική των κλάσεων του αντικειμενοστραφούς προγραμματισμού, δημιουργώντας όμως μια ιεραρχία αντικειμένων για να εκφράσει τις σχέσεις κληρονομικότητας. Υποστηρίζει μετακίνηση και δημιουργία αντιγράφων αντικειμένων, ενώ παράλληλα διατηρεί την ιδέα του αντικειμενοστραφούς προγραμματισμού.

Παρόλο που το σύστημα υλοποιεί πολλές χαρακτηριστικές λειτουργίες, υπάρχουν ακόμα και άλλες που μπορεί να αποτελέσουν μελλοντικές προσθήκες, όπως οι παρακάτω:

- Παράμετροι και μέτρηση της απόδοσης για μεγαλύτερα δίκτυα.
- Μέθοδοι κωδικοποίησης και αποκωδικοποίησης για κατανεμημένη αποθήκευση σε ανοιχτά περιβάλλοντα.
- Ύπαρξη πολλαπλής κληρονομικότητας.

## ΑΝΑΦΟΡΕΣ

---

- [1] Wilkinson, B., Allen, M.: Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers. Prentice Hall (1999).
- [2] Foster, I., Kesselman, C., Tuecke, S.: The Anatomy of the Grid: Enabling Scalable Virtual Organizations. International Journal of High Performance Computing Applications **15** (2001) 200–222 <http://www.globus.org/research/papers/anatomy.pdf>.
- [3] Noack, J., Mehmaneche, H., Mehmaneche, H., Zandler, A.: Architectural Patterns for Web Applications. In Hamza, M., ed.: 18th IASTED International Conference on Applied Informatics (AI 2000), Proceedings, Innsbruck, Austria, ACTA Press (2000) [citeseer.nj.nec.com/260788.html](http://citeseer.nj.nec.com/260788.html).
- [4] Streicher, M.: Creating Web Services with AXIS: Apache's Latest SOAP Implementation Bootstraps Web Services. Linux Magazine (2002) [http://www.linux-mag.com/2002-08/axis\\_01.html](http://www.linux-mag.com/2002-08/axis_01.html).
- [5] Buyya, R., ed.: High Performance Cluster Computing, Programming and Applications. Prentice Hall (1999).
- [6] Message Passing Interface Forum: MPI-2: Extensions to the Message-Passing Interface Standard. (1997).
- [7] Jones, M., Berkley, C., Bojilova, J., Schildhauer, M.: Managing Scientific Metadata. IEEE Internet Computing **5** (2001) 59–68.



- [8] Rhea, S., Wells, C., Eaton, P., Geels, D., Zhao, B., Weatherspoon, H., Kubiawicz, J.: Maintenance-Free Global Data Storage. *IEEE Internet Computing* **5** (2001) 40–49.
- [9] EU-DataGrid (EDG): The DataGrid Project (2003) <http://eu-datagrid.web.cern.ch/eu-datagrid>.
- [10] Globus: Fundamental Technologies Needed to Build Computational Grids (2003) <http://www.globus.org>.
- [11] GridLab: A Grid Application Toolkit and Testbed (2003) <http://www.gridlab.org>.
- [12] Coulouris, G., Dollimore, J., Kindberg, T.: *Distributed Systems – Concepts and Design*. Addison-Wesley (2001).
- [13] Fagg, G.E., Gabriel, E., Resch, M., Dongarra, J.J.: Parallel IO Support for Meta-computing Applications: MPI Connect IO Applied to PACX-MPI. In: *Recent Advances in Parallel Virtual Machine and Message Passing Interface*. Number 2131 in *Lecture Notes in Computer Science*, Springer-Verlag (2001) 135–147 8th European PVM/MPI Users' Group Meeting, Santorini/Thera, Greece, September 23-26, 2001, Proceedings.
- [14] Tsujita, Y., Imamura, T., Takemiya, H., Yamagishi, N.: Stampi-I/O: A Flexible Parallel-I/O Library for Heterogeneous Computing Environment. In: *Recent Advances in Parallel Virtual Machine and Message Passing Interface*. Number 2474 in *Lecture Notes in Computer Science*, Springer-Verlag (2002) 288–295 9th European PVM/MPI Users' Group Meeting, Linz, Austria, September/October, 2002, Proceedings.
- [15] Sunderam, V., Moyer, S.: PIOUS for PVM (1995) <http://www.mathcs.emory.edu/pious>.

- [16] Vahdat, A.M., Eastham, P.C., Anderson, T.E.: WebFS: A Global Cache Coherent File System. Technical report, Computer Science Division, University of California Berkeley (1996) <http://www.cs.duke.edu/~vahdat/webfs/webfs.html>.
- [17] Alexandrov, A.D., Ibel, M., Schauser, K.E., Scheiman, C.J.: Extending the Operating System at the User Level: the Ufo Global File System. In: Proceedings of the USENIX Annual Technical Conference, Anaheim, California, USA (1997) 77–90.
- [18] Bester, J., Foster, I., Kesselman, C., Tedesco, J., Tuecke, S.: GASS: A Data Movement and Access Service for Wide Area Computing Systems. In: Proceedings of the Sixth Workshop on Input/Output in Parallel and Distributed Systems, Atlanta, GA, ACM Press (1999) 78–88.
- [19] McClanahan, C.R.: Tomcat: Application Developer's Guide. (2002) Apache Jakarta Project, <http://jakarta.apache.org/tomcat/tomcat-4.1-doc/appdev/index.html>.
- [20] Czarnul, P.: Programming, Tuning and Automatic Parallelization of Irregular Divide-and-Conquer Applications in DAMPVM/DAC. *International Journal of High Performance Computing Applications* **17** (2003) 77–93.
- [21] Czarnul, P., Tomko, K., Krawczyk, H.: Dynamic Partitioning of the Divide-and-Conquer Scheme with Migration in PVM Environment. In: Recent Advances in Parallel Virtual Machine and Message Passing Interface. Number 2131 in Lecture Notes in Computer Science, Springer-Verlag (2001) 174–182 8th European PVM/MPI Users' Group Meeting, Santorini/Thera, Greece, September 23-26, 2001, Proceedings.
- [22] Czarnul, P., Krawczyk, H.: Dynamic Assignment with Process Migration in Distributed Environments. In: Recent Advances in Parallel Virtual Machine and Message Passing Interface. Number 1697 in Lecture Notes in Computer Science (1999) 509–516.
- [23] Web Services Flow Language (WSFL 1.0)

- [24] Web Services Conceptual Architecture (WSCA 1.0)
- [25] IBM Web Services Roadmap
- [26] Web Services and Business Process Management Technology
- [27] G.Manis: Introducing Inheritance of Data-objects in Distributed Shared Memory (submitted).
- [28] G.Manis: Data-Object Oriented Design for Distributed Shared Memory. Proc. High Performance Computing and Networking pp.583-586.
- [29] G.Manis: Persistent and non-Persistent Objects on top of PVM and MPI. Proc Euro PVM-MPI, pp.91-97.
- [30] AXIS <http://xml.apache.org/axis>
- [31] DISCO <http://msdn.microsoft.com/xml/general/disco.asp>
- [32] EbXML <http://www.ebxml.org/>
- [33] OAG <http://www.openapplications.org/>
- [34] SOAP <http://www.w3.org/TR/SOAP/>
- [35] UDDI <http://www.uddi.org/>
- [36] XMLP <http://www.w3.org/2000/xp/>
- [37] WSDL <http://www.uddi.org/submissions.html>
- [38] XML Schema Part 1 <http://www.w3.org/TR/xmlschema-1/>
- [39] XML Schema Part 2 <http://www.w3.org/TR/xmlschema-2/>

[40] JAVA Alamnac <http://www.exampledepot.com/index.html>

[41] JAVA <http://java.sun.com/>

[42] Bruce Eckel, Thinking in Java: The Definitive Introduction to Object-Oriented Programming in the Language of the World-Wide Web, 3<sup>rd</sup> edition

[43] Roosdiana Wonohoesodo Zahir Tari: A Role based Access Control for Web Services, RMIT University, School of Computer Science and Information Technology.

[44] Pawel Czarnul: Architecture and Implementation of Distributed Data Storage Using Web Services, CORBA and PVM\_Faculty of Electronics, Telecommunications and Informatics Gdansk University of Technology, Poland.

[45] Gunhee Lee<sup>1</sup>, Hongjin Yeh<sup>1</sup>, Wonil Kim<sup>2\*</sup>, and Dong-Kyoo Kim<sup>1</sup>: Web Security Using Distributed Role Hierarchy.

## **ΠΑΡΑΡΤΗΜΑ**

---

### **Διοικητική Οργάνωση Βιβλιοθήκης και Κέντρου Πληροφόρησης Πανεπιστημίου Ιωαννίνων**

Η Διεύθυνση Βιβλιοθήκης και Κέντρου Πληροφόρησης του Πανεπιστημίου Ιωαννίνων συγκροτείται από τα ακόλουθα Τμήματα:

1. Τμήμα Διοίκησης και Οικονομικής Διαχείρισης
2. Τμήμα Εξυπηρέτησης και Συλλογών
3. Τμήμα Προσκτήσεων
4. Τμήμα Βιβλιοθηκονομικής Επεξεργασίας
5. Τμήμα Ηλεκτρονικών Υπηρεσιών

Οι αρμοδιότητες των παραπάνω τμημάτων είναι οι ακόλουθες:

1. Τμήμα Διοίκησης και Οικονομικής Διαχείρισης

Το τμήμα είναι υπεύθυνο για την παρακολούθηση της παρουσίας του προσωπικού και τον ορισμό του προσωπικού που θα λειτουργεί τα αναγνωστήρια και τις άλλες υπηρεσίες της Βιβλιοθήκης κατά τις απογευματινές ώρες και τις αργίες. Είναι υπεύθυνο για την οργάνωση σεμιναρίων επιμόρφωσης και συνεχιζόμενης κατάρτισης του προσωπικού της Β.Κ.Π. και συντάσσει το αναγκαίο για τον σκοπό αυτό πληροφοριακό υλικό. Το Τμήμα είναι υπεύθυνο για το σχεδιασμό, την τυποποίηση και την παραγωγή εντύπων κάθε μορφής καθώς και τον σχεδιασμό των εσωτερικών διαδικασιών λειτουργίας.

Το τμήμα είναι αρμόδιο για την παραλαβή, τον χαρακτηρισμό και την πρωτοκόλληση της εισερχόμενης αλληλογραφίας και τη διανομή της στις καθ' ύλην αρμόδιες υπηρεσίες και υπαλλήλους.

Στις αρμοδιότητες του Τμήματος ανήκουν επίσης :

- Η μέριμνα για την αποστολή του υλικού κάθε μορφής (βιβλία, τεύχη περιοδικών, οπτικοακουστικό υλικό κ.α. ) που αποστέλλεται ταχυδρομικώς.
- Η μέριμνα για τη διεκπεραίωση της εξερχόμενης αλληλογραφίας κάθε μορφής.
- Η παρακολούθηση των πληρωμών προς τις ταχυδρομικές υπηρεσίες κάθε μορφής.
- Η μέριμνα για την τήρηση αρχείων εγγράφων, εγκυκλίων κ.λπ.
- Η μέριμνα για την επιμελητεία, την καθαριότητα και την ασφάλεια των κτηρίων της Β.Κ.Π.-Π.Ι.

Είναι αρμόδιο για τη διοικητική και γραμματειακή υποστήριξη της Συγκλητικής επιτροπής Βιβλιοθήκης (Σ.Ε.Β.) και κάθε άλλης σχετικής επιτροπής. Μεριμνά για την τήρηση των πρακτικών και την κοινοποίηση των αποφάσεων στις καθ' ύλην αρμόδιες υπηρεσίες . Τηρεί αρχείο πρακτικών και εκδίδει αντίγραφα ή αποσπάσματα αυτών. Ασχολείται με κάθε άλλο θέμα που αφορά τη λειτουργία του συλλογικού οργάνου.

Μεριμνά για την προβολή της εκπαιδευτικής, επιστημονικής, ερευνητικής και πολιτιστικής δραστηριότητας της Β.Κ.Π.- Π.Ι. με την έκδοση εντύπων, δελτίων τύπου και ανακοινώσεων, την παραγωγή ραδιοφωνικών και τηλεοπτικών εκπομπών, κ.α. Μεριμνά επίσης, για την οργάνωση και τη διεξαγωγή εκθέσεων, τελετών, εορτών, συνεδρίων κ.α., καθώς και για την εξασφάλιση χώρων για διαλέξεις, συγκεντρώσεις και λοιπές εκδηλώσεις. Το Τμήμα είναι υπεύθυνο για την παραγωγή ενημερωτικών φυλλαδίων και οδηγιών που αφορούν τις υπηρεσίες της Βιβλιοθήκης

Επίσης, είναι υπεύθυνο για κάθε άλλη εκδοτική δραστηριότητα της Β.Κ.Π.-Π.Ι., την αναπαραγωγή υλικού από τις συλλογές της, τις σχέσεις της Β.Κ.Π.-Π.Ι. με τους εκδότες, προμηθευτές, κ.α. Είναι υπεύθυνο για την εκτέλεση των εργασιών που αφορούν τη συμμετοχή της Β.Κ.Π.-Π.Ι. σε Κοινοπραξίες και Συνδέσμους Βιβλιοθηκών, τη διαχείριση Αναπτυξιακών και Ερευνητικών Έργων, κ.α.

Μεριμνά για τη διενέργεια κάθε είδους προμήθειας, όπως προμήθεια εξοπλισμού και αναλώσιμων υλικών. Τηρεί κτηματολόγιο του προμηθευόμενου εξοπλισμού και κάθε είδους υλικού. Εκτελεί τις απαραίτητες εργασίες για την πληρωμή προμηθευτών,

βιβλιοπωλείων, κ.α. Συγκεντρώνει και αποδίδει στις καθ' ύλην αρμόδιες υπηρεσίες του Πανεπιστημίου τα έσοδα από δραστηριότητες της Β.Κ.Π.-Π.Ι., όπως η χρήση των φωτοτυπικών μηχανημάτων, τα επιβαλλόμενα πρόστιμα, οι εκδόσεις της Β.Κ.Π.-Π.Ι. κ.α.

Στις αρμοδιότητές του ανήκει η εισήγηση για την εκποίηση ή την καταστροφή ακατάλληλου ή άχρηστου υλικού και η διαγραφή του από τα διαχειριστικά βιβλία, καθώς και η τήρηση όλων των αναγκαίων βιβλίων και στοιχείων. Επίσης, η εισήγηση στα αρμόδια όργανα της Β.Κ.Π.-Π.Ι., για κάθε θέμα που ανάγεται στην αποδοχή ή μη των κληρονομιών, κληροδοσιών ή δωρεών. Η μέριμνα για τη διασφάλιση της κινητής και ακίνητης περιουσίας της Β.Κ.Π.-Π.Ι. και η τήρηση των αναγκαίων βιβλίων και στοιχείων.

## 2. Τμήμα Εξυπηρέτησης και Συλλογών

Το Τμήμα καθιστά εφικτή και διασφαλίζει την πρόσβαση και τη χρήση των συλλογών της Β.Κ.Π.-Π.Ι. Λειτουργεί τις υπηρεσίες εξυπηρέτησης των αναγνωστών της Βιβλιοθήκης τόσο στις εγκαταστάσεις της όσο και εκτός αυτής.

Είναι υπεύθυνο για την παροχή υπηρεσιών πληροφόρησης και τεκμηρίωσης της Β.Κ.Π. Εκτελεί κάθε εργασία που σχετίζεται με την αποτελεσματική χρήση των συλλογών της από τους αναγνώστες και μεριμνά, μέσω συνεργασιών με άλλες Βιβλιοθήκες στην Ελλάδα και το εξωτερικό, για την ικανοποίηση των αναγκών τους στην περίπτωση που οι συλλογές της Β.Κ.Π.-Π.Ι. αδυνατούν να τις ικανοποιήσουν.

Παρέχει βιβλιογραφική και κάθε άλλης μορφής πληροφόρηση και εκτελεί εργασίες διαδανεισμού μέσω δικτύου Βιβλιοθηκών σύμφωνα με τα προβλεπόμενα στον εσωτερικό κανονισμό λειτουργίας της Β.Κ.Π.-Π.Ι. Συντάσσει πληροφοριακό υλικό και έχει την ευθύνη για την πληροφόρηση των επισκεπτών της.

Το τμήμα είναι υπεύθυνο για τη λειτουργία των αναγνωστηρίων της Β.Κ.Π.-Π.Ι. τον δανεισμό υλικού, την ευθύνη λειτουργίας του σχετικού εξοπλισμού χρήσης των συλλογών, όπως φωτοτυπικά μηχανήματα, αναγνώστες μικροταινιών, μικροϋπολογιστές κ.α.

Είναι υπεύθυνο για την ασφάλεια, επανατοποθέτηση και ταξιθέτηση των συλλογών της Β.Κ.Π.-Π.Ι. σύμφωνα με τα προβλεπόμενα στον εσωτερικό κανονισμό λειτουργίας. Επιπλέον για την επιλογή του υλικού για συντήρηση, βιβλιοδεσία και αναβιβλιοδέτηση. Επίσης, είναι υπεύθυνο για την οργάνωση και τη χρήση των συλλογών της Β.Κ.Π.-Π.Ι. οι οποίες στεγάζονται σε χώρους εκτός των κεντρικών βιβλιοστασιών της.

Είναι αρμόδιο για την έκδοση των ταυτοτήτων αναγνωστών με τη λειτουργία του σχετικού εξοπλισμού καταγραφής των χρηστών και παραγωγής ταυτοτήτων, για την τήρηση αρχείων δανειζομένων και τη χορήγηση βεβαιώσεων για τις υποχρεώσεις των αναγνωστών προς τη Βιβλιοθήκη. Επιπλέον, είναι υπεύθυνο για την είσπραξη προστίμων που επιβάλλονται στους αναγνώστες για καθυστερήσεις στην επιστροφή βιβλίων, για απώλεια δανεισμένων βιβλίων κ.α.

### 3. Τμήμα Προσκτήσεων

Το Τμήμα είναι υπεύθυνο για την παραλαβή του βιβλιακού υλικού και άλλου υλικού που εισέρχεται στη Β.Κ.Π.-Π.Ι. με οποιονδήποτε τρόπο. Παρακολουθεί τα βιβλιογραφικά δελτία των εκδοτών, τις βιβλιοπαρουσιάσεις κάθε μορφής και τις σχετικές πηγές του διαδικτύου για την παρακολούθηση της εθνικής και διεθνούς εκδοτικής παραγωγής και κάνει κάθε ενέργεια για τον εμπλουτισμό των συλλογών. Εκτελεί όλες τις διαδικασίες παραγγελίας, παραλαβής, καταγραφής στο κτηματολόγιο και προώθησης για καταλογογράφηση του υλικού κάθε μορφής που εισέρχεται στη Β.Κ.Π.-Π.Ι. Παραλαμβάνει το υλικό που έχει παραγγελθεί και ελέγχει αν είναι όντως αυτό που είχε παραγγελθεί. Αντιπαραβάλλει τα τιμολόγια των προμηθευτών με τις τιμές των καταλόγων των εκδοτικών οίκων για έλεγχο της πραγματικής αξίας του παραγγελθέντος υλικού και προωθεί την πληρωμή των σχετικών τιμολογίων

Το τμήμα είναι αρμόδιο για την παρακολούθηση της ομαλής ποής των συνδρομών περιοδικών και των εφημερίδων οι οποίες εισέρχονται στη Β.Κ.Π.-Π.Ι. και ενημερώνει τους εκδότες ή προμηθευτές για τυχόν ελλείψεις τευχών περιοδικών, καθυστερήσεις παραλαβής τευχών, κ.λπ.

Κάνει εισαγωγή στοιχείων στο μηχανογραφικό σύστημα περιοδικών της Β.Κ.Π.-Π.Ι. Μερικώς για την έκδοση σχετικών καταλόγων κάθε μορφής, τη διάθεση



βιβλιογραφικών εγγραφών περιοδικών εκδόσεων σε άλλες Βιβλιοθήκες, την παροχή στατιστικών στοιχείων για τις περιοδικές εκδόσεις και τις εφημερίδες της Β.Κ.Π.-Π.Ι. Είναι υπεύθυνο για τη διάθεση για χρήση από τους ερευνητές της Β.Κ.Π.-Π.Ι. των ηλεκτρονικών περιοδικών της Ελλάδας και του εξωτερικού και μεριμνά για τη διαχείριση των σχετικών συμβολαίων χρήσης και αποθήκευσης.

Το Τμήμα είναι υπεύθυνο για την πρόσκτηση οπτικοακουστικού και ψηφιακού υλικού της Β.Κ.Π.-Π.Ι., υλικού του διαδικτύου , κ.α.

#### 4. Τμήμα Βιβλιοθηκονομικής Επεξεργασίας

Το Τμήμα είναι υπεύθυνο για τη βιβλιοθηκονομική επεξεργασία του υλικού κάθε μορφής –εκτός των περιοδικών εκδόσεων- και την καταγραφή του στο μηχανογραφικό σύστημα της Β.Κ.Π.-Π.Ι. Είναι υπεύθυνο για την προετοιμασία του βιβλιακού και άλλου υλικού για ένταξή του σε συγκεκριμένες συλλογές.

Είναι υπεύθυνο για την ανταλλαγή βιβλιογραφικών εγγραφών της Β.Κ.Π.-Π.Ι. με άλλες Βιβλιοθήκες της Ελλάδας και του εξωτερικού και την παραγωγή σχετικών υποστηρικτικών εργαλείων, όπως αρχείων καθιερωμένων όρων, θησαυρών όρων κ.α.

Συνεργάζεται με άλλες Βιβλιοθήκες για τη δημιουργία συλλογικών καταλόγων Βιβλιοθηκών

#### 5. Τμήμα Ηλεκτρονικών υπηρεσιών

Έχει την ευθύνη της μηχανοργάνωσης της Β.Κ.Π.-Π.Ι. Ως εκ τούτου, εγκαθιστά και λειτουργεί τα μηχανογραφικά συστήματα που εξασφαλίζουν πρόσβαση στους καταλόγους της Β.Κ.Π.-Π.Ι. Είναι αρμόδιο για τη λειτουργία συστημάτων αποθήκευσης και χρήσης των ψηφιοποιημένων συλλογών κάθε μορφής, καθώς και των συστημάτων μηχανοργάνωσης των λειτουργιών της Β.Κ.Π.-Π.Ι. Είναι υπεύθυνο για τακτική λήψη αντιγράφων ασφαλείας των βάσεων δεδομένων που δημιουργούνται στη Β.Κ.Π.-Π.Ι.

Το Τμήμα είναι υπεύθυνο για την εκτέλεση των εργασιών που σχετίζονται με την παραγωγή, τη διάθεση και τη διαχείριση ψηφιοποιημένων τεκμηρίων κάθε μορφής που παράγονται από τις συλλογές της Β.Κ.Π.-Π.Ι. ή από συλλογές εκτός αυτής.

Είναι υπεύθυνο για τη σύνδεση και την καλή λειτουργία του εσωτερικού δικτύου Η/Υ της Β.Κ.Π. και την ορθολογική εκμετάλλευση αυτού. Μεριμνά για την αποτελεσματική σύνδεση της Β.Κ.Π.-Π.Ι. με διεθνή ή εθνικά δίκτυα πληροφόρησης. Είναι υπεύθυνο για την ασφάλεια του δικτύου δεδομένων της Β.Κ.Π.-Π.Ι. και εισηγείται σχετική πολιτική πρόσβασης σε διαθέσιμες πηγές πληροφόρησης του διαδικτύου. Μεριμνά για την εγκατάσταση και τη συντήρηση των πάσης φύσεως μηχανημάτων και λογισμικών Η/Υ που χρησιμοποιούνται στη Β.Κ.Π.-Π.Ι.

Είναι υπεύθυνο για τη σύνταξη προδιαγραφών για την προμήθεια νέου εξοπλισμού κάθε μορφής και ιδιαίτερα εξοπλισμού πληροφορικής. Παρέχει συμβουλευτικές υπηρεσίες για τεχνικά θέματα και εκπαιδεύει το προσωπικό της Β.Κ.Π.-Π.Ι. και τους χρήστες της στη χρήση ηλεκτρονικών μέσων και του λογισμικού που διαθέτει η Β.Κ.Π.-Π.Ι. Έχει την ευθύνη για την τήρηση αρχείου με τα λογισμικά πακέτα που αγοράζονται, τα εγχειρίδιά τους και τα συμβόλαια χρήσης τους.

## **ΣΥΝΤΟΜΟ ΒΙΟΓΡΑΦΙΚΟ**

Η Αναστασία Κώστα γεννήθηκε το 1981 στην Άρτα. Το 2001 ξεκίνησε τις σπουδές της στο Τμήμα Πληροφορικής του Πανεπιστημίου Ιωαννίνων, τις οποίες και ολοκλήρωσε το 2005. Το διάστημα 2005 – 2008 παρακολούθησε το μεταπτυχιακό πρόγραμμα σπουδών του ίδιου τμήματος.