

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

Μεταπτυχιακή Εργασία

Κατανεμημένο Σύστημα Αποθήκευσης Ροών Εργασίας Επιστημονικών Αντικειμένων

Κυριάκος Ε. Κρητικός

Ηράκλειο, Νοέμβριος 2002

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

Κατανεμημένο Σύστημα Αποθήκευσης Ροών Εργασίας Επιστημονικών Αντικειμένων

Εργασία που υποβλήθηκε από τον

ΚΥΡΙΑΚΟ Ε. ΚΡΗΤΙΚΟ

ως μερική εκπλήρωση των απαιτήσεων
για την απόκτηση

ΜΕΤΑΠΤΥΧΙΑΚΟΥ ΔΙΠΛΩΜΑΤΟΣ ΕΙΔΙΚΕΥΣΗΣ

Συγγραφέας:

Κυριάκος Ε. Κρητικός
Τμήμα Επιστήμης Υπολογιστών
Πανεπιστήμιο Κρήτης

Εξεταστική Επιτροπή:

Αικατερίνη Χούστη, Καθηγήτρια
Επόπτης

Δημήτρης Πλεξουσάκης, Αναπληρωτής Καθηγητής
Μέλος

Βασίλης Χριστοφίδης, Επίκουρος Καθηγητής
Μέλος

Δεκτή:

Πάνος Κωνσταντόπουλος, Καθηγητής
Πρόεδρος Επιτροπής Μεταπτυχιακών Σπουδών

Νοέμβριος 2002

Κατανεμημένο Σύστημα Αποθήκευσης Ροών Εργασίας Επιστημονικών Αντικειμένων

Κυριάκος Ε. Κρητικός

Μεταπτυχιακή Εργασία

Τμήμα Επιστήμης Υπολογιστών
Πανεπιστήμιο Κρήτης

Περίληψη

Στη σύγχρονή μας εποχή, όλοι σχεδόν οι τομείς της Επιστήμης Υπολογιστών παρουσιάζουν ραγδαία εξέλιξη, όπως και ο αντίστοιχος της Ψηφιακής Βιβλιοθήκης. Κύρια ενασχόληση των σύγχρονων Ψηφιακών Βιβλιοθηκών δεν αποτελεί πλέον η αποθήκευση, η αναζήτηση και η πρόσβαση σε ψηφιακά αντικείμενα, αλλά περισσότερο βάρος δίνεται σε νέα είδη υπηρεσιών. Χαρακτηριστικό παράδειγμα αποτελεί η Επιστημονική Ψηφιακή Βιβλιοθήκη, η οποία και ασχολείται ιδιαίτερα με τη ψηφιακή αναπαράσταση επιστημονικών αντικειμένων. Τα εν λόγω επιστημονικά αντικείμενα είναι είτε σύνολα δεδομένων (data sets) είτε προγράμματα προσομοίωσης είτε εργαλεία στατιστικής ανάλυσης ή οπτικοποίησης συνόλων δεδομένων. Τα προβλήματα, τόσο της αναζήτησης και πρόσβασης σε σύνολα δεδομένων όσο και της εκμετάλλευσης των εργαλείων οπτικοποίησης συνόλων δεδομένων, αντιμετωπίζονται επιτυχώς με τη χρήση συγκεκριμένων τεχνικών, όπως είναι η διαχείριση μετα-δεδομένων, η χρήση οντολογιών και η τρέχουσα τεχνολογία για το διαδίκτυο. Αντιθέτως, η εκμετάλλευση των προγραμμάτων προσομοίωσης, προκειμένου να παραχθεί νέα επιστημονική πληροφορία, καθίσταται ακόμη ανέφικτη.

Στην παρούσα Μεταπτυχιακή Εργασία παρουσιάζεται και αναλύεται αρχικά το σύστημα **ARION** και επακολούθως το υποσύστημά του **K.A.P.E.E.A**. Το σύστημα **ARION** αποτελεί πλαίσιο παροχής ηλεκτρονικών υπηρεσιών (e-services). Το πλαίσιο είναι σχεδιασμένο για να υποστηρίζει την αναζήτηση και την πρόσβαση σε επιστημονικά αντικείμενα και, επιπλέον, παρουσιάζει τη δυνατότητα ενοποίησης επιστημονικών συλλογών. Επιπροσθέτως, υποστηρίζει ενεργά τη λειτουργία ροών εργασίας, που επεξεργάζονται επιστημονικά δεδομένα, όπως δορυφορικές εικόνες, time series κ.α.. Επομένως, το σύστημα **ARION** αποτελεί αφενός Ψηφιακή Βιβλιοθήκη και αφετέρου σύστημα διαχείρισης ροών εργασίας.

Το σύστημα **Κατανεμημένης Αποθήκευσης Ροών Εργασίας Επιστημονικών Αντικειμένων – K.A.P.E.E.A** αναπτύχθηκε-κατασκευάστηκε στα πλαίσια της Μεταπτυχιακής Εργασίας. Το σύστημα είναι υπεύθυνο για τον ορισμό ροών εργασίας επιστημονικών αντικειμένων με βάση τη γλώσσα **AWL** μοντελοποίησης ροών εργασίας, για την οποία και γίνεται εκτενής περιγραφή. Επίσης, είναι υπεύθυνο τόσο για την αποθήκευση των προαναφερθέντων ορισμών σε μια βάση δεδομένων όσο και για τη διεξαγωγή επερωτήσεων σε αυτήν τη βάση. Τέλος, παρέχει τη λειτουργία της καταχώρησης στατιστικής πληροφορίας για τυχαία περίπτωση ροής εργασίας, που εκτελέστηκε στο σύστημα **ARION**. Συνεπώς, σε αυτήν την αναφορά παρατίθενται και όλα τα στάδια ανάπτυξης-κατασκευής (ανάλυση απαιτήσεων, σχεδίαση και

υλοποίηση) ενός πρωτότυπου και πρότυπου υπολογιστικού συστήματος, όπως είναι το σύστημα **Κ.Α.Ρ.Ε.Ε.Α.**

Επόπτης

Χούστη Αικατερίνη
Καθηγήτρια

Τμήμα Επιστήμης Υπολογιστών
Πανεπιστήμιο Κρήτης

Distributed Storage System of Workflows of Scientific Objects

Kyriakos E. Kritikos

Master of Science Thesis

Department of Computer Science
University of Crete
Greece

Abstract

Nowadays, every area of Computer Science is facing a tremendous development. One such area is the area of the Digital Library - DL. The synchronous DLs have stopped the exclusive pursuit of the storage, search and retrieval process of digital objects and are now dealing with new types of services. One particular example of a DL is a Scientific DL. A Scientific DL mainly deals with the digital representation of scientific objects. A scientific object can be either a data set or a simulation model or an associated tool for statistical analysis and data set visualization. The problems of search and retrieval of scientific data sets and the exploitation of the visualization tools are successfully met by a Scientific Digital Library. However, the exploitation of the simulation models, that can produce new scientific information, is not tackled.

In this Master thesis, there is a presentation of two different systems. The **ARION** system is a service-based infrastructure designed to support search and retrieval of scientific objects and is capable of integrating collections of scientific applications including datasets, simulation models and associated tools for statistical analysis and dataset visualization. It also actively supports on-demand scientific data processing workflows, in both interactive and batch mode. Consequently, the **ARION** system is both a digital library and a workflow management system. The **Arion Workflow DataBase System – AWDB** is a subsystem of the **ARION** system and was built during our Master project. It is responsible for the specification of workflows with the help of a new workflow specification language, called AWL, which is introduced. In addition, it is responsible for storing the above specifications in a database and for executing queries in this database. Lastly, it provides the means of storing statistical information of an instance of a workflow that was executed in the **ARION** system. Therefore, in this Master thesis, the three phases (requirements analysis, design and implementation) of the development of a prototype system like **AWDB** are also introduced.

Advisor

Catherine Houstis
Professor

Department of Computer Science
University of Crete
Greece

**Αφιερομένο στους ανθρώπους που με έχουν στηρίξει στη ζωή μου
μέχρι σήμερα.**

Ευχαριστίες

Η παρούσα Μεταπτυχιακή Εργασία δεν θα είχε πραγματοποιηθεί χωρίς τις επισημάνσεις καθηγητών και ερευνητών του Τομέα των Παράλληλων και Κατανεμημένων Συστημάτων.

Καταρχάς, θα ήθελα να ευχαριστήσω θερμά την επόπτη καθηγήτρια κα. Χούστη Αικατερίνη για την ουσιαστική καθοδήγησή, που μου πρόσφερε σε όλες τις φάσεις της εργασίας μου. Ακολούθως, επιθυμώ να ευχαριστήσω τον επίκουρο καθηγητή του Πανεπιστημίου Θεσσαλίας κ. Λάλη Σπύρο, για την τεχνική του υποστήριξη σε αρκετά από τα προβλήματα, που ανέκυψαν κατά το Σχεδιασμό και την Υλοποίηση του υποσυστήματος **K.A.P.E.E.A.** Επίσης, θα ήθελα να ευχαριστήσω τους: κ. Πλεξουσάκη Δημήτρη, αναπληρωτή καθηγητή του Πανεπιστημίου Κρήτης (Π.Κ), και τον κ. Χριστοφίδη Βασίλη, επίκουρο καθηγητή του Π.Κ, ως μέλη της Εισηγητικής Επιτροπής, για τις πολύτιμες παρατηρήσεις τους, συντελώντας έτσι στην ολοκλήρωσή της ερευνητικής εργασίας.

Επιπροσθέτως, για την τεχνική συνεργασία τους στα πλαίσια του ερευνητικού **PROJECT ARION**, επιθυμώ να ευχαριστήσω :

Τον κ. Πιτικάκη Μάριο, μέλος της ομάδας των Κατανεμημένων Συστημάτων του ΠΠ-ΠΤΕ, καθώς και τον κ. Σμαρδά Αντώνη, μεταπτυχιακό φοιτητή του Τμήματος Επιστήμης Υπολογιστών (Ε. Η/Υ).

Ιδιαίτερα, τη δις. Ειρήνη Καγιαλή, υποψήφια μεταπτυχιακή φοιτήτρια της Ε. Η/Υ του Π.Κ, τόσο για τη ψυχολογική συμπαράστασή, που μου προσέφερε σε όλη τη διάρκεια της μεταπτυχιακής μου πορείας, όσο και για την υποστήριξή της στη διατύπωση και σύνταξη της παρούσας αναφοράς.

Τέλος, το Τμήμα Επιστήμης Υπολογιστών του Πανεπιστημίου Κρήτης και το Ινστιτούτο Πληροφορικής του Ιδρύματος Τεχνολογίας και Έρευνας για την υλικοτεχνική υποστήριξη, που μου παρέιχαν.

Κρητικός Κυριάκος

Πίνακας Περιεχομένων

ΕΥΧΑΡΙΣΤΙΕΣ	I
ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ	III
ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ-ΣΧΗΜΑΤΩΝ	VII
1. ΕΙΣΑΓΩΓΗ	1
1.1. ΑΝΤΙΚΕΙΜΕΝΟ ΤΗΣ ΕΡΓΑΣΙΑΣ.....	2
1.2. ΟΡΓΑΝΩΣΗ ΤΗΣ ΑΝΑΦΟΡΑΣ.....	3
2. ΨΗΦΙΑΚΗ ΒΙΒΛΙΟΘΗΚΗ	5
2.1. ΟΡΙΣΜΟΣ ΤΗΣ ΨΗΦΙΑΚΗΣ ΒΙΒΛΙΟΘΗΚΗΣ	5
2.2. ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ	6
2.3. ΠΛΕΟΝΕΚΤΗΜΑΤΑ	7
2.4. ΜΕΛΛΟΝΤΙΚΕΣ ΚΑΤΕΥΘΥΝΣΕΙΣ	8
2.5. ΤΕΧΝΙΚΕΣ ΑΝΑΖΗΤΗΣΗΣ – ΜΕΤΑ-ΔΕΔΟΜΕΝΑ	8
2.5.1. <i>Πρότυπα Μετα-δεδομένων</i>	9
2.5.2. <i>Οντολογίες</i>	10
2.6. ΜΕΡΙΚΕΣ ΣΗΜΑΝΤΙΚΕΣ ΨΗΦΙΑΚΕΣ ΒΙΒΛΙΟΘΗΚΕΣ	12
2.6.1. <i>Stanford Digital Library</i>	12
2.6.2. <i>UC Berkley Digital Library</i>	12
2.6.3. <i>Ariadne</i>	13
2.6.4. <i>Alexandria Digital Library</i>	13
2.6.5. <i>Master Environmental Library (MEL)</i>	13
2.6.6. <i>Earth Observing System Data and Information System (EOSDIS)</i>	14
2.6.7. <i>DODS</i>	14
3. ΡΟΕΣ ΕΡΓΑΣΙΑΣ	15
3.1. ΟΡΙΣΜΟΣ ΡΟΗΣ ΕΡΓΑΣΙΑΣ ΚΑΙ ΜΟΝΤΕΛΟΥ ΡΟΗΣ ΕΡΓΑΣΙΑΣ	15
3.2. ΔΙΑΣΤΑΣΕΙΣ ΜΙΑΣ ΡΟΗΣ ΕΡΓΑΣΙΑΣ	15
3.3. ΚΑΤΗΓΟΡΙΕΣ ΡΟΩΝ ΕΡΓΑΣΙΑΣ.....	16
3.4. ΩΦΕΛΕΙΕΣ ΑΠΟ ΡΟΕΣ ΕΡΓΑΣΙΑΣ.....	17
3.5. ΣΥΣΤΗΜΑΤΑ ΔΙΑΧΕΙΡΙΣΗΣ ΡΟΩΝ ΕΡΓΑΣΙΑΣ	18
3.5.1 <i>Build-time</i>	19
3.5.2 <i>Run-time</i>	20
3.5.3. <i>Μερικά συστήματα διαχείρισης ροών εργασίας</i>	20
3.5.3.1. <i>Exotica και WIDE</i>	20
3.5.3.2. <i>METEOR2</i>	21
3.5.3.3. <i>WebFlow and XPECT</i>	21
3.5.3.4. <i>Middleware for Enterprise-Wide Workflow Management (MENTOR)</i>	22
3.5.3.5. <i>Panta Rhei</i>	23
3.5.3.6. <i>WASA</i>	24
3.6. ΓΛΩΣΣΕΣ ΚΑΙ ΜΕΤΑ-ΓΛΩΣΣΕΣ ΟΡΙΣΜΟΥ ΡΟΩΝ ΕΡΓΑΣΙΑΣ	25
3.6.1 <i>Ορισμός Γλώσσας Προσδιορισμού Ροών Εργασίας</i>	25
3.6.2. <i>Είδη Γλωσσών Προσδιορισμού Ροών Εργασίας</i>	26
3.6.3. <i>Μετα-γλώσσες για τον προσδιορισμό διαδικασιών</i>	26
3.6.4. <i>Παραδείγματα Γλωσσών Προσδιορισμού Ροών Εργασίας</i>	27
3.6.4.1. <i>IPL και WFSL</i>	27
3.6.4.2. <i>FDL</i>	27
3.6.4.3. <i>WIDE language</i>	28
3.6.4.4. <i>XRL</i>	28
3.6.4.5. <i>Workflow Definition Language 2 (WDL2)</i>	29
3.6.5. <i>Παραδείγματα Μετα-γλωσσών Προσδιορισμού Ροών Εργασίας</i>	29
3.6.5.1. <i>Workflow Process Definition Language (WPDL)</i>	29
3.6.5.2. <i>Process Interchange Framework (PIF)</i>	31

3.6.5.3. <i>Process Specification Language (PSL)</i>	32
3.6.5.4. <i>Generalized Process Structure Grammars (GSPG)</i>	33
3.6.5.5. <i>Unified Modelling Language (UML)</i>	34
3.7. ΔΙΑΛΕΙΤΟΥΡΓΙΚΟΤΗΤΑ ΡΟΩΝ ΕΡΓΑΣΙΑΣ	36
3.7.1. <i>Η βάση του προβλήματος και η λύση των πρακτόρων</i>	36
3.7.2. <i>Η Γενική λύση</i>	37
3.7.2.1. <i>Χρήση Ορισμού Διαδικασίας Κατά Μήκος Πολλαπλών Πεδίων Εφαρμογής</i>	37
3.7.2.2. <i>Αλληλεπιδράσεις Runtime Ελέγχου</i>	38
4. ΤΟ ΣΥΣΤΗΜΑ ARION	39
4.1 ΕΙΣΑΓΩΓΗ	39
4.2. ΑΝΤΙΚΕΙΜΕΝΙΚΟΙ ΣΤΟΧΟΙ ΤΟΥ ARION	40
4.2.1. <i>Λειτουργίες και τεχνικές της Ψηφιακής Βιβλιοθήκης</i>	41
4.2.2. <i>Λειτουργίες και τεχνικές του συστήματος διαχείρισης ροών εργασίας</i>	42
4.2.3. <i>Λειτουργίες και τεχνικές του συστήματος ταυτοποίησης και εξουσιοδότησης</i>	43
4.2.4. <i>Βασικά αρχιτεκτονικά χαρακτηριστικά όλων των υποσυστημάτων</i>	43
4.3. ΣΧΕΔΙΑΣΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ ARION	43
4.3.1. <i>Εισαγωγή</i>	43
4.3.2. <i>Σχεδίαση Οντολογίας</i>	44
4.3.3. <i>Σχεδίαση του μοντέλου της επιστημονικής ροής εργασίας</i>	46
4.3.4. <i>Η αρχιτεκτονική του συστήματος ARION</i>	47
4.3.4.1. <i>Η Μηχανή Αναζήτησης</i>	49
4.3.4.1.1. Προσδιορισμοί μετα-δεδομένων	50
4.3.4.1.2. Εννοιολογικές ερωτήσεις	51
4.3.4.2. <i>Το Σύστημα Κατανεμημένης Αποθήκευσης Ροών Εργασίας Επιστημονικών Αντικειμένων</i>	52
4.3.4.3. <i>Το Σύστημα Εκτέλεσης Ροών Εργασίας</i>	53
4.3.4.3.1. Μέρη Βρισκόμενα στον Υπηρετητή	54
4.3.4.3.2. Μέρη βρισκόμενα στον πελάτη	55
4.3.4.4. <i>Το Σύστημα Ταυτοποίησης και Εξουσιοδότησης Χρήστη</i>	56
4.3.4.4.1. Η διαδικασία Ταυτοποίησης και Εξουσιοδότησης	56
4.3.4.5. <i>Ειδικές λεπτομέρειες-χαρακτηριστικά</i>	57
4.3.4.5.1. Ειδικές Λεπτομέρειες	57
4.3.4.5.2. Ειδικά χαρακτηριστικά	58
4.3.5. <i>Διεπαφές Χρήσης</i>	59
4.4. ΣΥΜΠΕΡΑΣΜΑΤΑ ΓΙΑ ΤΟ ΣΥΣΤΗΜΑ ARION	61
5. ΤΟ ΣΥΣΤΗΜΑ ΚΑΤΑΝΕΜΗΜΕΝΗΣ ΑΠΟΘΗΚΕΥΣΗΣ ΡΟΩΝ ΕΡΓΑΣΙΑΣ ΕΠΙΣΤΗΜΟΝΙΚΩΝ ΑΝΤΙΚΕΙΜΕΝΩΝ	63
5.1. ΕΙΣΑΓΩΓΗ	63
5.2. ΣΤΟΧΟΙ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ	63
5.3. ΣΧΕΔΙΑΣΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ	66
5.3.1. <i>Σχεδίαση της γλώσσας ορισμού ροών εργασίας</i>	66
5.3.1.1. <i>Ορισμός μοντέλου ροής εργασίας</i>	66
5.3.1.2. <i>Σχεδίαση γλώσσας ροής εργασίας</i>	69
5.3.2. <i>Αρχιτεκτονική του συστήματος</i>	78
5.3.2.1. <i>Χαρακτηριστικά της αρχιτεκτονικής του συστήματος Κ.Α.Ρ.Ε.Ε.Α</i>	85
5.4. ΥΛΟΠΟΙΗΣΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ	89
5.5. ΕΠΕΚΤΑΣΕΙΣ-ΑΝΟΙΚΤΑ ΘΕΜΑΤΑ	101
5.5.1. <i>Προσθήκη Οικονομικού Μοντέλου</i>	101
5.5.2. <i>Αναθεώρηση της ARION Workflow Language-AWL</i>	102
5.5.3. <i>Διαλειτουργικότητα ροών εργασίας</i>	103
5.5.4. <i>Χρησιμοποίηση μιας XML Βάσης Δεδομένων</i>	104
5.5.5. <i>Επεκτάσεις στις Επερωτήσεις</i>	105
5.5.6. <i>Επεκτάσεις στους Κειμενογράφους</i>	106
5.6. ΣΥΜΠΕΡΑΣΜΑΤΑ	107
ΠΑΡΑΡΤΗΜΑΤΑ	109
Α. ΤΟ DOCUMENT TYPE DEFINITION ΤΗΣ ΓΛΩΣΣΑΣ XRL	109
Β. ΤΟ DOCUMENT TYPE DEFINITION ΤΗΣ ΓΛΩΣΣΑΣ AWL	111
Γ. ΠΕΡΙΓΡΑΦΗ ΚΕΙΜΕΝΟΓΡΑΦΟΥ (ΠΕΡΙΠΤΩΣΕΩΝ) ΡΟΩΝ ΕΡΓΑΣΙΑΣ	114
Γ.1. <i>Εισαγωγή</i>	114

<i>Γ.ΙΙ. Ο Κειμενογράφος Ροών Εργασίας</i>	114
Βασικές Λειτουργίες	114
Η Γραφική Διεπαφή Χρήσης του Κειμενογράφου	116
<i>Γ.ΙΙΙ. Ο Κειμενογράφος Περιπτώσεων Ροών Εργασίας</i>	125
Βασικές Λειτουργίες	125
Η Γραφική Διεπαφή Χρήσης του Κειμενογράφου	126
<i>Γ.ΙV. Ένα Παράδειγμα Χρήσης των Κειμενογράφων</i>	131
ΒΙΒΛΙΟΓΡΑΦΙΑ	135

Πίνακας Εικόνων-Σχημάτων

Σχήμα 1 - Αρχιτεκτονική του MENTOR project.....	23
Σχήμα 2 - Αρχιτεκτονική του WASA2 project.....	24
Σχήμα 3 - Workflow Reference Model.....	29
Σχήμα 4 - Το μετα-μοντέλο του μέρους της διαδικασίας της WPDL	30
Σχήμα 5 - Η ιεραρχία των μερών του πυρήνα της PIF	31
Σχήμα 6 - Η δομή της οντολογίας για κυματικό κλίμα	45
Σχήμα 7 - Το μοντέλο της επιστημονικής ροής εργασίας	46
Σχήμα 8 - Αρχιτεκτονική του συστήματος ARION	48
Σχήμα 9 - Το σύστημα ARION αποτελεί σύστημα middleware	49
Σχήμα 10 - Αρχιτεκτονική της Μηχανής Αναζήτησης.....	50
Σχήμα 11 - Η αρχιτεκτονική του συστήματος AWDB.....	52
Σχήμα 12 - Αρχιτεκτονική του Workflow Runtime System.....	53
Σχήμα 13 - Η διαδικασία ταυτοποίησης χρήστη	56
Εικόνα 14 - Κατανεμημένη αρχιτεκτονική τριών δεσμών του συστήματος ARION..	58
Σχήμα 15 - Οι τρεις περιπτώσεις backtracking.....	68
Σχήμα 16 - Το νέο μοντέλο της επιστημονικής ροής εργασίας.....	69
Σχήμα 17 - Η κεντρική αρχιτεκτονική του συστήματος K.A.P.E.E.A	80
Σχήμα 18 - Η αρχιτεκτονική ενός τοπικού συστήματος K.A.P.E.E.A	84
Σχήμα 19 - Αρχιτεκτονική Ειδοποιητή και DNS Server	85
Σχήμα 20 - Αρχιτεκτονική του Workflow Storage Server	95
Σχήμα 21 - Αρχιτεκτονική του Statistic Database Server.....	97
Σχήμα 22 - Η νέα αρχιτεκτονική του K.A.P.E.E.A	100
Εικόνα 23 - Ο Κειμενογράφος Ροών Εργασίας.....	115
Εικόνα 24 - Η φόρμα Ταυτοποίησης Χρήστη	116
Εικόνα 25 - Φόρμα Ιδιοτήτων Απομακρυσμένου Υπολογιστή	117
Εικόνα 26 - Επιλογή του Απομακρυσμένου Αρχείου.....	118
Εικόνα 27 - Η Φόρμα Επιλογής Ιδιοτήτων Ροής Εργασίας.....	118
Εικόνα 28 - Φόρμα Επερώτησης Πληροφορίας για Έργο	120
Εικόνα 29 - Η Φόρμα Ιδιοτήτων Συνόλου Δεδομένων	121
Εικόνα 30 - Η Φόρμα Ιδιοτήτων Παραμέτρου	123
Εικόνα 31 - Η Φόρμα Ιδιοτήτων Προγράμματος	123
Εικόνα 32 - Η Φόρμα Επιλογής Ονόματος Κλάσης της Επιστημονικής Οντολογίας	124
Εικόνα 33 - Η Φόρμα Ιδιοτήτων του Στοιχείου 'While-do'	124
Εικόνα 34 - Ένας πλήρης προσδιορισμός ροής εργασίας.....	125
Εικόνα 35 - Ο Κειμενογράφος Περιπτώσεων Ροών Εργασίας.....	126
Εικόνα 36 - Η Φόρμα Επιλογής της Ροής Εργασίας 1	127
Εικόνα 37 - Φόρμα Επιλογής Ροής Εργασίας 2.....	128
Εικόνα 38 - Φόρμα Εισαγωγής Ιδιοτήτων Περίπτωσης Ροής Εργασίας.....	128
Εικόνα 39 - Απλή Περίπτωση Ροής Εργασίας.....	129
Εικόνα 40 - Φόρμα Επιλογής Περίπτωσης Συνόλου Δεδομένων.....	130
Εικόνα 41 - Περίπτωση Ροής Εργασίας που τρέχει στο background	130

1. Εισαγωγή

Τα τελευταία χρόνια, οι ερευνητές στηρίζονται ολοένα και περισσότερο στην τεχνολογία των υπολογιστών και της επικοινωνίας, προκειμένου να επιτελέσουν ένα σημαντικό μέρος της καθημερινής τους έρευνας. Η Ηλεκτρονική Επιστήμη (e-Science) προσφέρει ένα όραμα για το πώς θα χρησιμοποιηθεί η τεχνολογία των υπολογιστών και της επικοινωνίας με στόχο την υποστήριξη και τη βελτίωση των επιστημονικών διαδικασιών, παρέχοντας νέους και πιο ισχυρούς τρόπους εργασίας. Επιπλέον, παρέχοντας καταναμημένους, ασφαλής και υψηλών επιδόσεων μηχανισμούς πρόσβασης σε απομακρυσμένους πόρους, ενισχύει αφενός τη δυναμική επιστημονικών συνεργασιών με το διαμοιρασμό πόρων σε μια πρωτοφανή κλίμακα και αφετέρου τη συνεργασία γεωγραφικά καταναμημένων ομάδων με τρόπους που άλλοτε δεν ήταν εφικτοί.

Η έννοια του διαμοιρασμού καταναμημένων πόρων είτε πληροφοριακών είτε υπολογιστικών δεν είναι πρόσφατη. Τώρα, όμως, ο συνδυασμός των τάσεων της τεχνολογίας με τα ερευνητικά επιτεύγματα καθιστά δυνατή την πραγματοποίηση του οράματος, το οποίο και ονομάζεται *'The Grid'*. Το σύστημα **ARION** αναπτύχθηκε χρησιμοποιώντας τα αποτελέσματα δύο προσφάτως ανακινημένων ερευνητικών δραστηριοτήτων: του Σημασιολογικού Δικτύου (Semantic Web) και του Πλέγματος (Grid). Μέσω της πρώτης δραστηριότητας, η γλώσσα RDF/S προωθεί τη σημασιολογική διαλειτουργικότητα στο επίπεδο της μοντελοποίησης πληροφοριών με τη χρήση οντολογιών και αντίστοιχων εργαλείων (όπως είναι το RDFSuite), ενώ μέσω της δεύτερης δραστηριότητας, ορίζεται ένα υπολογιστικό πλαίσιο για το σύστημα **ARION**. Επιπροσθέτως, τόσο οι προσδιορισμοί προτύπων μετα-δεδομένων όσο και τα συντακτικά πρότυπα δια-λειτουργικότητας, αποτελούν ευθύ και πολύτιμη συνεισφορά στην υλοποίηση βασικών υπολογιστικών λειτουργιών και λειτουργιών οπτικοποίησης.

Το σύστημα **ARION** αποτελεί πλαίσιο παροχής ηλεκτρονικών υπηρεσιών (e-services). Παρουσιάζει δυνατότητα ενοποίησης επιστημονικών συλλογών και είναι σχεδιασμένο για να υποστηρίζει αφενός την αναζήτηση και την πρόσβαση σε επιστημονικά αντικείμενα και αφετέρου τη λειτουργία ροών εργασίας, που επεξεργάζονται επιστημονικά δεδομένα, όπως δορυφορικές εικόνες, time series κ.α. Επομένως, το σύστημα **ARION** συνδυάζει και τις λειτουργίες μιας Ψηφιακής Βιβλιοθήκης και τις λειτουργίες ενός συστήματος διαχείρισης ροών εργασίας. Θα αποτελείται, άρα, από δύο ανεξάρτητα υποσυστήματα που θα επικοινωνούν μεταξύ τους, καθένα από τα οποία θα παρέχει τις προαναφερθέντες λειτουργίες. Μάλιστα, το σύστημα της Ψηφιακής Βιβλιοθήκης είναι ήδη υλοποιημένο, μέσω του πακέτου εργαλείων RDFSuite, εκτός από μερικά ιδιαίτερα κομμάτια του. Αντιθέτως, το σύστημα διαχείρισης ροών εργασίας οφείλει να αναπτυχθεί από την αρχή, και ο λόγος παρατίθεται ακολούθως.

Τα περισσότερα συστήματα διαχείρισης ροών εργασίας, που διατίθενται σήμερα ως προϊόντα ή ως ερευνητικά επιτεύγματα, είναι κεντρικοποιημένα. Υποφέρουν, δηλαδή, από έλλειψη κλιμάκωσης (scalability) και ταυτοχρονισμού (concurrency), ενώ παράλληλα δεν παρουσιάζουν ανεκτικότητα σε λάθη. Επιπροσθέτως, τα καταναμημένα συστήματα διαχείρισης ροών εργασίας δεν είναι αρκετά αποδοτικά και lightweight. Το πρόβλημα διαιωνίζεται, ακόμη περισσότερο, από την έλλειψη μίας καθολικά αποδεκτής πρότυπης γλώσσας ορισμού ροών εργασίας. Η έλλειψη έγκειται στο γεγονός, ότι σχεδόν όλες οι γλώσσες ορισμού ροών εργασίας είτε είναι φτωχές από πλευράς εκφραστικότητας είτε εξαρτώνται από την εκάστοτε υπολογιστική εφαρμογή. Μάλιστα, για το λόγο αυτό έχουν αναπτυχθεί και

μετα-γλώσσες ορισμού ροών εργασίας, ώστε να προσφέρουν δια-λειτουργικότητα μεταξύ συστημάτων διαχείρισης ροών εργασίας που χρησιμοποιούν διαφορετικές γλώσσες ορισμού ροών εργασίας. Συνεπώς, κρίνεται αναγκαία η ανάπτυξη του υποσυστήματος διαχείρισης των ροών εργασίας από την αρχή και πάντοτε σύμφωνα με τις προδιαγραφές και απαιτήσεις του συστήματος **ARION**, προκειμένου να παρέχει τις κατάλληλες λειτουργίες, να συνεργάζεται με το υποσύστημα της Ψηφιακής Βιβλιοθήκης και, τέλος, να στηρίζεται σε κατάλληλα αρχιτεκτονικά και τεχνικά χαρακτηριστικά.

1.1. Αντικείμενο της εργασίας

Ένα σύστημα διαχείρισης ροών εργασίας αποτελείται από τρία βασικά μέρη. Το πρώτο μέρος, το κατασκευαστικό, ασχολείται με τον ορισμό και την αποθήκευση ροών εργασίας. Ο ορισμός ροών εργασίας γίνεται με βάση μια γλώσσα ορισμού ροών εργασίας, ενώ η αποθήκευση των προαναφερθέντων ορισμών πραγματοποιείται στο δεύτερο μέρος ενός συστήματος διαχείρισης ροών εργασίας, τη βάση δεδομένων. Σε αυτήν τη βάση αποθηκεύεται και στατιστική πληροφορία για περιπτώσεις ροών εργασίας που εκτελέστηκαν. Το τρίτο μέρος είναι το εκτελεστικό μέρος, το οποίο είναι υπεύθυνο για την εκτέλεση και την παρακολούθηση περιπτώσεων ροών εργασίας.

Σκοπός της παρούσας Μεταπτυχιακής Εργασίας είναι η παρουσίαση και η ανάλυση ενός πρωτότυπου και πρότυπου συστήματος **Κατανεμημένης Αποθήκευσης Ροών Εργασίας Επιστημονικών Αντικειμένων – Κ.Α.Ρ.Ε.Ε.Α.**, το οποίο και ενοποιεί τα δύο πρώτα βασικά μέρη ενός συστήματος διαχείρισης ροών εργασίας. Επιπλέον, στηριζόμενο στη σωστή σχεδίασή και υλοποίησή του, πραγματοποιεί τις κατάλληλες λειτουργίες και τεχνολογικές απαιτήσεις του συστήματος **ARION**. Θα αποδειχθεί ότι το σύστημα **Κ.Α.Ρ.Ε.Ε.Α** είναι αξιόπιστο, ανοικτό, ελαφρύ, κατανεμημένο, ασφαλές, κλιμακώσιμο, ταυτοχρονικό και ότι βασίζεται σε διεθνή πρότυπα. Δικαίως, λοιπόν, προσδίδεται σε αυτό ο τίτλος του πρωτότυπου και πρότυπου συστήματος.

Επιπροσθέτως, θα παρουσιαστεί όχι μόνο το μοντέλο επιστημονικών ροών εργασίας, όπως προκύπτει από την εφαρμογή του **ARION** σε περιβαλλοντολογικά συστήματα, αλλά και η γλώσσα ορισμού ροών εργασίας με όνομα **AWL**, που υλοποιεί το εν λόγω μοντέλο. Η γλώσσα **AWL** αποτελεί τροποποίηση της γλώσσας **XRL**, τόσο στο κομμάτι της εκφραστικότητας όσο και στο κομμάτι της σημασιολογίας. Παρουσιάζει τη δυνατότητα παροχής ορισμών τόσο για ροές εργασίας όσο και για περιπτώσεις ροών εργασίας. Επιπλέον, εκφράζεται σε αρχείο τύπου **DTD**, οπότε οι ορισμοί (περιπτώσεων) ροών εργασίας θα μπορούν να εκφράζονται σε αρχεία τύπου **XML**. Παρουσιάζει δύο σημαντικά χαρακτηριστικά/πλεονεκτήματα. Το πρώτο είναι ότι το σχήμα μιας ροής εργασίας μπορεί να τροποποιείται, καθώς οι περιπτώσεις της εκτελούνται. Το δεύτερο χαρακτηριστικό είναι ότι η γλώσσα **AWL** μπορεί να αντιστοιχιστεί στα δίκτυα **Petri Nets**. Συνεπώς, η γλώσσα μας και συμμορφώνεται με διεθνή πρότυπα, όπως είναι η **XML** και η **XRL**, αλλά και παρουσιάζει αρκετά πλεονεκτήματα. Άρα, αποτελεί μια σύγχρονη και πρότυπη γλώσσα ορισμού επιστημονικών ροών εργασίας.

1.2. Οργάνωση της Αναφοράς

Η αναφορά της Μεταπτυχιακής Εργασίας οργανώθηκε ως εξής :

- Στο κεφάλαιο 2, αναλύεται η έννοια της Ψηφιακής Βιβλιοθήκης και των μετα-δεδομένων. Επίσης, αναφέρεται η συνεισφορά των οντολογιών στην ανάπτυξη μιας Ψηφιακής Βιβλιοθήκης. Έπειτα, παρουσιάζονται, σημαντικά από πλευράς λειτουργικότητας, συστήματα Ψηφιακής Βιβλιοθήκης.
- Στο κεφάλαιο 3, αναλύονται διεξοδικά οι έννοιες της ροής εργασίας, του μοντέλου ροής εργασίας και του συστήματος διαχείρισης ροών εργασίας. Επίσης, παρουσιάζονται και περιγράφονται τα πιο σημαντικά καταναμημένα συστήματα διαχείρισης ροών εργασίας. Τέλος, παρουσιάζονται και αναλύονται οι πιο γνωστές γλώσσες και μετα-γλώσσες ορισμού ροών εργασίας.
- Στο κεφάλαιο 4, παρουσιάζεται το σύστημα **ARION** και αναλύονται τα δύο πρώτα στάδια της ανάπτυξής του, τα οποία αφορούν την ανάλυση απαιτήσεων και τη σχεδίαση. Η υλοποίηση δεν αναφέρεται, διότι το εν λόγω σύστημα είναι υπό εξέλιξη.
- Στο κεφάλαιο 5, παρουσιάζεται το σύστημα **K.A.P.E.E.A** και αναλύονται όλα τα στάδια της ανάπτυξής του. Επίσης, αναφέρονται και πιθανές μελλοντικές επεκτάσεις-τροποποιήσεις του εν λόγω συστήματος. Τέλος, πραγματοποιείται μικρή ανασκόπηση της συνολικής παράθεσης του συστήματος **K.A.P.E.E.A**.

2. Ψηφιακή Βιβλιοθήκη

2.1. Ορισμός της Ψηφιακής Βιβλιοθήκης

Η έννοια της Ψηφιακής Βιβλιοθήκης είναι απολύτως συνυφασμένη με την έννοια της κοινής βιβλιοθήκης. Αφενός διότι περιέχει αντικείμενα, τα οποία είναι τοποθετημένα με οργανωμένο τρόπο και σε συγκεκριμένο σημείο στο τεχνητό χώρο της, και αφετέρου διότι προσφέρει υπηρεσίες ταχείας εύρεσης ή και αγοράς ενός αντικείμενου. Εντούτοις, η ψηφιακή βιβλιοθήκη δεν αποτελεί μια κοινή, συμβατική βιβλιοθήκη. Καταρχάς, διότι παρουσιάζεται σε ψηφιακή μορφή. Αυτό σημαίνει ότι περιλαμβάνει ψηφιακά αντικείμενα, όπως ηλεκτρονικά έγγραφα και αρχεία, ηλεκτρονικά μηνύματα, ψηφιακές εικόνες, βίντεο κ.λ.π. Ακολούθως, διότι τα αντικείμενα που περιέχει μπορεί να μην βρίσκονται σε μια συγκεκριμένη τοποθεσία πρόσβασης (υπολογιστή). Κατά συνέπεια, είναι μια τεχνητή βιβλιοθήκη, αφού τα αντικείμενα που περιέχει βρίσκονται καταναμημένα σε πολλούς ηλεκτρονικούς υπολογιστές μέσα στο δίκτυο.

Έχουν διατυπωθεί αρκετοί ορισμοί για το τι είναι μια Ψηφιακή Βιβλιοθήκη. Εμείς θα αναφέρουμε τους δύο πιο αντιπροσωπευτικούς. Σύμφωνα με τον πρώτο ορισμό, Ψηφιακή Βιβλιοθήκη αποτελεί κάθε οργανωμένη συλλογή ψηφιακής πληροφορίας, η οποία συνδυάζει όχι μόνο τη δόμηση και τη συλλογή πληροφορίας, που πάντα προσέφεραν οι συμβατικές βιβλιοθήκες και οι κατάλογοι αρχείων, αλλά και τη ψηφιακή αναπαράσταση, που οι υπολογιστές έχουν κάνει εφικτή. Ο δεύτερος ορισμός αναφέρει ότι η ψηφιακή βιβλιοθήκη είναι μια διαχειριζόμενη συλλογή από πληροφορία, με τις αντίστοιχες υπηρεσίες, όπου η πληροφορία είναι αποθηκευμένη σε ψηφιακή μορφή και είναι προσβάσιμη μέσω δικτύου. Το σημαντικό μέρος αυτού του ορισμού είναι ότι η πληροφορία καθίσταται πλέον διαχειριζόμενη. Συνεπώς, ένα σύνολο από δεδομένα, που στέλνονται στη γη μέσω ενός δορυφόρου δεν είναι μια βιβλιοθήκη. Όμως, αν τα ίδια δεδομένα οργανωθούν συστηματικά τότε μπορεί να μετατραπούν σε μια συλλογή ψηφιακής βιβλιοθήκης.

Όπως μπορεί να αντιληφθεί κανείς, δεν υπάρχει ένας πλήρης ορισμός για το τι είναι μια Ψηφιακή Βιβλιοθήκη. Άλλωστε, πολλοί αναφέρουν τις Ψηφιακές Βιβλιοθήκες και ως 'τεχνητές' βιβλιοθήκες ή ως ηλεκτρονικές βιβλιοθήκες. Για να κάνουμε ξεκάθαρη στον αναγνώστη την έννοια της Ψηφιακής Βιβλιοθήκης, θα διακρίνουμε, αρχικά, τα πέντε χαρακτηριστικά στοιχεία της και ακολούθως θα δώσουμε ένα δικό μας αρκετά περιεκτικό ορισμό.

Τα πέντε αντιπροσωπευτικά στοιχεία είναι τα ακόλουθα :

- Δεν είναι *απλή οντότητα*.
- Χρειάζεται την τεχνολογία, προκειμένου, αρχικά, να συνδεθεί με πολλαπλές πηγές, και εν συνεχεία να αντλήσει και να φιλτράρει τη συνεχόμενη ροή πληροφοριών που δέχεται από το σύνολό τους.
- Οι συνδέσεις ανάμεσα στη Ψηφιακή Βιβλιοθήκη και στις πληροφοριακές υπηρεσίες είναι αφανής στους τελικούς χρήστες.
- Βασικός στόχος είναι η καθολικά παγκόσμια πρόσβαση.
- Οι συλλογές, που την αποτελούν, δεν περιορίζονται στο να περιέχουν μόνο ηλεκτρονικά κείμενα, αλλά παράλληλα να περιέχουν και ψηφιακά τεχνουργήματα, τα οποία δεν είναι εφικτό να αναπαρασταθούν ή να καταναμηθούν σε τυπωμένες μορφές.

Επομένως, ένας πλήρης ορισμός της Ψηφιακή Βιβλιοθήκη είναι ο εξής: **Ψηφιακή Βιβλιοθήκη** είναι ο ευρύτατα αποδεκτός όρος που περιγράφει τη χρήση των ψηφιακών τεχνολογιών για την ανάκτηση, αποθήκευση, διατήρηση και πρόσβαση σε πληροφορία και υλικό, τα οποία είτε είναι από την αρχή σε ψηφιακή μορφή είτε ψηφιοποιήθηκαν από υπάρχουσες τυπωμένες, οπτικο-ακουστικές ή άλλες μορφές.

2.2. Ιστορική αναδρομή

Ιστορικά, η έννοια της Ψηφιακής Βιβλιοθήκης είχε τεθεί και προσδιορισθεί πριν την ανάπτυξη στην επιστήμη και την τεχνολογία των ηλεκτρονικών υπολογιστών. Η πρώτη προσέγγιση της εν λόγω έννοιας έγινε το 1945 από τον *Vannevar Bush* στο άρθρο του *'As we may think'*, όπου και αναφέρει το *'Memex'*, μια συσκευή η οποία προσέφερε στους επιστήμονες την δυνατότητα να συλλέγουν, να ανιχνεύουν και να ανακτούν την (επιστημονική) πληροφορία. Η δεύτερη αναφορά έγινε από τον *Ted Nelson* το 1963 στο project του με τίτλο *'Xanadu'*, όπου προσεγγίζει την έννοια του υπερ-κειμένου (hypertext). Η τρίτη και πληρέστερη αναφορά δόθηκε από τον *J.C.R. Licklider* το 1965 στο βιβλίο του *'Libraries of the future'*, όπου περιγράφει την έρευνα και την ανάπτυξη που χρειάζεται για την κατασκευή μιας αντικειμενικά χρήσιμης Ψηφιακής Βιβλιοθήκης. Πρέπει να σημειωθεί, ότι το συγκεκριμένο βιβλίο θεωρείται ένα από τα κλασσικότερα στην ιστορία των ψηφιακών βιβλιοθηκών.

Εκτός από τις αρκετά επιθυμητές και πολύτιμες υπηρεσίες που δύναται να προσφέρει η Ψηφιακή Βιβλιοθήκη, η αύξηση του ενδιαφέροντος και η μεγάλη ώθηση για τη δημιουργία και την ανάπτυξη της, αποδίδονται στα τεχνολογικά επιτεύγματα της Επιστήμης Υπολογιστών, από το δεύτερο μισό του εικοστού αιώνα έως σήμερα. Τα σημαντικότερα από αυτά είναι τα ακόλουθα :

- ❖ Το κόστος της αποθήκευσης ψηφιακών δεδομένων μειώνεται κατά 30% ανά χρόνο.
 - Το 1987, η αποθήκευση σε CD-ROM γίνεται πιο φθηνή συγκριτικά με την αποθήκευση βιβλίων σε μία κοινή βιβλιοθήκη.
 - Σήμερα, η αποθήκευση πληροφορίας σε υπολογιστές είναι πιο οικονομική από την αποθήκευση σε αντίστοιχα φυσικά μέσα (χαρτί).
- ❖ Οι οθόνες των ηλεκτρονικών υπολογιστών γίνονται ολοένα και πιο ευχάριστες στη χρήση τους, καθώς βελτιώνονται προοδευτικά. Ολοένα και περισσότερα άτομα ενημερώνονται, διαβάζουν απευθείας από τον υπολογιστή μέσω των ηλεκτρονικών βιβλίων (e-books).
- ❖ Υψηλής ταχύτητας δίκτυα έχουν απλωθεί ευρύτατα ανά τον κόσμο.
- ❖ Οι υπολογιστές έχουν γίνει φορητοί :
 - Τα laptops μπορούν να συνδεθούν στο Internet από οπουδήποτε.
 - Επιπλέον, έχουν γίνει πιο ισχυροί και το σημαντικότερο περισσότερο οικονομικά προσιτοί.

2.3. Πλεονεκτήματα

Το μεγαλύτερο κίνητρο για την ανάπτυξη μιας Ψηφιακής Βιβλιοθήκης είναι οι υπηρεσίες-πλεονεκτήματα, που προσφέρει ή δύναται να προσφέρει. Οι σημαντικότερες υπηρεσίες αναφέρονται ακολούθως :

- ❖ Η Ψηφιακή Βιβλιοθήκη φέρνει τη βιβλιοθήκη στον χρήστη:
 - Οι πληροφορίες έρχονται στο χρήστη, είτε αυτός βρίσκεται σε προσωπικό χώρο είτε σε εργασιακό.
 - Ο χρήστης δεν χρειάζεται να επισκεφθεί το κτήριο μιας πραγματικής βιβλιοθήκης, αλλά 'φορτώνει' μια τεχνητή στον υπολογιστή του.
 - Υπάρχει βιβλιοθήκη εφόσον υπάρχει προσωπικός υπολογιστής, που είναι συνδεδεμένος στο δίκτυο.
- ❖ Έρευνα και Αναδίφηση:
 - Υποστήριξη αναζήτησης πλήρους κειμένου (full text search), πράγμα δύσκολο να γίνει σε υλικό βασιζόμενο σε χαρτί.
 - Τα συστήματα αναζήτησης βελτιώνονται.
 - Η σύνδεση υπερ-κειμένων (hypertext) υποστηρίζει τόσο την αναζήτηση πληροφορίας όσο και την αναδίφηση (browse) της.
- ❖ Η πληροφορία μπορεί να διαμοιραστεί πιο εύκολα:
 - ψηφιακή πληροφορία τοποθετούμενη στο δίκτυο γίνεται διαθέσιμη σε όλους όσους είναι συνδεδεμένοι σε αυτό – οι τόποι καθρέφτες (mirror sites) βελτιώνουν ακόμη περισσότερο την πρόσβαση – αντίθετα η δημιουργία αντιγράφου από χαρτί είναι αρκετά ακριβό και χρονοβόρο.
- ❖ Είναι εύκολη τόσο η διατήρηση, όσο και η ανανέωση της πληροφορίας.
- ❖ Η πληροφορία είναι πάντα διαθέσιμη:
 - Δεν περιορίζεται ούτε χρονικά ούτε γεωγραφικά.
 - Τα ψηφιακά υλικά δεν νοικιάζονται, δεν τοποθετούνται λανθασμένα ούτε κλέπτονται.
- ❖ Νέες μορφές πληροφορίας γίνονται δυνατές:
 - Η ψηφιακή αναπαράσταση μπορεί να υποστηρίξει χαρακτηριστικά και χειρισμούς που δεν είναι δυνατά να αναπαρασταθούν σε τυπωμένη μορφή (χημικές δομές, μαθηματικές εξισώσεις, πολυμέσα).
- ❖ Ενίσχυση συνεργασίας και συναλλαγής ιδεών:
 - Η τεχνολογία των ψηφιακών βιβλιοθηκών είναι άρρηκτα συνδεδεμένη με το ηλεκτρονικό μήνυμα και την τηλε-συνδιάσκεψη.
 - Υπάρχει τάση για σύγκλιση των άνω, καθώς και ολοκλήρωσης με τη Διαχείριση Γνώσης.
- ❖ Η Ψηφιακή Βιβλιοθήκη μπορεί να εξοικονομήσει χρήματα:
 - Τα hard data δεν είναι ακόμη διαθέσιμα.
 - Οι συμβατικές βιβλιοθήκες είναι πολύ ακριβές διότι χρειάζονται κτήρια, επαγγελματικό προσωπικό και καλή διατήρηση.
 - Όσο το τεχνολογικό κόστος θα μικραίνει και καλύτερα εργαλεία θα γίνονται διαθέσιμα, η δημιουργία Ψηφιακής Βιβλιοθήκης θα γίνεται ολοένα και πιο οικονομική.

2.4. Μελλοντικές Κατευθύνσεις

Εντούτοις, ο τομέας της Ψηφιακής Βιβλιοθήκης εμφανίζει αρκετούς περιορισμούς-μειονεκτήματα ακόμη και στη σύγχρονη μας εποχή. Ακολούθως παραθέτουμε τα σημαντικότερα από αυτά τα θέματα :

- ❖ Η τεχνολογική παλαίωση:
 - του υλικού. Το μεγαλύτερο ρίσκο για την ύπαρξη ψηφιακών αντικειμένων δεν είναι η φυσική τους φθορά, αλλά η τεχνολογική παλαίωση των συσκευών που τα διαβάζουν. Παρόλο που η ζωή των οπτικών και οπτικο-μαγνητικών ταινιών διαρκεί πολλές δεκαετίες, η αντίστοιχη των συσκευών ανάγνωσης διαρκεί το πολύ μια δεκαετία.
 - του λογισμικού. Έχει σημειωθεί ότι η ποικιλία των μορφών λογισμικού ξεπερνά κατά πολύ τον αριθμό των συσκευών υλικού που κατασκευάζονται, και ότι αυτά τα προγράμματα έρχονται και φεύγουν αρκετά πιο γρήγορα. Ίσως μια λύση να είναι οι ψηφιακές βιβλιοθήκες να βασίζονται σε πρότυπα όπως το SGML[14].
- ❖ Το κόστος της ανανέωσης περιεχομένου. Η ψηφιακή διατήρηση είναι μια συνεχιζόμενη διαδικασία που απαιτεί ένα αρκετά επαναλαμβανόμενο κόστος.
- ❖ Η διαχείριση δικαιωμάτων. Είναι πολύ εύκολο να αντιγράψεις, να αυτό-διπλασιάσεις και να κατανείμεις ψηφιακή πληροφορία. Η εξασφάλιση της προστασίας των πνευματικών δικαιωμάτων είναι αναμφισβήτητα σημαντικό θέμα σε ένα ψηφιακό περιβάλλον.
- ❖ Η διαλειτουργικότητα. Ένα κρίσιμο ζήτημα είναι αυτό της διαλειτουργικότητας διαφορετικών ψηφιακών βιβλιοθηκών. Δεδομένης της κατανομής των πηγών μιας βιβλιοθήκης στον παγκόσμιο ιστό, προκύπτει ότι χρειαζόμαστε αφενός, μεθόδους αναζήτησης μεμονωμένων κομματιών και συλλογών σε διαφορετικά μέρη και αφετέρου, μεθόδους ένωσης τεχνητών συλλογών.
- ❖ Το bandwidth του δικτύου. Με τον εκρηκτικό αριθμό των χρηστών, του περιεχομένου των πολυμέσων, των τόπων δικτύου (web sites) και των εφαρμογών, το διαθέσιμο bandwidth για την πρόσβαση στις ψηφιακές βιβλιοθήκες θα αποτελεί επί μακρόθεν ένα σημαντικό ζήτημα. Ακόμη και ένα απλής ταχύτητας CD-ROM drive παρέχει γρηγορότερη πρόσβαση από ένα 56 KBPS modem.
- ❖ Πολλά και ποικίλα είναι τα είδη των οργανισμών (online υπηρεσίες, βιβλιοθήκες, βιβλιοπωλεία, εκδότες, εταιρείες), που διεκδικούν το δικαίωμα για τη νόμιμη προμήθεια της πληροφορίας για το desk-top. Είναι προφανές λοιπόν, ότι σε ένα ψηφιακό περιβάλλον δικτύου, οι βιβλιοθήκες δεν έχουν μοναδική διεκδίκηση για την παροχή λόγιας πληροφορίας.

2.5. Τεχνικές Αναζήτησης – Μετα-δεδομένα

Ο τρόπος με τον οποίο γίνεται η αναζήτηση στα δεδομένα μιας συλλογής αποτελεί ένα μεγάλο τμήμα της έρευνας στον τομέα της Ψηφιακής Βιβλιοθήκης. Η αναζήτηση στα ίδια τα δεδομένα όχι μόνο θα ήταν πολύ αργή, αλλά θα ήταν και ανακριβής. Το τελευταίο οφείλεται στο γεγονός, ότι ένα κείμενο μπορεί να περιέχει λέξεις που δεν ανήκουν αναγκαστικά στην ίδια θεματική περιοχή στην οποία ανήκει

το κείμενο το ίδιο. Επομένως, μια αναζήτηση με μια λέξη κλειδί θα επέστρεφε πολλά κείμενα άσχετα με το θεματικό ενδιαφέρον του χρήστη. Για την εξάλειψη αυτού του φαινομένου επιστρατεύτηκαν τα μετα-δεδομένα. Αυτά είναι ειδικά αρχεία, που περιέχουν την ακριβή περιγραφή του ψηφιακού αντικειμένου. Παραδείγματος χάριν, στη περίπτωση ενός κειμένου, τα μετα-δεδομένα συνήθως περιέχουν όλες τις βιβλιογραφικές πληροφορίες για το κείμενο, όπως τον τίτλο, τους συγγραφείς, τον εκδότη, την ημερομηνία δημοσίευσης κ.τ.λ.. Πολλές φορές περιέχουν και μια περίληψη του ίδιου του κειμένου. Επομένως, η αναζήτηση γίνεται με βάση πεδία-κλειδιά που βρίσκονται σε ειδικά αρχεία, τα οποία και ονομάζονται μετα-δεδομένα και περιγράφουν επακριβώς το περιεχόμενο των ψηφιακών αντικειμένων μιας συλλογής. Ο χρήστης, επομένως, αρκεί να δώσει τιμές σε ορισμένα από αυτά τα πεδία για να εντοπίσει με ακρίβεια το αντικείμενο που αναζητά.

Τα μετα-δεδομένα αρκετές φορές αναφέρονται και ως *‘δεδομένα για τα δεδομένα’*. Όμως, ένας τέτοιος όρος είναι κανονικά αποδεκτός να σημαίνει: δομημένα δεδομένα, σχετικά με αντικείμενα τα οποία μπορεί να χρησιμοποιηθούν για την υποστήριξη μεγάλου εύρους από λειτουργίες. Οι πιο συνηθισμένες λειτουργίες είναι η περιγραφή, η ανάκληση, η διαχείριση, καθώς και η μακροπρόθεσμη διατήρηση πληροφοριακών αντικειμένων. Για παράδειγμα, τα μετα-δεδομένα μπορούν να χρησιμοποιηθούν για τη διοίκηση και διαχείριση αντικειμένων μέσα από την καταγραφή πληροφορίας. Η πληροφορία αυτή αφορά την τοποθεσία και την ανάκλησή των εν λόγω αντικειμένων. Επιπροσθέτως, τα μετα-δεδομένα μπορούν να χρησιμοποιηθούν για την καταγραφή οποιουδήποτε πνευματικού δικαιώματος σχετικά με πληροφοριακά αντικείμενα, καθώς επίσης και για την διαχείριση της πρόσβασης του χρήστη σε αυτά τα αντικείμενα. Άλλα μετα-δεδομένα μπορεί να είναι τεχνικά στη φύση τους, όπως αυτά που σχετίζονται είτε με συγκεκριμένα περιβάλλοντα υλικού ή λογισμικού, είτε που καταγράφουν παραμέτρους ψηφιοποίησης. Η δημιουργία και η σωστή συντήρηση των μετα-δεδομένων μπορεί να θεωρηθεί ως ένας σημαντικός παράγοντας τόσο για τη μακροπρόθεσμη διαχείριση των ψηφιακών αντικειμένων, όσο και για τη προστασία του περιεχομένου και της αυθεντικότητας των αντικειμένων.

Η πρώτη χρήση των μετα-δεδομένων προήλθε από περιεχόμενα που σχετίζονταν με ψηφιακή πληροφορία. Ωστόσο, η γενική αντίληψη του όρου έχει διευρυνθεί έτσι ώστε να περιέχει οποιοδήποτε είδος τυποποιημένης και περιγραφικής πληροφορίας για αντικείμενα, ακόμη και μη ψηφιακά. Επί παραδείγματι, οι κατάλογοι βιβλιοθήκης, οι αφαιρετικές και ευρετηριακές υπηρεσίες, η βοήθεια στην αναζήτηση αρχείου μπορούν να θεωρηθούν ότι περιέχουν μετα-δεδομένα. Το γεγονός αυτό επιφέρει διπλό όφελος. Πρώτον, επιτρέπει στους βιβλιοθηκάρους, τους αρχειοφύλακες και τους ειδικούς μουσειακών αρχείων να συνεργαστούν κατά μήκος των επαγγελματικών τους συνόρων. Δεύτερον, καθιστά εφικτή την επικοινωνία ανάμεσα στα επαγγέλματα της πολιτιστικής κληρονομιάς και σε εκείνους τους τομείς που παρουσιάζουν ενδιαφέρον για μετα-δεδομένα, όπως τους παραγωγούς λογισμικού, τους παραγωγούς ψηφιακού εκπαιδευτικού περιεχομένου, τη βιομηχανία εγγραφής, τις τηλεοπτικές εταιρείες, και γενικότερα τους ενδιαφερόμενους για γεωγραφική και δορυφορικά-βασισμένη πληροφορία.

2.5.1. Πρότυπα Μετα-δεδομένων

Ως αποτέλεσμα της εκδήλωσης ενδιαφέροντος από ένα μεγάλο και ποικιλόμορφο πλήθος χρηστών για τα μετα-δεδομένα, αποτελεί ο τεράστιος όγκος

από πρότυπα και φόρμες μετα-δεδομένων, που είτε ήδη έχουν δημιουργηθεί είτε συντελούν υπό κατασκευή. Με τον όρο *πρότυπο μετα-δεδομένων* εννοούμε ένα ελεγχόμενο λεξικό για την περιγραφή σχετικών πληροφοριακών αντικειμένων. Το πρότυπο αναπτύσσεται με τη συγκρότηση μιας έγκυρης ομοφωνίας για τη σημασιολογία, τη δομή και τη σύνταξη των μετα-δεδομένων. Μια έγκυρη ομοφωνία επιτυγχάνεται με τη βοήθεια πολιτικών μέσων και είναι σημαντικό αυτή η ομοφωνία να εξαπλωθεί και να διερευνηθεί. Ένα παράδειγμα προτύπου μετα-δεδομένων είναι η φόρμα MARC [15], που έχει αναπτυχθεί στον κόσμο των βιβλιοθηκών ως μέσο κωδικοποίησης μετα-δεδομένων οριζόμενων σε κανόνες πινακογράφησης. Στην ίδια κοινότητα έχουν επίσης, οριστεί περιγραφικά πρότυπα τα ISBD [16], για τα οποία γίνεται αναφορά στη σειρά *‘International Standard Bibliographic Description’*. Ένα άλλο παράδειγμα είναι η πρωτοβουλία κινήσεων Dublin Core [17], το οποίο ορίζει ένα νέο πρότυπο μετα-δεδομένων με ομώνυμο τίτλο. Σε αυτό αναφέρονται 15 γενικές ιδιότητες, που πρέπει να προσδιορίζουν ένα οποιοδήποτε ψηφιακό αντικείμενο. Τέλος, στην επεξεργασία των επιστημονικών δεδομένων για την τελική διαμόρφωσή τους ως Ψηφιακή Βιβλιοθήκη, καθοριστικό ρόλο αποτέλεσε η ανάπτυξη ιδιαίτερων προτύπων μετα-δεδομένων, όπως το *‘Federal Geographic Data Committee metadata standard’* FGDC [18] για γεω-χωρικά (geospatial) και βιολογικά δεδομένα, καθώς και το UDK [19], το οποίο είναι ένας ευρωπαϊκός περιβαλλοντολογικός κατάλογος δεδομένων.

2.5.2. Οντολογίες

Όπως προκύπτει από την προηγούμενη παράγραφο, τα πρότυπα μετα-δεδομένων καταγράφουν τη γνώση σε μία κοινότητα – τομέα της επιστήμης. Τι γίνεται όμως όταν έχουμε αντικείμενα που ανήκουν σε πολλούς τομείς ή που δεν καλύπτονται σημασιολογικά από ένα πρότυπό που ειδικεύεται στον τομέα τους; Μία πρόχειρη λύση είναι η μίξη προτύπων και η προσθήκη νέων ιδιοτήτων. Τα περισσότερα πρότυπα περιέχουν ένα επίπεδο χώρο αναζήτησης με βάση ορισμένες ιδιότητες. Αν αυξηθεί το πλήθος των ιδιοτήτων, θα αυξηθεί κατά συνέπεια και ο χώρος αναζήτησης. Επομένως, κάθε αναζήτηση θα χρειαστεί περισσότερο χρόνο για να ολοκληρωθεί. Η βέλτιστη λύση, σε αυτό το πρόβλημα δίδεται με τις *οντολογίες*, ένα εργαλείο καταγραφής γνώσης που χρησιμοποιείται κατά κόρον σχεδόν σε όλες τις σύγχρονες μορφές Ψηφιακής Βιβλιοθήκης.

Πολύ συχνά, υπάρχει μια ανάγκη για διαμοιρασμό της σημασίας των όρων σε ένα τομέα. Η απόκτηση αυτής της κοινής αντίληψης γίνεται με τη συμφωνία σε ένα κατάλληλο τρόπο σύλληψης του τομέα. Οι οντολογίες αντιπροσωπεύουν ένα συστατικό κλειδί τόσο στη διαχείριση γνώσης, όσο και στα συστήματα βασιζόμενα στο περιεχόμενο (content-based). Το να σχεδιάσεις μια οντολογία, τελικά, σημαίνει να προσδιορίσεις το σύνολο των σημασιολογικών κατηγοριών που αντικατοπτρίζουν την ιδιαίτερη εννοιολογική οργάνωση της επικράτειας-πλαισίου της πληροφορίας, σύμφωνα με την οποία το σύστημα πρέπει να λειτουργήσει. Με τον προσδιορισμό αυτό γίνεται εφικτή η βελτιστοποίηση της ποσότητας και της ποιότητας της εν λόγω πληροφορίας.

Σύμφωνα με τον κλασικό ορισμό, η οντολογία είναι ένας τυπικός και σαφής προσδιορισμός μιας κοινής αντίληψης. Επομένως, η οντολογία είναι ένα σύνολο από έννοιες – όπως πράγματα, γεγονότα και σχέσεις – που προσδιορίζεται με τέτοιο τρόπο ώστε να δημιουργήσει ένα προσυμφωνημένο λεξιλόγιο για την ανταλλαγή πληροφορίας.

Μία οντολογία είναι κάτι παραπάνω από μία ταξινόμια ή μια διαβάθμιση-κατάταξη όρων. Αν και η ταξινόμια συνεισφέρει στην σημασιολογία ενός όρου, οι οντολογίες περιέχουν πιο πλούσιες σχέσεις μεταξύ όρων. Αυτού του είδους οι σχέσεις είναι που καθιστούν εφικτή την έκφραση της γνώσης σε έναν τομέα.

Η επιλογή μιας καλής οντολογίας παίζει αποφασιστικό ρόλο σε πολλαπλές διαδικασίες στην οργάνωση και τη δόμηση της γνώσης. Επίσης, η οντολογία αποτελεί ένα δομημένο σύστημα από κατηγορίες ή σημασιολογικούς τύπους. Το σύστημα αυτό δομείται με τέτοιο τρόπο ώστε η γνώση για ένα συγκεκριμένο τομέα να μπορεί να οργανωθεί με την κατηγοριοποίηση των οντοτήτων που υπάρχουν τηρώντας πιστά τους τύπους της οντολογίας. Επομένως, η δύναμη της αναπαράστασης μιας οντολογίας εξαρτάται από το εάν η αρχιτεκτονική του συστήματος των τύπων μπορεί να εκφράσει την οργανωτική δομή της γνώσης του συγκεκριμένου τομέα.

Εκτός από τις δυνατότητες πλοήγησης που προσφέρει, μια οντολογία παρέχει κοινή σημασιολογία, που μπορεί να χρησιμοποιηθεί για τη βελτίωση της επικοινωνίας μεταξύ ανθρώπων ή υπολογιστών. Γενικά, οι ωφέλειες μιας οντολογίας μπορούν να ομαδοποιηθούν στις επόμενες τρεις περιοχές, σύμφωνα βέβαια με το ρόλο τους:

- i. η διευκόλυνση της επικοινωνίας μεταξύ των ανθρώπων
- ii. η επίτευξη δια-λειτουργικότητας μεταξύ υπολογιστικών συστημάτων
- iii. η βελτίωση της διαδικασίας και της ποιότητας των εφαρμοσμένων υπολογιστικών συστημάτων.

Επίσης, οι ωφέλειες από τη χρήση των οντολογιών μπορούν να χωριστούν σε πέντε κατηγορίες-κλάσεις :

- **Χρήσιμες ερωτήσεις.** Μια οντολογία χρησιμοποιείται για την αναζήτηση αντικειμένων σε μια αποθήκη από μετα-δεδομένα. Το κύριο κέρδος από αυτή την προσέγγιση είναι η πιο γρήγορη και έξυπνη αναζήτηση για σχετικά πληροφοριακά αντικείμενα, που οδηγεί σε μια πιο αποτελεσματική χρήση και ανα-χρησιμοποίηση των αντικειμένων αυτών.
- **Διαμοιρασμός γνώσης.** Γενικά, μια οντολογία βοηθάει στην ολοκλήρωση πολλών αλληλοεπικαλυπτόμενων κομματιών πληροφορίας. Ιδανικά, οι άνθρωποι θα συνεισφέρουν σε μια κοινή, παγκόσμια και καλά οργανωμένη γνωστική βάση της πληροφορίας. Φυσικά, αυτό χρειάζεται πολύ προσπάθεια και στον τεχνικό και στον θεσμικό κόσμο.
- **Επεκτασιμότητα.** Η οντολογία μπορεί να πλαισιωθεί με νέες κλάσεις σε κάθε κλάδο και σε κάθε επίπεδο. Όμως, η οντολογία πρέπει να σχεδιαστεί καλά ώστε να είναι επαρκώς γενική.
- **Επαναχρησιμότητα.** Η οντολογία είναι η βάση για μια τυπική κωδικοποίηση των σημαντικών οντοτήτων, των ιδιοτήτων, των διεργασιών και των αλληλοσυσχετίσεων των παραπάνω στον τομέα του ενδιαφέροντος. Ένα κοινό ιδανικό για κάθε οντολογία είναι να αποτελεί (επανα)χρήσιμο και διαμοιραζόμενο μέρος ενός συστήματος λογισμικού.
- **Προσδιορισμός δικαιωμάτων.** Τα συστήματα, που βασίζονται σε πράκτορες, χρειάζονται μετα-δεδομένα, τα οποία πρέπει να στηρίζονται σε οντολογίες, προκειμένου να υπολογίσουν αφενός, τις άδειες που απαιτούνται για τη παροχή διαφόρων υπηρεσιών και αφετέρου, την πληρωμή των κατάλληλων δασμών πνευματικής ιδιοκτησίας.

Συμπερασματικά λοιπόν, μπορούμε να θεωρήσουμε ότι οι οντολογίες στοχεύουν αρχικά, στο να εγκαθιδρύσουν ένα γενικά αποδεκτό μοντέλο καταγραφής της γνώσης ενός τομέα συστημάτων, το οποίο θα χαρακτηρίζεται από σαφήνεια και συνέπεια. Ακολουθώντας, στοχεύουν στην ολοκλήρωση μιας κλίμακας από προοπτικές, καθώς και στη διαλειτουργικότητα των υπολογιστικών συστημάτων του τομέα. Τέλος, συνεισφέρουν στη δημιουργία γραφικών εργαλείων, που παρέχουν πλοήγηση στα μετα-δεδομένα.

2.6. Μερικές Σημαντικές Ψηφιακές Βιβλιοθήκες

2.6.1. Stanford Digital Library

Στο Πανεπιστήμιο Stanford [20] έχουν επικεντρώσει την έρευνά τους σε τέσσερις περιοχές. Στην πρώτη περιοχή ασχολούνται με την ολοκλήρωση ετερογενών πληροφοριών και υπηρεσιών. Στην δεύτερη ασχολούνται με την δημιουργία δυναμικών μηχανισμών φιλτραρίσματος που θα βελτιώνουν αισθητά την ποιότητα της επιστρεφόμενης από αναζήτηση πληροφορίας. Ο τρίτος τομέας ασχολείται με τη διάθεση διεπαφών χρήσης και εργαλείων βιβλιοθήκης που θα εκτελούνται επαρκώς σε φορητές συσκευές. Ο τελευταίος τομέας ασχολείται με τη δημιουργία μιας συμπαγούς οικονομικής υποδομής, που αφενός θα ενθαρρύνει τους προμηθευτές να διαθέτουν πληροφορίες και αφετέρου θα εξασφαλίζει ιδιωτικότητα για τους χρήστες.

Το σύστημα, όπως σχεδιάζεται στο υψηλότερο επίπεδο, ονομάζεται 'Infobus' και φιλοδοξεί να παρέχει ένα ευρέως αποδεκτό τρόπο πρόσβασης σε μια ποικιλία από υπηρεσίες και σε πληροφοριακά αντικείμενα.

2.6.2. UC Berkley Digital Library

Το 'UC Berkeley Digital Library' project [21] ασχολείται με την ανάπτυξη εργαλείων και τεχνολογιών για την υποστήριξη υψηλά βελτιωμένων μοντέλων του 'κύκλου ζωής της λόγιας πληροφορίας' (scholarly information life cycle). Στόχος του ερευνητικού έργου είναι η διευκόλυνση της μετακίνησης από το τρέχων κεντρικοποιημένο και διακριτής δημοσίευσης μοντέλο (discrete publishing model) σε ένα καταναμημένο και συνεχές αυτο-δημοσιευόμενο μοντέλο, αλλά και η ταυτόχρονη διατήρηση των καλύτερων προοπτικών του τρέχοντος μοντέλου, όπως αυτή της ισότιμης όψης (peer view). Οι τεχνολογίες-περιοχές έρευνας αυτού του Ερευνητικού έργου (project) είναι οι εξής :

- Ανάκτηση εικόνων με βάση το περιεχόμενό τους.
- Νέα μοντέλα κειμένων όπως το Web-based GIS και το TilePic.
- Ανάλυση εικόνας που είναι σε μορφή εγγράφου.
- Δυνατές Αναφορές (Robust References).
- Η καταναμημένη αναζήτηση.
- Επεξεργασία φυσικής γλώσσας (Natural Language Processing) για πρόσβαση σε πληροφορία.

2.6.3. Ariadne

Το ‘*Ariadne*’ project [22] του Ινστιτούτου Επιστημών Πληροφορίας του Πανεπιστημίου της Νότιας Καλιφόρνια αφοσιώνεται στο πρόβλημα της ανάκτησης και ενοποίησης δεδομένων από πολλαπλά μέρη (sites), αξιοποιώντας με αυτό το τρόπο τον πλούτο από δεδομένα που είναι διαθέσιμα προς ανάγνωση στο διαδίκτυο. Στα πλαίσια αυτής της ερευνητικής μελέτης έχουν αναπτυχθεί η τεχνολογία και τα εργαλεία για την ταχεία κατασκευή ‘έξυπνων πρακτόρων’ (intelligent agents) που θα εξάγουν, επερωτούν και θα ενοποιούν δεδομένα που βρίσκονται σε διαφορετικές πηγές στο διαδίκτυο. Επίσης, έχουν αναπτυχθεί εργαλεία για την κατασκευή wrappers, που κάνουν τις πηγές διαδικτύου να μοιάζουν με βάσεις δεδομένων, και τεχνολογία μεσάζοντος (mediator), που απαντά σε επερωτήσεις με τη βοήθεια-χρησιμοποίηση των παραπάνω πηγών. Το τελικό σύστημα προσδίδει στην κατασκευή νέων εφαρμογών πρακτόρων από ήδη υπάρχουσες πηγές του διαδικτύου, γρήγορο και φθινό ρυθμό ανάπτυξης. Η προσέγγιση του μεσάζοντα και του wrapper διευκολύνει, τόσο τη διατήρηση αυτών των εφαρμογών, όσο και την ενσωμάτωση νέων πηγών καθώς αυτές γίνονται διαθέσιμες.

2.6.4. Alexandria Digital Library

Το ‘*Alexandria Digital Library*’ project [23] υλοποιεί μια Ψηφιακή Βιβλιοθήκη, η οποία περιέχει συλλογές από γεωγραφικά αναφερόμενα υλικά και υπηρεσίες πρόσβασης των παραπάνω συλλογών. Το ‘*Alexandria Digital Earth Prototype*’ (ADEPT) είναι ένα μεγάλο εύρους επακόλουθο της αλεξανδρινής ψηφιακής βιβλιοθήκης. Το ADEPT στοχεύει στη χρήση μιας μεταφοράς (αλληγορίας) της ψηφιακής γης για να οργανώσει, χρησιμοποιήσει και παρουσιάσει πληροφορία τοπικής και χρονικής ανάλυσης. Τα ψηφιακά περιβάλλοντα βασιζόμενα στη προαναφερόμενη αλληγορία ονομάζονται Iscapes (Information Landscapes). Οι δραστηριότητες του ADEPT εστιάζονται στα εξής σημεία :

- Δημιουργία γεω-χωρικής (geospatial) πληροφορίας και συλλογών μετα-πληροφορίας.
- Κατασκευή λειτουργικών υπηρεσιών για την ανακάλυψη ετερογενών και κατανομημένων συλλογών. Εξόρυξη μετα-πληροφοριών και διευκόλυνση μιας ενοποιημένης χρήσης τους. Οργάνωση αυτών των πηγών σε Iscapes, τα οποία είναι προσαρμοσμένα στις ανάγκες των εφαρμογών. Συνεργατική χρήση και οπτικοποίηση των Iscapes.
- Εφαρμογή και αποτίμηση των υπηρεσιών του ADEPT στη προπτυχιακή γνώση. Ανάπτυξη κλιμακωτών, επαρκών και ασφαλών συστημάτων.

2.6.5. Master Environmental Library (MEL)

Τα δεδομένα φυσικού περιβάλλοντος βρίσκονται σε διάφορα sites του διαδικτύου. Αυτά τα δεδομένα περιέχουν στατικά γνωρίσματα και δυναμικά, χαρακτηριστικά και φαινόμενα του εδάφους, του ωκεανού, του αέρα και του χώρου καθώς και επιλεγμένα μόνιμα ή ημι-μόνιμα γεω-χωρικά χαρακτηριστικά, όπως οι δρόμοι, οι γέφυρες, τα κτίρια, οι πόλεις, τα λιμάνια και τα αεροδρόμια.

Ο σκοπός του MEL [24] είναι να παρέχει απευθείας χρονική πρόσβαση σε πληροφορίες, δεδομένα και προϊόντα φυσικού περιβάλλοντος, όπου και αν

βρίσκονται αυτά. Από αυτά άλλα μπορεί να είναι μη γεω-χωρικά δεδομένα, όπως μοντέλα, αλγόριθμοι και έγγραφα και άλλα να είναι βασικά περιβαλλοντολογικά δεδομένα. Ο τελικός σκοπός του MEL είναι να γίνει ένα ‘Μαγαζί Περιβάλλοντος Μιας Στάσης’, όπου κάθε χρήστης μπορεί να έχει απομακρυσμένη πρόσβαση και να κάνει αίτηση για περιβαλλοντολογικές πηγές, προκειμένου να ικανοποιήσει τις ποικίλες και συνεχώς προστιθέμενες ανάγκες του.

2.6.6. Earth Observing System Data and Information System (EOSDIS)

Το Σύστημα Παρατήρησης της Γης Δεδομένων και Πληροφοριών (EOSDIS) [25] έχει αναπτυχθεί από τη NASA [26], ώστε να γίνουν εύκολα διαθέσιμα στο ευρύ κοινό δεδομένα που έχουν σχέση με τη γη και τη παρατήρησή της. Τα δεδομένα αυτά σχετίζονται με παλαιότερα δεδομένα, προερχόμενα από παρατηρήσεις της NASA. Συγκεκριμένα, το EOSDIS είναι υπεύθυνο για τον έλεγχο δορυφόρων και εργαλείων μετρήσεων. Επίσης, συντελεί στη διαθεσιμότητα των δεδομένων που προκύπτουν, όπως είναι τα μοντέλα κλίματος που παράγονται από προσομοιώσεις.

Το EOSDIS περιέχει ένα ευρύ φάσμα πληροφοριών και δεδομένων και μπορεί να υποστηρίξει τόσο την αμερικανική όσο και την παγκόσμια επιστημονική κοινότητα. Επιπλέον, παρέχει πληθώρα υπηρεσιών, οι οποίες και απευθύνονται και σε διαφορετικούς τύπους χρηστών.

Η επικοινωνία του χρήστη με το σύστημα γίνεται με δύο τρόπους: με την πρόσβαση στο διαδίκτυο ή με την χρησιμοποίηση του προγράμματος του EOSDIS για άμεση πρόσβαση στις βάσεις δεδομένων.

2.6.7. DODS

Το DODS [27] είναι ένα σύστημα που έχει άμεση σχέση με καταναμημένα συστήματα δεδομένων. Ουσιαστικά πρόκειται για μια βιβλιοθήκη, που χρησιμοποιείται για την υλοποίηση ενός απομακρυσμένου συστήματος αρχείων από το πακέτο της MathLab [28].

Το DODS είναι ένα πλαίσιο εργασίας, το οποίο απλουστεύει όλες τις πτυχές της διαδικτύωσης επιστημονικών δεδομένων. Παρέχει εργαλεία για να κάνει διαθέσιμα τοπικά δεδομένα σε απομακρυσμένες τοποθεσίες ανεξαρτήτου της μορφής της τοπικής αποθήκευσης. Επίσης, παρέχει εργαλεία για την μετατροπή υπάρχουσων εφαρμογών σε πελάτες DODS. Με αυτό τον τρόπο, επιτρέπει την απομακρυσμένη πρόσβαση σε δεδομένα που βρίσκονται σε υπηρετητές DODS.

Το σύστημα αυτό χρησιμοποιείται για την διευκόλυνση επιστημονικών προσομοιώσεων και υπολογισμών, στους οποίους χρησιμοποιούνται τεράστιες ποσότητες δεδομένων. Ο χρήστης αντί να μεταφέρει (ύστερα από ένα μεγάλο χρονικό διάστημα) ένα τεράστιο σύνολο δεδομένων προκειμένου να το αποθηκεύσει στον υπολογιστή του και να εκτελέσει μια προσομοίωση ή ένα πρόγραμμα, ορίζει πού βρίσκεται το αρχείο αυτό και το σύστημα αναλαμβάνει να μεταφέρει τα απαραίτητα κομμάτια για τον υπολογισμό. Κατά συνέπεια, εξοικονομεί χρόνο και χώρο, ενώ παράλληλα δεν επαναλαμβάνει την ίδια διαδικασία μεταφοράς του συνόλου δεδομένων, στις περιπτώσεις που αυτό αλλάζει ή το διαγράφει από τον σκληρό δίσκο.

3. Ροές Εργασίας

3.1. Ορισμός Ροής Εργασίας και Μοντέλου Ροής Εργασίας

Ο πιο δημοφιλής ορισμός για το τι είναι ακριβώς μια ροή εργασίας (workflow) έχει δοθεί από το Workflow Management Coalition [29,31]. Ροή εργασίας ορίζεται να είναι ο αυτοματισμός πλήρης ή μερικός, ενός επιχειρησιακού μοντέλου κατά τον οποίο έγγραφα, πληροφορίες ή έργα (tasks) μεταφέρονται από τον ένα συμμετέχοντα στον επόμενο για εκτέλεση, σύμφωνα με ένα σύνολο κανόνων. Μια επιχειρησιακή διαδικασία ορίζεται να είναι ένα σύνολο από μια ή περισσότερες διαδικασίες ή ενέργειες, οι οποίες συλλογικά πραγματοποιούν μια επιχειρησιακή επιδίωξη ή ένα στόχο πολιτικής-τακτικής, κανονικά στο πλαίσιο μιας οργανωτικής δομής οριζόμενης από λειτουργικούς ρόλους και σχέσεις.

Το μοντέλο εργασίας περιγράφει τη δομή μιας επιχειρησιακής διαδικασίας στον πραγματικό κόσμο. Ορίζει όλα τα δυνατά μονοπάτια μέσα από την επιχειρησιακή διαδικασία, περιλαμβανομένου των κανόνων που ορίζουν ποια μονοπάτια πρέπει να ακολουθηθούν και ποιές ενέργειες πρέπει να γίνουν. Αυτό το μοντέλο είναι περίγραμμα από το οποίο κάθε διαδικασία αποκτά περιπτώσεις. Ο όρος 'περίπτωση διαδικασίας' ταιριάζει τέλεια όταν αναφερόμαστε στις παραπάνω περιπτώσεις. Οι διαδικασίες δεν τρέχουν αναγκαστικά σε ένα υπολογιστή. Ένας μεγάλος αριθμός από επιχειρησιακές διαδικασίες μπορεί να εκτελεστεί χωρίς να απαιτείται κανένα απολύτως βήμα εκτέλεσης από κάποιον υπολογιστή. Επομένως, οι επιχειρησιακές διαδικασίες μπορεί να αποτελούνται από κομμάτια, που εκτελούνται από υπολογιστή καθώς και από κομμάτια που δεν προκύπτουν από υπολογιστή. Τα μέρη, που εκτελούνται μόνο με την υποστήριξη ηλεκτρονικού υπολογιστή ονομάζονται 'μοντέλο ροής εργασίας'.

Ένα μοντέλο ροής εργασίας μπορεί να είναι ένα μικρό μέρος ενός τεράστιου μοντέλου διαδικασίας ή μπορεί και να περικλείει ολόκληρο το μοντέλο διαδικασίας. Τα προγράμματα υπολογιστών, που εκτελούν τη ροή εργασίας μπορεί να αποτελούν ένα γενικού σκοπού σύστημα διαχείρισης ροών εργασίας ή ειδικές εφαρμογές που υλοποιούν το μοντέλο διαδικασίας. Το μοντέλο ροής εργασίας είναι ένα περίγραμμα για τη δημιουργία ροών εργασίας με την ίδια έννοια, που το μοντέλο διαδικασίας αποτελεί ένα περίγραμμα για τη δημιουργία διαδικασιών.

Εν κατακλείδι, ξεχωρίζουμε τις έννοιες της ροής εργασίας και της διαδικασίας. Ορίζουμε ως διαδικασία να είναι ο αυτοματισμός μιας επιχειρησιακής διαδικασίας στον πραγματικό κόσμο και ως ροή εργασίας τον αντίστοιχο αυτοματισμό στον κόσμο των υπολογιστών.

3.2. Διαστάσεις μιας Ροής Εργασίας

Οι επιχειρησιακές διαδικασίες, άρα και οι ροές εργασίας, έχουν τρεις ανεξάρτητες διαστάσεις.

Η πρώτη διάσταση αντιπροσωπεύει τη 'λογική της διαδικασίας'. Η λογική της διαδικασίας περιγράφει τις ενέργειες, που πρέπει να εκτελεστούν και την ακολουθία εκτέλεσής τους. Μια ενέργεια μπορεί να είναι χειρωνακτική (μη υποστηριζόμενη από υπολογιστή) ή αυτοματοποιημένη (υποστηριζόμενη από υπολογιστή). Αν είναι

αυτοματοποιημένη, μπορεί να είναι ένα πρόγραμμα ή μια άλλη ανάλογη διαδικασία. Η ροή εκτέλεσης (ελέγχου) από μια ενέργεια σε μια άλλη μπορεί να είναι ακολουθιακή (σειριακή) ή παράλληλη. Αν είναι ακολουθιακή, τότε μια ενέργεια εκτελείται μετά από την ολοκλήρωση μιας άλλης. Αν είναι παράλληλη, τότε δύο ή περισσότερες ενέργειες εκτελούνται ταυτόχρονα, μειώνοντας με αυτό τον τρόπο το χρόνο εκτέλεσης της επιχειρησιακής διαδικασίας. Μερικές φορές, η ροή εκτέλεσης μπορεί να είναι υπό όρους. Δηλαδή εξαρτώμενοι από τα αποτελέσματα μιας ενέργειας, μπορούμε είτε να συνεχίσουμε μπροστά στην εκτέλεση μιας νέας ενέργειας, είτε να πάμε πίσω και να ξαναεκτελέσουμε μια προηγούμενη ενέργεια.

Η δεύτερη διάσταση είναι η οργανωτική. Περιγράφει την οργανωτική δομή μιας εταιρείας σε σχέση με τμήματα, ρόλους και άτομα. Παρέχει, δηλαδή, την πληροφορία, που απαιτείται για το ποιός θα εκτελέσει ποιά ενέργεια. Αν αυτή η ενέργεια δεν αλληλεπιδρά με ένα χρήστη, τότε το σύστημα διαχείρισης ροών εργασίας αναλαμβάνει την εκτέλεση αυτής της, εκ μέρους του επιλεγμένου χρήστη.

Η τρίτη διάσταση μιας ροής εργασίας είναι η διάσταση της υποδομής της τεχνολογίας της πληροφορίας (Information Technology - IT infrastructure). Περιγράφει ποιοι IT πόροι, όπως τα προγράμματα που εκτελούν μια ιδιαίτερη ενέργεια, χρειάζονται για τη διεκπεραίωση μιας επιχειρησιακής εργασίας.

Συμπερασματικά, η ιστορία εκτέλεσης μιας διαδικασίας είναι μια ακολουθία από τριάδες (ενέργεια, χρήστης, IT πόρος) ή με άλλα λόγια μια σειρά από σημεία στον τρισδιάστατο χώρο των ροών εργασίας (τι, ποιος, με τι).

3.3. Κατηγορίες Ροών Εργασίας

Χρησιμοποιώντας δύο βασικά χαρακτηριστικά των ροών εργασίας, την επιχειρησιακή αξία και την επανάληψη, μπορούμε να διακρίνουμε τέσσερις τύπους ροών εργασίας: συνεργατική, επί τούτου, διαχειριστική και παραγωγική ροή εργασίας. Η ιδιότητα της επιχειρησιακής αξίας ορίζει την σημασία μιας ροής εργασίας στην επιχείρηση μιας εταιρείας. Μια διαδικασία υψηλής επιχειρησιακής αξίας είναι ένα ουσιώδες τμήμα (πυρήνας) για μια εταιρεία. Από την άλλη πλευρά, η ιδιότητα της επανάληψης μετράει πόσο συχνά μια ιδιαίτερη διαδικασία εκτελείται με τον ίδιο τρόπο. Είναι ένας δείκτης για το εάν μια διαδικασία αξίζει να σχεδιαστεί.

Οι συνεργατικές ροές εργασίας χαρακτηρίζονται από μια υψηλή επιχειρησιακή αξία, αλλά εκτελούνται ελάχιστες φορές. Η θεμελιώδης διαδικασία τους είναι σχετικά πολύπλοκη και δημιουργείται ειδικά για ένα ιδιαίτερο έργο, συχνά με τη προσαρμογή σε ένα δεδομένο πλάνο εργασίας.

Οι επί τούτου ροές εργασίας εκδηλώνουν μια μικρή επιχειρησιακή αξία και ένα χαμηλό ρυθμό επανάληψης. Γενικά, αυτές οι ροές εργασίας είτε δεν έχουν μια προκαθορισμένη δομή και το επόμενο βήμα στη διαδικασία προσδιορίζεται από κάθε χρήστη που εμπλέκεται σε αυτή τη διαδικασία, είτε κάθε επιχειρησιακή διαδικασία κατασκευάζεται ξεχωριστά κάθε φορά που υπάρχει ανάγκη εκτέλεσης μιας σειράς από ενέργειες.

Οι διαχειριστικές ροές εργασίας παρουσιάζουν, επίσης, χαμηλή επιχειρησιακή αξία. Παρόλα αυτά, ο παράγοντας επανάληψης είναι αρκετά υψηλός. Αυτού του είδους οι ροές διεκπεραιώνονται διαχειριστικά, όπως γίνεται με την επεξεργασία των λογαριασμών εξόδων ή των καταναλωτικών παραγγελιών.

Οι παραγωγικές ροές εργασίας χαρακτηρίζονται από την κατοχή υψηλής επιχειρησιακής αξίας και υψηλού παράγοντα επανάληψης. Είναι οι ροές, που υλοποιούν τον πυρήνα της επιχείρησης για μια εταιρεία. Η αποτελεσματικότητά τους

εκτέλεση παρέχει ανταγωνιστικό πλεονέκτημα. Συχνά, οι παραγωγικές ροές υλοποιούνται χωρίς ένα σύστημα διαχείρισης ροών εργασίας, διότι οι αντίστοιχες διαδικασίες λειτουργούν σωστά για αρκετά χρόνια. Και λόγω της σημασίας τους, οι αντίστοιχες εφαρμογές τυπικά υλοποιούνταν σαν προγράμματα παρακολούθησης επεξεργαστικών δΟΣΟΛΗΨΙΩΝ. Η κλιμάκωση, η διαθεσιμότητα και η ευρωστία αυτών των περιβαλλόντων είναι, επομένως, μια αναγκαία προϋπόθεση για τις μηχανές ροών εργασίας που τρέχουν παραγωγικές ροές. Τα συστήματα διαχείρισης ροών εργασίας λέγονται παραγωγικά.

Ένα, επίσης, βασικό χαρακτηριστικό των ροών εργασίας αποτελεί ο βαθμός αυτοματισμού της ροής. Η μέτρηση αυτή δείχνει την ανεξαρτησία μιας ροής εργασίας από την ανθρώπινη παρέμβαση, δηλαδή δείχνει το πότε οι ενέργειες μιας ροής εργασίας εκτελούνται ή από ανθρώπους ή από το σύστημα. Μια υψηλά αυτοματοποιημένη διαδικασία είναι εντατικού υπολογισμού ενέργεια και τυπικά ολοκληρώνει ετερογενή και αυτόνομα συστήματα εφαρμογών.

Μια διαδικασία με υψηλό βαθμό αυτοματισμού δεν σημαίνει αναγκαστικά ότι δεν έχει αλληλεπιδράσεις με ανθρώπους. Τυπικά, μια υψηλά αυτοματοποιημένη ροή εργασίας 'τεμαχίζεται' σε ακολουθίες από ενέργειες, που κάθε μια από αυτές εκτελείται από ένα χρήστη, αναμειγμένες με ακολουθίες από ενέργειες αυτοματοποιημένης εκτέλεσης.

3.4. Ωφέλειες από Ροές Εργασίας

Οι πιο πολλοί οργανισμοί, που χρησιμοποιούν ροές εργασίας, παρακινούνται από τους εξής παράγοντες :

1. Βελτιωμένη αποδοτικότητα, που οδηγεί σε χαμηλότερο κόστος ή σε υψηλότερη ικανότητα φόρτου εργασίας (workload).
2. Βελτιωμένος έλεγχος, που παράγεται από την τυποποίηση των διαδικασιών.
3. Βελτιωμένη ικανότητα διαχείρισης διαδικασιών. Τα προβλήματα απόδοσης γίνονται ρητά και κατανοητά.

Για πολλούς οργανισμούς, ο κυριότερος οδηγός στην απόφαση χρήσης της τεχνολογίας των ροών εργασίας είναι :

- Η πίεση για την μείωση κόστους.
- Η πίεση για αυξανόμενη ποιότητα ή ικανότητα.

Πρέπει να σημειωθεί ότι η βελτίωση του ισολογισμού δεν είναι μια αναπόφευκτη επίπτωση από την εισαγωγή των ροών εργασίας. Οι βελτιώσεις στις διαδικασίες συμβαίνουν ως αποτέλεσμα λιγότερο αφηρημένων αλλαγών όπως :

- Τυποποίηση διαδικασιών.
- Διαχείριση διαδικασιών.
- Αποτελεσματική παράδοση των βασιζόμενων σε πληροφορία έργων στους εργάτες.
- Ρητή εστίαση στη σχεδίαση των διαδικασιών.

Λαμβάνοντας υπόψιν εάν οι ροές εργασίας μπορούν να βελτιώσουν τις επιχειρησιακές διαδικασίες, είναι σημαντικό να εστιάσουμε την προσοχή μας στις ανωτέρω αλλαγές, διότι συνοψίζουν τι πραγματικά εκτελεί μια ροή

εργασίας. Η βελτίωση της διαδικασίας, αν συμβεί, θα λάβει χώρα μέσω αυτών των χειροπιαστών επιδράσεων-αλλαγών.

Τα συστήματα ροών εργασίας χρησιμοποιούνται ολοένα και περισσότερο από διαφορετικά είδη οργανισμών με ποικίλους τρόπους. Για παράδειγμα:

- Οι εταιρείες ασφάλισης χρησιμοποιούν ροές εργασίας, για να επιταχύνουν τη διαχείριση απαιτήσεων, διατηρώντας ταυτόχρονα τη διατήρηση ελέγχου αυτών των απαιτήσεων.
- Τα τμήματα μιας κυβέρνησης χρησιμοποιούν ροές εργασίας, για να βελτιώσουν την αποδοτικότητα των αποφάσεων για την πληρωμή επιδομάτων κοινωνικής ασφάλισης.
- Οργανισμοί όλων των ειδών χρησιμοποιούν ροές εργασίας, για να βελτιώσουν την αποτελεσματικότητα των λειτουργιών πελατειακής υποστήριξης και επεξεργασίας παραγγελιών.
- Οι ροές εργασίας χρησιμοποιούνται για την υποστήριξη συνηθισμένων εσωτερικών διαχειριστικών διαδικασιών, όπως η αναφορά προσωπικού και η διαχείριση αξίωσης δαπανών (expense-claims).
- Οι ροές εργασίας χρησιμοποιούνται, για να ωθήσουν τους υπαλλήλους ενός οργανισμού, να κατασκευάσουν τις δικές τους διαδικασίες ροών εργασίας, προκειμένου να αντιμετωπίσουν μεμονωμένα τις δικές τους ειδικές διαδικαστικές ευθύνες.
- Ακόμη και πολύπλοκες διαδικασίες, όπως τα εξαιρετικά μεγάλα projects ανάπτυξης προσωπικού, υποστηρίζονται από τα συστήματα ροών εργασίας.

Ένα είναι σίγουρο. Ότι η αγορά αυξάνει με γοργούς ρυθμούς όσον αφορά τους οργανισμούς, που υιοθετούν την τεχνολογία των ροών εργασίας.

3.5. Συστήματα Διαχείρισης Ροών Εργασίας

Ένα σύστημα διαχείρισης ροών εργασίας (Workflow Management System or WFMS) είναι ένα σύστημα που ορίζει, δημιουργεί και διαχειρίζεται την εκτέλεση των ροών εργασίας μέσω της χρήσης λογισμικού, που τρέχουν μια ή περισσότερες μηχανές ροών εργασίας (workflow engines). Οι τελευταίες μπορούν να ερμηνεύσουν τον ορισμό μιας ροής εργασίας, να αλληλεπιδράσουν με συμμετέχοντες ροών εργασίας (όπως είναι οι χρήστες ή τα προγράμματα) και, όπου είναι αναγκαίο, να βάλουν σε ενέργεια τη χρήση IT εργαλείων και εφαρμογών. Αυτού του είδους τα συστήματα έχουν τη δυνατότητα να χρησιμοποιήσουν μια ευρύτερη ποικιλία από IT και επικοινωνιακών υποδομών και να λειτουργήσουν σε ένα περιβάλλον που εκτείνεται από μικρές ομάδες εργασίας σε παγκόσμιες επιχειρήσεις. Παρόλη αυτήν τη ποικιλία, όλα τα WFM συστήματα επιδεικνύουν συγκεκριμένα όμοια χαρακτηριστικά, τα οποία παρέχουν τη βάση για την ανάπτυξη δυνατοτήτων ολοκλήρωσης και διαλειτουργικότητας μεταξύ διαφορετικών προϊόντων.

Σε ένα πιο υψηλό επίπεδο, όλα τα WFM συστήματα μπορεί να χαρακτηρισθεί ότι παρέχουν υποστήριξη σε τρεις λειτουργικές περιοχές :

1. Λειτουργίες κατασκευής (build-time), οι οποίες εμπλέκονται με τον ορισμό και πιθανά τη μοντελοποίηση της διαδικασίας ροής εργασίας και των συνιστώμενων ενεργειών.
2. Λειτουργίες ελέγχου εκτέλεσης (run-time), οι οποίες εμπλέκονται αφενός με τη διαχείριση των διαδικασιών ροών εργασίας σε ένα λειτουργικό

περιβάλλον και αφετέρου με την τοποθέτηση κατά σειρά των διάφορων ενεργειών, που αποτελούν μέρη των εν λόγω διαδικασιών.

3. Αλληλεπιδράσεις εκτέλεσης με χρήστες και εργαλεία εφαρμογών για την επεξεργασία των διάφορων βημάτων κάθε ενέργειας.

Όλα τα WFM συστήματα θεωρούνται ως ένα ενδιάμεσο επίπεδο μεταξύ του δικτύου και των εφαρμογών, δηλαδή κρίνονται ως middleware. Ακολουθούν μια πρότυπη δομή ως προϊόντα middleware, όπως και τα συστήματα διαχείρισης σχεσιακών βάσεων δεδομένων. Αποτελούνται από ένα σύνολο από συνιστώσες-μέρη που είναι τα εξής: το Meta-model, το Build-time, το Run-time και μια βάση δεδομένων. Αυτές οι συνιστώσες αναλύονται παρακάτω:

- Το Meta-model ορίζει τα κατασκευαστικά υλικά και τις συσχετιζόμενες λειτουργίες που υποστηρίζονται από το σύστημα διαχείρισης ροών εργασίας, όπως είναι η δομή του μοντέλου διαδικασίας και οι λειτουργίες που μπορούν να εκτελεστούν από μια ειδική περίπτωση του μοντέλου διαδικασίας.
- Το Build-time παρέχει τις λειτουργίες ορισμού κατασκευαστικών μερών, οριζόμενων από το χρήστη σε σχέση με το μετα-μοντέλο. Τα κατασκευαστικά μέρη μπορούν να οριστούν ως μοντέλα διαδικασιών, οργανωτικές δομές, και IT όψεις, όπως είναι τα προγράμματα που υλοποιούν τις ενέργειες μιας διαδικασίας και την κατανομή των μερών ενός συστήματος διαχείρισης ροών εργασίας μεταξύ υπηρετητών και δικτύων.
- Το Run-time διεκπεραιώνει λειτουργίες πάνω σε κατασκευαστικά μέρη, που ορίζονται από το χρήστη και βασίζονται στη λειτουργική σημασιολογία του μετα-μοντέλου, όπως είναι η δημιουργία διαδικασιών, η πλοήγηση μέσα σε αυτές και ο έλεγχος τους.
- Η βάση δεδομένων περιέχει όλη την πληροφορία, που διαχειρίζονται τα μέρη του build-time και του run-time. Συνεπώς, περιέχει όχι μόνο τα κατασκευαστικά μέρη, που ορίζονται από τους χρήστες, όπως είναι τα μοντέλα διαδικασιών, αλλά και τις περιπτώσεις αυτών των κατασκευαστικών μερών, όπως είναι οι πραγματικές διαδικασίες.

3.5.1 Build-time

Το build-time κομμάτι-μέρος ενός συστήματος διαχείρισης ροών εργασίας παρέχει τις λειτουργίες και τις δυνατότητες του ορισμού, της δοκιμασίας και της διαχείρισης της πληροφορίας σχετιζόμενης με ροές εργασίας. Το σύνολο αυτό περιλαμβάνει και τις τρεις διαστάσεις μιας ροής εργασίας: το μοντέλο διαδικασίας, την οργανωτική πληροφορία και τα προγράμματα, που χρειάζονται για εκπληρώσουν τις διαδικασίες. Επίσης, βρίσκεται αποθηκευμένο στη βάση δεδομένων του συστήματος διαχείρισης ροών εργασίας.

Γενικά, το build-time μέρος παρέχει τις λειτουργίες του ορισμού, της διοικητικής διαχείρισης και της 'διαχείρισης συστήματος' της πληροφορίας. Τυπικά παραδείγματα αυτού του τύπου της πληροφορίας είναι η πληροφορία σχετιζόμενη με σύνοδο (session), όπως είναι η μέγιστη ποσότητα του χρόνου δουλειάς ενός χρήστη με ένα σύστημα διαχείρισης ροών εργασίας, ή επιχειρησιακά χαρακτηριστικά, όπως είναι οι ενέργειες που το σύστημα διαχείρισης ροών εργασίας πρέπει να κάνει εάν ο χρόνος απόκρισης υπερβεί ένα καθορισμένο όριο.

Συνήθως, το build-time ενός συστήματος διαχείρισης ροών εργασίας εμφανίζεται-εκδηλώνεται με δύο διαφορετικούς τρόπους. Ο πρώτος τρόπος είναι ο

ορισμός της πληροφορίας μιας ροής εργασίας μέσω μιας γραφικής διεπαφής τελικού χρήστη. Ο άλλος τρόπος γίνεται μέσω μιας ιδιόκτητης ή πρότυπης γλώσσας ορισμού ροών εργασίας.

3.5.2 Run-time

Η κύρια επιδίωξη του run-time μέρους ενός συστήματος διαχείρισης ροών εργασίας είναι να καθοδηγήσει εκ των προτέρων διαδικασίες. Το run-time μέρος πλοηγείται μέσω του μοντέλου διαδικασίας και αλληλεπιδρά με χρήστες και εφαρμογές.

Ο τύπος της δράσης ενός συστήματος διαχείρισης ροών εργασίας για κάθε μια από τις ενέργειες εξαρτάται από τη ρύθμιση των παραμέτρων των ενεργειών και τις υλοποιήσεις των αλληλο-συσχετιζόμενων ενεργειών. Οι ενέργειες μπορεί να είναι αυτόματες, χωρίς ο χρήστης να αρχίζει ρητά την υλοποίησή τους, ή στην αντίθετη περίπτωση να είναι χειρωνακτικές. Η υλοποίηση μιας ενέργειας μπορεί να τρέξει στο σταθμό εργασίας (H/Y) ενός χρήστη, στον επεξεργαστή όπου το σύστημα διαχείρισης ροών εργασίας εκτελείται, σε ένα διαφορετικό επεξεργαστή ή σε μια ανάμειξη των παραπάνω αντιπροσωπεύοντας μια εφαρμογή πελάτη-εξυπηρετητή. Μπορεί να τρέξει ασυνόδευτα, δηλαδή χωρίς την αλληλεπίδραση με ένα χρήστη, ή συνοδευτικά στην αντίθετη περίπτωση. Μπορεί να εκτελέσει λειτουργίες βάσης δεδομένων, όπως είναι η ανάκτηση, η αποθήκευση ή η ενημέρωση δεδομένων, πιθανότατα να χρησιμοποιηθεί και από άλλες υλοποιήσεις ενεργειών, ή μπορεί απλά να εμφανίζει ή να συλλέγει πληροφορία, η οποία και μεταφέρεται από το σύστημα διαχείρισης ροών εργασίας από ενέργεια σε ενέργεια.

3.5.3. Μερικά συστήματα διαχείρισης ροών εργασίας

Η τεχνολογία των ροών εργασίας είναι ένας ευρύς ερευνητικός τομέας, που πραγματεύεται τη μοντελοποίηση και τον αυτοματισμό διαδικασιών, οι οποίες περιλαμβάνουν ποικίλα έργα (tasks). Δυστυχώς, οι πιο πολλές υλοποιήσεις μέχρι τώρα είναι κεντροποιημένες, με ένα απλό μέρος υπεύθυνο για την διαρρύθμιση της εκτέλεσης διαδικασίας. Επομένως, υποφέρουν από έλλειψη αξιοπιστίας και δυνατότητα κλιμάκωσης. Ευτυχώς, υπάρχουν και ορισμένα καταναμημένα συστήματα, τα οποία θα αναλύσουμε παρακάτω.

3.5.3.1. Exotica και WIDE

Τα συστήματα Exotica/FMQM [32] και WIDE [33] παρουσιάζουν καταναμημένες αρχιτεκτονικές για το 'ανέβασμα' (enactment) ροών εργασίας βασισμένα σε τεχνολογίες middleware.

Η σχεδίαση του Exotica/FMQM είναι σε διαδοχικά επίπεδα πάνω από ένα εξακολουθητικό (persistent) σύστημα ουράς και υποστηρίζει τη σημασιολογία δοσοληπιών μέσω κλήσεων ενός API για την πρόσβαση στην ουρά. Ο ορισμός ενός μοντέλου διαδικασίας διαιρείται σε μέρη από ένα μεταγλωττιστή και κάθε μέρος κατανέμεται στον κατάλληλο κόμβο. Κάθε κόμβος συγχρονίζεται με τους υπόλοιπους μέσω του δικτύου των εξακολουθητικών ουρών.

Το WIDE είναι μια αρχιτεκτονική για τη διαχείριση ροών εργασίας, στηριζόμενη στη CORBA [34]. Βασίζεται σε ένα σύστημα διαχείρισης ροών εργασίας με υποστήριξη ενεργών κανόνων (active rule). Κάθε μηχανή ροών εργασίας συντηρεί μια βάση δεδομένων από γεγονότα και περιλαμβάνει ένα διαρρυθμιστή (scheduler), ο οποίος ταιριάζει γεγονότα με κανόνες. Οι επιλεγμένοι κανόνες καταγράφονται σε μια λίστα εκτέλεσης (to-execute), η οποία με τη σειρά της 'δημοσκοπείται' (polled) από έναν ερμηνευτή κανόνων. Και οι δύο προσεγγίσεις φαίνονται να εφαρμόζονται στα ιδιωτικά δίκτυα μεγάλων οργανισμών-εταιριών. Ο λόγος για αυτό είναι ότι η ανεξαρτησία από ειδικά μέρη middleware επιβάλλει περιορισμούς στην αυτονομία των προμηθευτών υπηρεσιών.

3.5.3.2. METEOR2

Το project METEOR2 [35] προσεγγίζει τη δρομολόγηση δικτύων από κομμάτια-μέρη (components). Με αυτό τον τρόπο μειώνεται η ποσότητα της πληροφορίας, που πρέπει να μεταφερθεί ανάμεσα στις οντότητες επεξεργασίας. Το METEOR2 έχει αναπτύξει δύο πρωτότυπα: το ORB Work, που βασίζεται στη CORBA και το WebWork, που βασίζεται ολοκληρωτικά στις τεχνολογίες διαδικτύου. Και τα δύο παραπάνω πρωτότυπα υποστηρίζουν τη δρομολόγηση κατανεμημένων ροών εργασίας. Ο προσδιορισμός της ροής εργασίας αποθηκεύεται σε μια μορφή, η οποία περιλαμβάνει τις εξαρτήσεις προκάτοχου-διάδοχου των έργων (tasks), τους ορισμούς των αντικειμένων των δεδομένων που μεταφέρονται μεταξύ έργων, καθώς επίσης και τις πληροφορίες έναρξης λειτουργίας των έργων (task invocation information). Ένας γεννήτορας κώδικα παράγει μέρη διαχείρισης έργων και wrappers, τόσο για έργα που έχουν αλληλεπίδραση με ανθρώπους όσο και για έργα αυτοματοποιημένα. Κάθε wrapper περιλαμβάνει ένα έμφυτο (hard-wire) προσδιορισμό των προκατόχων και διαδόχων του έργου που διαχειρίζεται, καθώς επίσης και κώδικα, ο οποίος αποτιμά τη συνθήκη ενεργοποίησης του έργου. Ο wrapper, αναλυτικότερα, βάζει το έργο σε λειτουργία, χειρίζοντας παράλληλα, την ανάκαμψη (recovery) από λάθη. Μόλις τα παραπάνω μέρη εγκατασταθούν χειρωνακτικά σε κάθε τόπο (site), οι χρήστες μπορούν να δημιουργήσουν περιπτώσεις (instantiate) διαδικασιών. Ένα κεντρικοποιημένο όργανο παρακολούθησης παρέχει περιορισμένη υποστήριξη για εντοπισμό (tracking) και παρακολούθηση (monitoring) της εκτέλεσης περιπτώσεων διαδικασιών. Όταν ο κώδικας του wrapper για έργα ενεργοποιηθεί, παίρνει την είσοδο του έργου και βάζει σε λειτουργία τον κώδικα του έργου στην απονεμημένη οντότητα επεξεργασίας. Μόλις τελειώσει προσδιορίζει την τελική κατάσταση του έργου, πακετάρει τα αντικείμενα δεδομένων της εξόδου και ειδοποιεί τα διάδοχα έργα.

3.5.3.3. WebFlow and XPECT

Το paper [36] αναφέρει ότι το περιβάλλον του WebFlow παρέχει κατανεμημένο και μεταξύ πολλών υπηρετητών συντονισμό, αφενός με το να διευρύνει τους τυπικούς εξυπηρετητές του διαδικτύου σε διαχειριστές πόρων (resource managers) και αφετέρου με το να παρέχει πράκτορες συντονισμού. Το WebFlow βασίζεται στο Coordination Language Facility (CLF) [37], το οποίο και παρέχει ένα διευρυμένο μοντέλο αντικειμένων υποθέτοντας ότι όλα τα αντικείμενα μπορούν να θεωρηθούν ως διαχειριστές πόρων. Επιπλέον, το CLF υποστηρίζει δύο

βασικές λειτουργίες: την εισαγωγή και την αφαίρεση συγκεκριμένων πόρων. Οι πόροι μπορεί να μην είναι ορατοί από τους χρήστες. Ωστόσο, έχουν τη δυνατότητα να προσπελαστούν αλληλοσχετιζόμενα μέσω των ιδιοτήτων τους, οι οποίες έχουν ήδη οριστεί στη διεπαφή του αντικειμένου. Αυτού του είδους η αλληλοσχετιζόμενη πρόσβαση σε πόρους είναι παρόμοια με τη δυναμική προσάρτηση (binding), που υποστηρίζεται από μια υπηρεσία αποθήκης (repository). Οι αλληλεπιδράσεις πελάτη-εξυπηρετητή στο CLF προσδιορίζονται μέσω μιας γλώσσας, η οποία βασίζεται σε κανόνες. Οι κανόνες ορίζουν τους πράκτορες συντονισμού, καθορίζοντας κατά αυτόν τον τρόπο τη συμπεριφορά ενός πελάτη. Πρέπει να σημειωθεί ότι οι κανόνες της CLF είναι συναλλακτικές μονάδες, αφού εξασφαλίζουν ότι τόσο η αφαίρεση των πόρων (η αριστερή μεριά των κανόνων) όσο και η εισαγωγή των πόρων (δεξιά μεριά των κανόνων) θα εκτελούνται ατομικά. Το πρωτόκολλο της CLF περιλαμβάνει υποστήριξη για διαπραγμάτευση, με την έννοια ότι ένας πελάτης μπορεί να ζητήσει από έναν εξυπηρετητή να του χορηγήσει ειδοποιήσεις για τις υπηρεσίες που προσφέρει. Η CLF είναι, επίσης, η βάση για το πλαίσιο XPECT [38] για το ηλεκτρονικό εμπόριο. Σε αυτό το πλαίσιο, η CLF χρησιμοποιείται για το συντονισμό των 'εμπορικά συνδιαλλαγόμενων' ('commercial actors') (πελάτες, προμηθευτές, τραπεζίτες, μεσίτες) με τη φόρμα της διαπραγμάτευσης, της δοσοληψίας και της ειδοποίησης. Η διαπραγμάτευση συντελείται μεταξύ ενός πελάτη και πολλαπλών προμηθευτών, με σκοπό τον προσδιορισμό των επιλογών ενός συνόλου από τεμάχια που τον ενδιαφέρουν. Επίσης, συντελείται μεταξύ ενός προμηθευτή και ενός τραπεζίτη για τον προσδιορισμό του τρόπου πληρωμής (αν ο πελάτης εξουσιοδοτήσει αυτή την ενέργεια στο προμηθευτή). Αντιθέτως, η δοσοληψία συμβαίνει όταν μια φόρμα παραγγελίας διεκπεριώνεται από ένα τραπεζίτη. Τέλος, η ειδοποίηση χρησιμοποιείται σε περίπτωση που ο πελάτης ζητήσει να είναι ενήμερος για την εξέλιξη της αποστολής του εμπορεύματος σε όλες τις φάσεις της διαδικασίας παραδόσεως.

3.5.3.4. Middleware for Enterprise-Wide Workflow Management (MENTOR)

Το MENTOR [39] project στοχεύει στην ανάπτυξη ενός κλιμακωτού και υψηλά διαθέσιμου περιβάλλοντος για την εκτέλεση, την παρακολούθηση και τον έλεγχο των ροών εργασίας. Από αυτή την πλευρά, το MENTOR βασίζεται επί το πλείστο σε πρότυπα μέρη και προσθέτει-δημιουργεί νέα μέρη μόνο όταν τα πρότυπα μέρη τελειώσουν από λειτουργικότητα ή κλιμακωσιμότητα. Άρα, η αρχιτεκτονική του είναι ανοικτή και αρθρωτή (modular). Επιπλέον, η αρχιτεκτονική του MENTOR ολοκληρώνει μια οθόνη παρακολούθησης της επεξεργασίας δοσοληψιών, που χρησιμεύει στην ανοχή λαθών, ενώ παράλληλα εφαρμόζει μια υλοποίηση της CORBA, που αντιμετωπίζει την ετερογένεια [40] των εφαρμογών που έχουν τεθεί σε λειτουργία. Επιπρόσθετες ανάγκες, που προκύπτουν από τις εφαρμογές είναι οι ακόλουθες: υποστήριξη επερωτήσεων πάνω στην ιστορία των ροών εργασίας, υποστήριξη για διαχείριση τόσο του χρόνου όσο και της λίστας εργασιών, διαλειτουργικότητα των ετερογενών συστημάτων διαχείρισης ροών εργασίας και τέλος, επικύρωση των προσδιορισμών ροών εργασίας.

Η αρχιτεκτονική του MENTOR, η οποία φαίνεται στο σχήμα 1, βασίζεται στην αρχιτεκτονική πελάτη-εξυπηρετητή. Σύμφωνα, λοιπόν, με αυτή την αρχιτεκτονική, μια ροή εργασίας ενορχηστρώνεται από κατάλληλα σχηματισμένους εξυπηρετητές, ενώ οι εφαρμογές που τίθενται σε λειτουργία από διάφορες ενέργειες

μιας ροής εργασίας τρέχουν σε υπολογιστές του πελάτη (όπου οι εφαρμογές με τη σειρά τους μπορεί να εκδώσουν αιτήσεις για άλλους εξυπηρετητές, ανεξαρτήτου εάν η εκάστοτε εφαρμογή τίθεται σε λειτουργία μέσα από μία ροή εργασίας ή όχι). Στις εφαρμογές μεγάλων οργανισμών, οι ροές εργασίας μπορεί να χωριστούν σε πολλαπλά οργανωτικά μέρη, τα οποία είναι συχνά και σε μεγάλο βαθμό αυτόνομα. Επομένως, για λόγους ποικιλίας και κλιμακωσιμότητας, ακολουθείται η προσέγγιση ότι μια τεράστια ροή εργασίας μπορεί να διαμεριστεί σε ένα πεπερασμένο αριθμό από υπο-ροές εργασίας, κάθε μια από τις οποίες χειρίζεται ένας μόνο εξυπηρετητής.

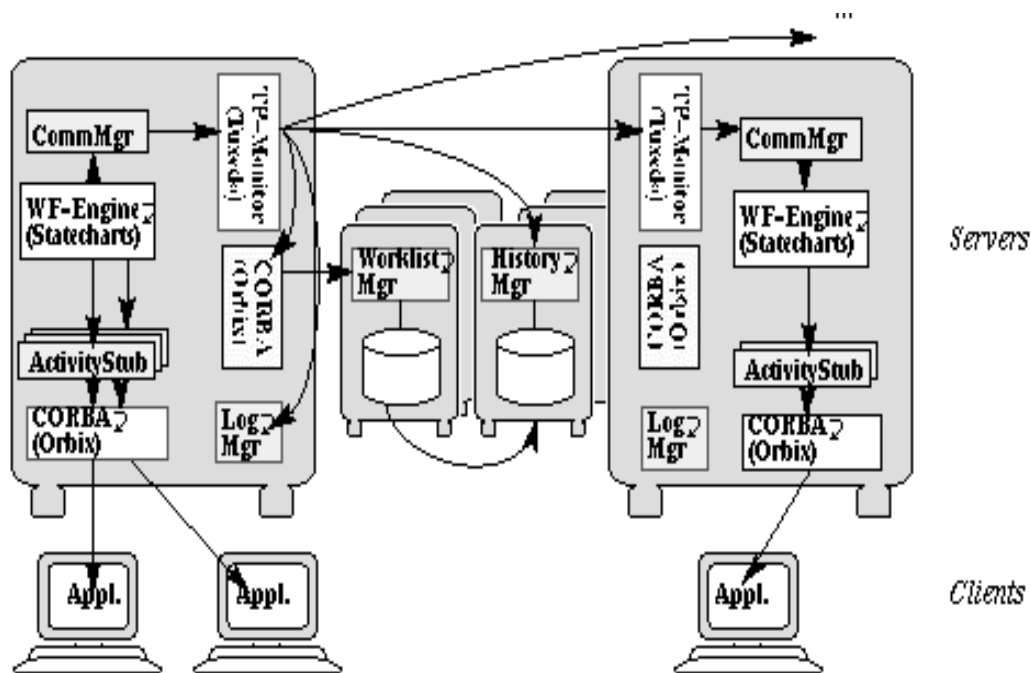


Figure 1: Client/Server Architecture of Mentor

Σχήμα 1 - Αρχιτεκτονική του MENTOR project

3.5.3.5. *Panta Rhei*

Το *Panta Rhei* [41] είναι ένα πρωτότυπο και κατανεμημένο σύστημα διαχείρισης ροών εργασίας, που έχει στρέψει την εστίασή του στις εξής τρεις κατευθύνσεις :

1. Οι εταιρείες, που είναι διανεμημένες σε πολλές τοποθεσίες, πρέπει να ενοποιηθούν. Ορισμένες εφαρμογές ενώνουν μεταξύ τους 4000 τόπους, γεωγραφικά κατανεμημένους, χρησιμοποιώντας ετερογενείς πλατφόρμες.
2. Οι διεπαφές από χρήστες, που δεν ανήκουν σε έναν οργανισμό, οφείλουν να είναι διαθέσιμες. Μια επιχειρησιακή διαδικασία, τυπικά, αρχίζει έξω από ένα οργανισμό, όπως για παράδειγμα η υποβολή αιτήσεων ή παραγγελιών.
3. Τα συστήματα ροών εργασίας διαφορετικών εταιρειών πρέπει να συνεργαστούν στις επιχειρησιακές τους διαδικασίες.

Το σύστημα αυτό θεωρεί ότι ο ορισμός μια ροής εργασίας βασίζεται στη μεταφορά του *formflow* [42]. Μέσα σε ένα μικρό οργανισμό, ο συντονισμός μεταξύ

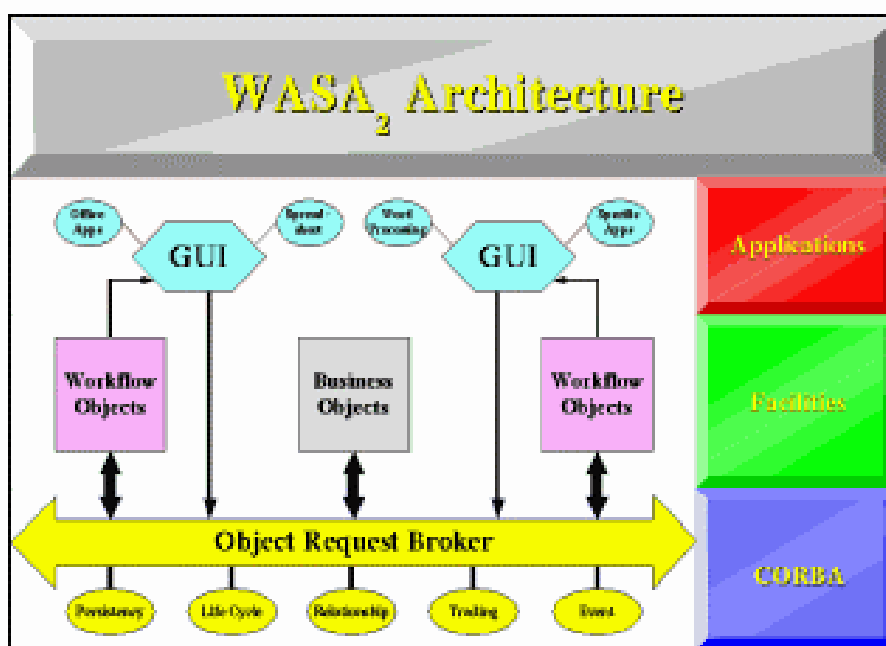
των έργων εκπληρώνεται με τη διαβίβαση των φορμών, οι οποίες περιέχουν εκείνα τα δεδομένα που είναι χρήσιμα για την εκτέλεση ενός έργου.

Η επικοινωνία μεταξύ δύο ή περισσότερων επιχειρήσεων πραγματοποιείται με την ανταλλαγή εγγράφων. Η παραλαβή ενός εγγράφου εγκαινιάζει ή συνεχίζει μια επιχειρησιακή διαδικασία από τη πλευρά του παραλήπτη. Η αποστολή ενός εγγράφου σε μια άλλη επιχείρηση είναι ένα (μερικό) αποτέλεσμα μιας επιχειρησιακής διαδικασίας. Επομένως, η formflow μεταφορά μπορεί να χρησιμοποιηθεί για την συνεργασία ροών εργασίας σε διαφορετικούς οργανισμούς.

3.5.3.6. WASA

Το WASA [43] είναι ένα ευέλικτο σύστημα διαχείρισης ροών εργασίας και παρουσιάζει τα εξής χαρακτηριστικά :

- Γραφικοί ορισμοί ροών εργασίας, που αλληλεπιδρούν μεταξύ τους.
- Κατανεμημένες και εξακολουθητικές (persistent) εκτελέσεις ροών εργασίας.
- Δυναμική προσαρμοστικότητα των ροών εργασίας που εκτελούνται.
- Ολοκλήρωση επιχειρησιακών αντικειμένων.
- Αντικειμενοστραφής σχεδίαση μέσω της UML [67].
- Τυπικές υπηρεσίες middleware με τη βοήθεια της τεχνολογίας CORBA.



Σχήμα 2 - Αρχιτεκτονική του WASA2 project

Το WASA παρουσιάζει μια αρχιτεκτονική τριών επιπέδων, όπως φαίνεται και στο σχήμα 2. Οι χρήστες προσπελαίνουν το σύστημα με μια γραφική διεπαφή χρήσης, η οποία μπορεί να προκαλέσει-ξεκινήσει εξωτερικά προγράμματα, τα οποία θα υλοποιήσουν τις ενέργειες της ροής εργασίας. Παραδείγματα εξωτερικών προγραμμάτων είναι εφαρμογές γραφείου και συστήματα λογισμικού, που υλοποιούν συγκεκριμένα έργα σε μια δεδομένη ροή εργασίας.

Το κύριο επίπεδο στην αρχιτεκτονική του WASA είναι το επίπεδο των εγκαταστάσεων-μονάδων (facilities), όπου εδρεύουν και αντικείμενα ροών εργασίας και επιχειρησιακά αντικείμενα. Ενώ τα αντικείμενα ροών εργασίας εκτελούν ειδικά έργα σχετιζόμενα με ροές εργασίας, τα επιχειρησιακά αντικείμενα είναι ειδικά του πεδίου εφαρμογής. Εφόσον, τόσο τα επιχειρησιακά αντικείμενα όσο και τα αντικείμενα ροών εργασίας, βασίζονται στην ίδια υποδομή middleware, επιτυγχάνεται μια μονοκόμμη ολοκλήρωση των επιχειρησιακών αντικειμένων μέσα στις εφαρμογές ροών εργασίας. Άρα, η αρχιτεκτονική βασιζόμενη σε κομμάτια-μέρη (component-based) του WASA επιτρέπει την επαναχρησιμοποίηση των επιχειρησιακών και ροών εργασίας αντικειμένων σε διαφορετικές CORBA εφαρμογές.

Στο χαμηλότερο επίπεδο, πρότυπες υπηρεσίες CorbaServices χρησιμοποιούνται για την κάλυψη των αναγκών των αξιόπιστων και αποδοτικά κατανεμημένων ροών εργασίας.

3.6. Γλώσσες και Μετα-γλώσσες Ορισμού Ροών Εργασίας

3.6.1 Ορισμός Γλώσσας Προσδιορισμού Ροών Εργασίας

Οι γλώσσες ορισμού ροών εργασίας είναι ειδικές γλώσσες, που χρησιμοποιούνται για τον ορισμό όλων των πλευρών μιας ροής εργασίας. Οι πλευρές αυτές περιλαμβάνουν :

- Ορισμό των ενεργειών, που πρέπει να γίνουν σε κάθε πιθανή κατάσταση.
- 'Πριν' και 'μετά από' συνθήκες των καταστάσεων.
- Μεταβάσεις μεταξύ καταστάσεων.
- Ορισμό της ακολουθίας των καταστάσεων / έργων.
- Ορισμό των αυτοματοποιημένων καταστάσεων και των καταστάσεων που απαιτούν είσοδο από το χρήστη.

Οι ενέργειες μπορεί να είναι απλές λειτουργίες, όπως η μεταφορά ενός εγγράφου από ένα άτομο σε ένα άλλο ή ακόμη και πολύπλοκες, εμπλέκοντας άλλες λειτουργίες μέσα σε μια εξωτερική legacy εφαρμογή. Οι 'πριν και μετά' συνθήκες μπορούν να χρησιμοποιηθούν για τον έλεγχο της ακεραιότητας του συστήματος ροών εργασίας, παραδείγματος χάριν για τον υπολογισμό του ολικού ισοζυγίου των νομισματικών συναλλαγών ενός λογαριασμού. Μπορεί να υπάρχουν πολλές μεταβάσεις από μια κατάσταση, και η επιλεγμένη κάθε φορά μετάβαση εξαρτάται είτε από τους αυτοματοποιημένους υπολογισμούς είτε από την είσοδο του χρήστη. Μια κατάσταση μπορεί να μοντελοποιηθεί έτσι ώστε να αποτελείται από πολλές υποκαταστάσεις, οι οποίες να συμβαίνουν είτε ταυτόχρονα είτε ακολουθιακά με αυτήν την κατάσταση. Αυτή η τεχνική μοντελοποίησης είναι χρήσιμη ειδικά για εκείνες τις περιπτώσεις, όπου το σύστημα ροών εργασίας είναι αρκετά πολύπλοκο, οπότε και δύναται να παρουσιαστεί σε πολλά επίπεδα αφαίρεσης.

Η γλώσσα ορισμού ροών εργασίας ερμηνεύεται και εκτελείται από το λογισμικό πράξεως ροών εργασίας (workflow enactment software), το οποίο με τη σειρά του εκτελεί τα καθορισμένα έργα, εμπλέκει άλλες εφαρμογές (όπως είναι ορισμένες από το σύστημα) και απαιτεί είσοδο από το χρήστη όποτε αυτή κρίνεται απαραίτητη.

3.6.2. Είδη Γλωσσών Προσδιορισμού Ροών Εργασίας

Για τον προσδιορισμό των διαδικασιών ροών εργασίας, διάφορες γλώσσες έχουν αναπτυχθεί κατά καιρούς. Σχεδόν κάθε πωλητής βασίζεται σε μια αποκλειστική φόρμα για τον προσδιορισμό των μοντέλων ροών εργασίας, αλλά μόνο λίγοι χρησιμοποιούν υπάρχουσες γλώσσες μοντελοποίησης διαδικασιών, όπως είναι τα υψηλού επιπέδου Petri-Nets, για το κομμάτι-μέρος της μοντελοποίησης του προϊόντος τους. Σύμφωνα με τον Carlsen [45], οι υπάρχουσες γλώσσες μοντελοποίησης των διαδικασιών ροών εργασίας μπορούν να ταξινομηθούν στις εξής πέντε κατηγορίες :

1. Input-Process-Output (IPO) βασιζόμενες γλώσσες, όπως είναι τα δίκτυα ενεργειών που χρησιμοποιούνται στο IBM MQSeries Workflow [46]. Αυτές οι γλώσσες περιγράφουν μια ροή εργασίας ως άκυκλο γράφο από ενέργειες, υποδηλώνοντας την ακολουθία της εκτέλεσής τους.
2. Η προσέγγιση Speech-Act (που μερικές φορές λέγεται και προσέγγιση Language Action) όπως χρησιμοποιείται στο προϊόν 'Action Technologies Action Workflow' [47]. Αυτού του είδους οι προσεγγίσεις μοντελοποιούν μια ροή εργασίας ως μια αλληλεπίδραση μεταξύ (τουλάχιστο) δύο συμμετεχόντων, που ακολουθούν ένα δομημένο κύκλο συνομιλίας. Ονομαστικά ξεχωρίζουν οι φάσεις της διαπραγμάτευσης, της αποδοχής, της εκτέλεσης και της ανασκόπησης.
3. Μέθοδοι μοντελοποίησης βασιζόμενοι σε περιορισμούς, όπως είναι η γραμματική 'Generalized Process Structure Grammar', που προτάθηκε από το GLANCE [48]. Αυτού του είδους οι προσεγγίσεις περιγράφουν μια διαδικασία ως ένα σύνολο από περιορισμούς, αφήνοντας χώρο για ευλυγισία, που αλλιώς χάνεται από τις αντίστοιχες προσεγγίσεις του IPO και του Speech-Act. Αυτές οι επικείμενες γλώσσες μοντελοποίησης είναι βασισμένες σε κείμενο και μοιάζουν με τις παραδοσιακές γλώσσες προγραμματισμού. Οπότε, η γραφική αναπαράσταση αυτών των μοντέλων φαίνεται δύσκολη.
4. Προσδιορισμοί διαδικασιών βάση του μοντέλου των ρόλων, όπως είναι τα 'Role Activity Diagrams (RADs).
5. Η σκέψη και η δυναμική συστημάτων (systems thinking and systems dynamics), που χρησιμοποιούνται παράλληλα με την έννοια των μαθητευόμενων οργανισμών (learning organizations) [49].

3.6.3. Μετα-γλώσσες για τον προσδιορισμό διαδικασιών

Η ποικιλία των προσεγγίσεων και των εργαλείων προσδιορισμού ροών εργασίας οδηγεί σε πολλά προβλήματα όχι μόνο τους χρήστες, που θέλουν να ασχοληθούν με την ανάπτυξη προϊόντων διαφορετικής μοντελοποίησης, αλλά και εκείνους που σχεδιάζουν να ολοκληρώσουν τις επιχειρησιακές τους διεργασίες με προηγούμενες και επόμενες φάσεις της αλυσίδας αξιών (όπως είναι οι προμηθευτές και οι πελάτες). Για την διευκόλυνση της ολοκλήρωσης των εργαλείων από τους πωλητές (vendors), διάφορες πρωτοβουλίες έχουν αναδυθεί, οι οποίες εστιάζονται στην ανάπτυξη των μετα-γλωσσών για την μοντελοποίηση διαδικασιών.

Η μετα-γλώσσα είναι το σύνολο εκείνων των δομικών μερών, που εντοπίζονται στις γλώσσες μοντελοποίησης διαδικασιών και χρησιμοποιούνται για την αντιστοίχιση εννοιών μεταξύ δύο γλωσσών μοντελοποίησης.

3.6.4. Παραδείγματα Γλωσσών Προσδιορισμού Ροών Εργασίας

3.6.4.1. IPL και WFSL

Γλώσσες προσδιορισμού ροών ελέγχου και δεδομένων έχουν παρουσιαστεί στα άρθρα [50,51]. Το [50] περιγράφει το πρότυπο InterBase, το οποίο υποστηρίζει ένα εύκαμπτο μοντέλο δοσοληψιών που επιτρέπει καθολικές (global) δοσοληψίες να εκτελούνται πάνω από ετερογενή συστήματα βάσεων δεδομένων. Οι καθολικές δοσοληψίες προσδιορίζονται με τη χρησιμοποίηση της γλώσσας InterBase Parallel Language (IPL), ως υπο-δοσοληψίες με εξαρτήσεις ροών ελέγχου και δεδομένων. Ο προσδιορισμός περιλαμβάνει τις αποδεκτές εκβάσεις για κάθε καθολική δοσοληψία, σε σχέση με τα αποτελέσματα των εκβάσεων των υπο-δοσοληψιών της εν λόγω δοσοληψίας. Το [51] παρουσιάζει τη γλώσσα προσδιορισμού ροών εργασίας WFSL για τη μοντελοποίηση της ροής των διαδικασιών και τη γλώσσα προσδιορισμού έργων TSL, η οποία και παρέχει μια διεπαφή χρήσης σε ετερογενή, αυτόνομα και καταναμημένα συστήματα. Επομένως, διαχωρίζεται η σχεδίαση της ροής εργασίας από τις λεπτομέρειες της αλληλεπίδρασης της με τις εφαρμογές. Η WFSL επιτρέπει μια δηλωτική περιγραφή της ροής ελέγχου και δεδομένων, βασιζόμενη σε κανόνες. Ένα σημαντικό σημείο στο άρθρο [51] είναι το λεπτομερές μοντέλο εξάρτησης έργων, που επιτρέπει τη διαχείριση της ατομικότητας και της συνέπειας των ροών εργασίας, με το να συλλαμβάνει τη συμπεριφορά τόσο των συναλλακτικών όσο και των μη συναλλακτικών έργων (non-transactional tasks).

3.6.4.2. FDL

Υπάρχουν πολλά προϊόντα διαχείρισης ροών εργασίας, που ειδικεύονται στην διαχείριση δομημένων διαδικασιών. Ένα από αυτά είναι το FlowMark [52] από την IBM [53], το οποίο προσφέρει τη γλώσσα προσδιορισμού FDL για τον ορισμό μοντέλων διαδικασίας με βάση ένα κατευθυνόμενο ακυκλικό γράφο από ενέργειες. Αυτή η γλώσσα είναι αντιπροσωπευτική της κορυφής της τεχνολογίας στα εμπορικά προϊόντα ροών εργασίας. Οι κόμβοι στο γράφο της διαδικασίας αντιστοιχούν σε ενέργειες, οι οποίες κατά το χρόνο εκτέλεσης 'ζευγαρώνουν' με πόρους. Οι πόροι μπορεί να είναι προγράμματα εφαρμογών ή χρήστες. Για κάθε ενέργεια, ένα container εισόδου αντιπροσωπεύει το πλαίσιο της ενέργειας. Το πραγματικό περιεχόμενο μιας περίπτωσης δοσμένης ενέργειας είναι ένα σύνολο από τιμές τυπικών παραμέτρων, που καθορίστηκαν για τη συγκεκριμένη περίπτωση ενέργειας. Το container εξόδου εγκλείει τα αποτελέσματα, που παράγονται από την εκτέλεση μιας ενέργειας. Τα τόξα στο γράφο του μοντέλου διαδικασίας ορίζουν τη ροή του ελέγχου και τη ροή των δεδομένων. Η αληθινή ροή πραγματοποιείται κατά το χρόνο εκτέλεσης, λογαριάζοντας και τις συνθήκες που εξαρτώνται από το πλαίσιο. Οι συνδετήρες ελέγχου περιγράφουν τη ενδεχόμενη ροή του ελέγχου, ενώ οι συνδετήρες δεδομένων, που συνδέουν το container εξόδου μιας ενέργειας με το container εισόδου μιας άλλης, περιγράφουν τη ροή των δεδομένων καθώς η ροή εργασίας εξελίσσεται. Μεταβατικές συνθήκες ορίζονται (Boolean εκφράσεις) για τους συνδετήρες ελέγχου, έτσι ώστε η πραγματική ροή του ελέγχου να είναι ανεξάρτητη του πλαισίου. Συνθήκη εξόδου ορίζεται για κάθε ενέργεια. Η συνθήκη εξόδου παίρνει την τιμή TRUE μόνο εάν η

αντίστοιχη ενέργεια ολοκληρώνεται με επιτυχία. Ένα τελευταίο χαρακτηριστικό είναι ότι συνθήκες συγχρονισμού μπορούν να οριστούν για την αντιμετώπιση ακολουθιών ελέγχου και δεδομένων, όταν πολλαπλοί συνδεδετές ελέγχου έχουν ως αφετηρία μια ενέργεια.

3.6.4.3. WIDE language

Εκτός από τους γράφους διαδικασιών, και τα μοντέλα διαδικασιών μπορούν να περιγραφθούν με βάση τις αναθέσεις δουλειάς για συγκεκριμένους ρόλους, αντί για ανεξάρτητους εργάτες ή πόρους. Επίσης, ένα σύστημα διαχείρισης ροών εργασίας μπορεί να εφαρμόσει μια πολιτική ανάλυσης ρόλων για να προσδιορίσει, από μια 'δεξαμενή' από διαθέσιμους και κατάλληλους εργάτες και πόρους, την πραγματική κατανομή κατά το χρόνο εκτέλεσης μιας διαδικασίας. Η γλώσσα προσδιορισμού ροών εργασίας, που παρέχει το WIDE πρωτότυπο [54], υποστηρίζει τα παραπάνω με τον ορισμό των προ-συνθηκών, των μετα-συνθηκών, των δράσεων και των διαχειριστών εξαιρέσεων για κάθε ενέργεια καθώς και ενός περιορισμού ρόλου, ο οποίος ελέγχει τη ανάθεση της ενέργειας σε ένα πράκτορα προς εκτέλεση. Ο προσδιορισμός μιας διαδικασίας περιλαμβάνει όχι μόνο τη ροή ελέγχου και τη μεταχείριση λαθών μέσω κανόνων, αλλά και τη ροή δεδομένων μέσω του περάσματος παραμέτρων σε φόρμες, οι οποίες χρησιμοποιούνται σαν ένα μέσο ομαδοποίησης και διαμοιρασμού δεδομένων.

3.6.4.4. XRL

Η XRL [55] είναι αρχιτεκτονική και γλώσσα, που παρέχει δρομολόγηση ροών εργασίας μεταξύ εμπορικών συνεργατών για την παροχή διαδικτυακών ηλεκτρονικών και εμπορικών υπηρεσιών. Το κύριο χαρακτηριστικό της είναι ότι παρέχει ένα μηχανισμό περιγραφής διαδικασιών στο επίπεδο των περιπτώσεων. Είναι βασισμένη στη τεχνολογία της XML και η σημασιολογία της εκφράζεται με βάση τα Petri-nets [56,57,58,59]. Η αρχιτεκτονική της βασίζεται στην έννοια του σημειώματος δρομολόγησης (routing slip), που ορίζει το σχήμα της δρομολόγησης. Η routing slip είναι μια ακολουθία από ρόλους ή χρήστες, που πρέπει να αναθεωρήσουν έγγραφα, τα οποία βρίσκονται σε μια συγκεκριμένη σειρά. Ένα ή παραπάνω έγγραφα μπορούν να προστεθούν, να αλλαχθούν ή να αφαιρεθούν από τη routing slip με την παρέμβαση υπαλλήλων. Ένας ιδιοκτήτης δημιουργεί τη routing slip και ορίζει τους συνεργάτες δρομολόγησης. Κάθε slip έχει ένα μοναδικό ID, που μπορεί να χρησιμοποιηθεί για την ανίχνευσή του. Το σχήμα δρομολόγησης της XRL γίνεται αποδεκτό σε ένα κόμβο μέσω ηλεκτρονικού μηνύματος. Αυτό επεξεργάζεται με τη βοήθεια ενός XML parser και αποθηκεύεται ως μια δομή δεδομένων. Ο πυρήνας της μηχανής ροών εργασίας διαβάζει αυτή τη δομή και δημιουργεί μια Petri-net αναπαράσταση. Βασιζόμενη σε αυτή την αναπαράσταση, η μηχανή προσδιορίζει το επόμενο βήμα(τα) που πρέπει να εκτελεστεί και το παρουσιάζει στο χρήστη μέσω μιας διεπαφής χρήσης. Ο χρήστης εισάγει τη δράση, που θα εκτελέσει για να ολοκληρώσει το βήμα και ενημερώνει τη μηχανή ροών εργασίας, η οποία και αναλαμβάνει να προσδιορίσει το επόμενο βήμα. Όταν όλα τα βήματα σε ένα κόμβο ολοκληρωθούν, η μηχανή ροών εργασίας αποθηκεύει την αναθεωρημένη δομή δεδομένων και τη στέλνει στον επόμενο κόμβο της ροής εργασίας.

3.6.4.5. Workflow Definition Language 2 (WDL2)

Η γλώσσα Workflow Definition Language 2 [60] είναι μια γλώσσα προσδιορισμού ροών εργασίας και χρησιμοποιείται στο σύστημα ροών εργασίας WorkMan [61]. Το WorkMan είναι ένα project του πανεπιστημίου του Helsinki. Η WDL2 γλώσσα ορίζεται με τη βοήθεια της XML. Κάθε στοιχείο της ροής εργασίας και κάθε λογικό έργο ορίζεται με τη χρησιμοποίηση ενός XML αρχείου. Εν συνεχεία, τα αρχεία ενώνονται μέσω της μεθόδου XLink [62] που ορίζεται από το World Wide Web Consortium (W3C) [63]. Αυτή η δόμηση προωθεί τη διαρθρωτικότητα και την αναχρησιμοποίηση των στοιχείων μιας ροής εργασίας.

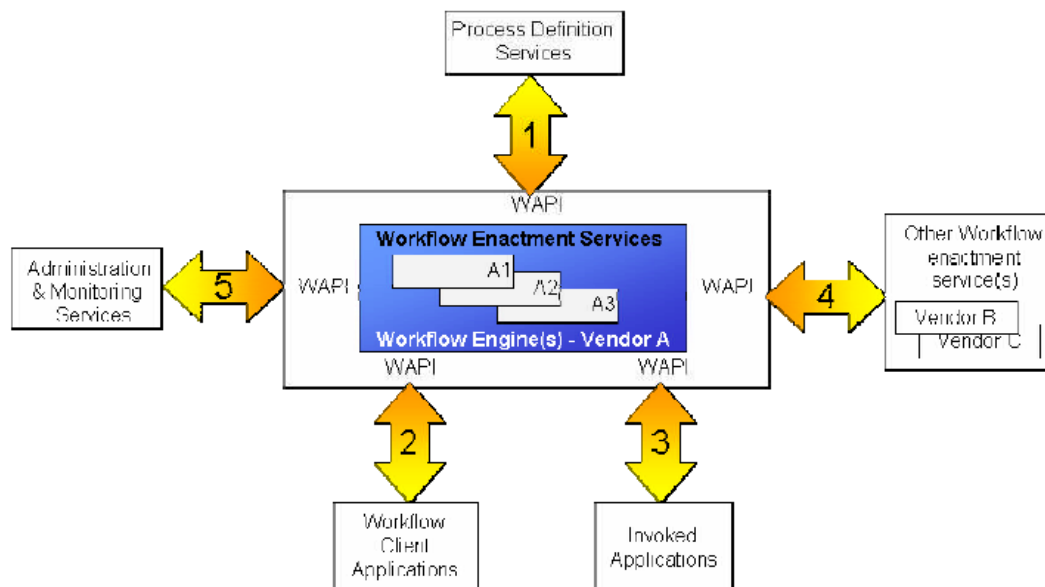
Η WDL2 περιέχει δομικά μέρη για υποθετικές δηλώσεις, ανακυκλώσεις, λογικούς χειριστές (operator), υπολογισμούς και πέρασμα παραμέτρων.

3.6.5. Παραδείγματα Μετα-γλωσσών Προσδιορισμού Ροών Εργασίας

Στις επόμενες υπο-ενότητες θα παρουσιάσουμε πέντε διαφορετικές προσεγγίσεις μετα-γλωσσών για μοντελοποίηση διαδικασιών, οι οποίες και είναι: Workflow Process Definition Language, Process Interchange Framework, Process Specification Language, Generalized Process Structured Grammar και Unified Modelling Language.

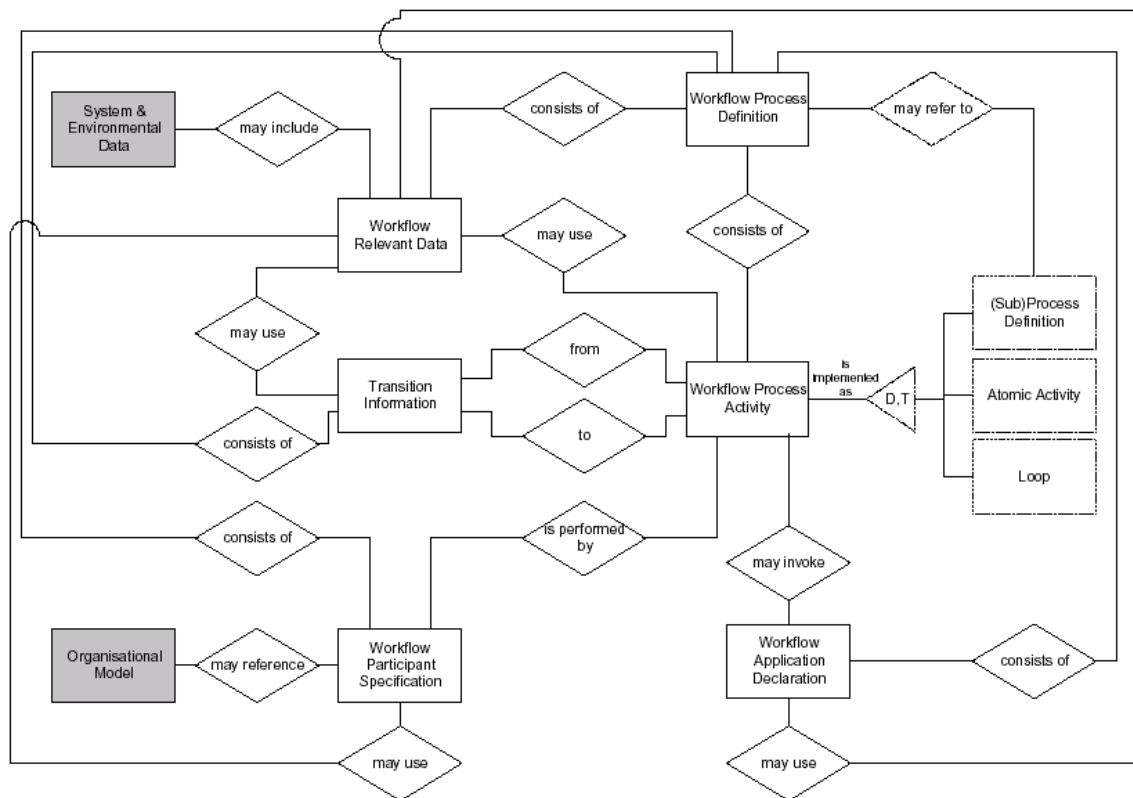
3.6.5.1. Workflow Process Definition Language (WPDL)

Η γλώσσα 'Workflow Process Definition Language' (WPDL) συστήθηκε πρώτα από τον οργανισμό WFMC το 1994. Έκτοτε έχει εξελιχθεί αρκετά και μάλιστα έχει αλλάξει και φόρμα αναπαράστασης, διότι τώρα βασίζεται στη φόρμα XML και έχει μετονομαστεί σε 'XML Process Definition Language' [64].



Σχήμα 3 - Workflow Reference Model

Ο προσδιορισμός του WFMC κινείται γύρω από ένα μοντέλο αναφοράς (reference model), που περιγράφει τα βασικά στοιχεία ενός συστήματος διαχείρισης ροών εργασίας και εστιάζεται στις διεπαφές χρήσης με εξωτερικά συστήματα. Το μοντέλο αυτό, το οποίο φαίνεται στο σχήμα 3, περιγράφει αποκλειστικά και μόνο εκείνα τα μέρη ενός συστήματος διαχείρισης ροών εργασίας, που είναι υποψήφια για τη διαλειτουργικότητα με άλλα συστήματα λογισμικού. Οι πέντε διεπαφές χρήσης του μοντέλου αναφοράς συνδέονται με τις υπηρεσίες πράξης ροών εργασίας μέσω του Workflow Application Programming Interface (WAPI), το οποίο έχει οριστεί σε ένα αφαιρετικό επίπεδο [65]. Η WPDL εγκαθιδρύθηκε ως μια μεταγλώσσα για την ανταλλαγή μοντέλων διαδικασίας ροών εργασίας του build-time μέσω μιας batch διεργασίας. Οι λέξεις κλειδιά της WPDL βασίζονται στους όρους, που ορίζονται στο γλωσσάριο του WFMC. Η σχεδίαση της γλώσσας βασίζεται σε ένα ελάχιστο μετα-μοντέλο, το οποίο ορίζει τα στοιχειώδη μέρη που πρέπει να υποστηριχθούν από ένα εργαλείο που διαβάζει ή / και γράφει σε WPDL. Αυτό το ελάχιστο μετα-μοντέλο μπορεί να επεκταθεί ανάλογα με τις ανάγκες μιας εφαρμογής.



Σχήμα 4 - Το μετα-μοντέλο του μέρους της διαδικασίας της WPDL

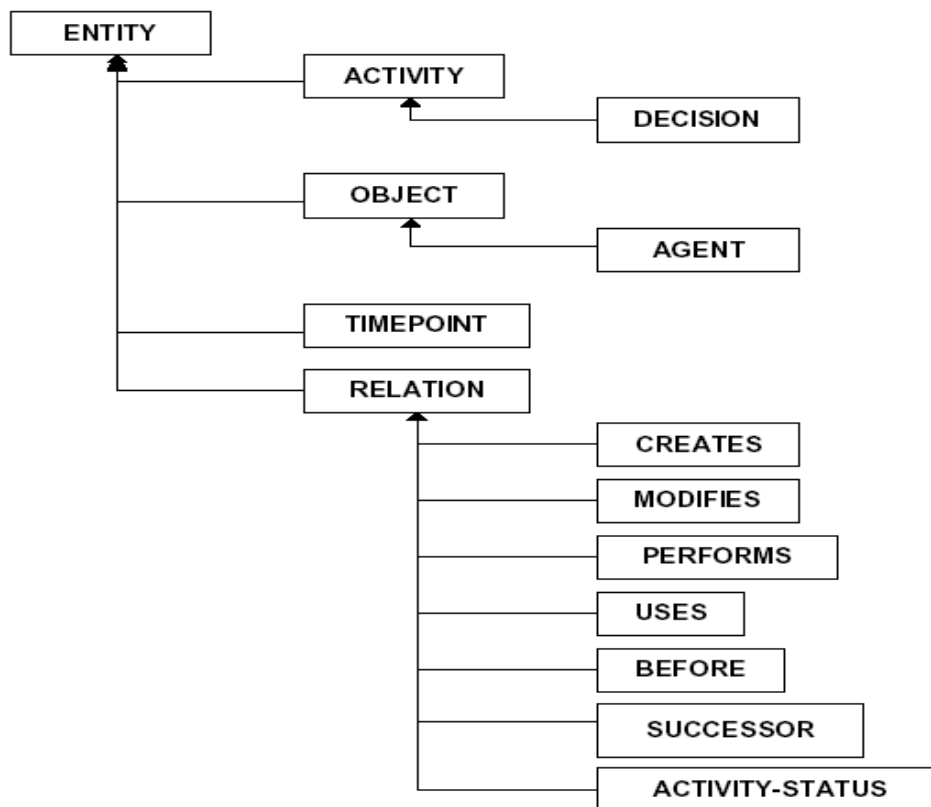
Το μετα-μοντέλο του μέρους της διαδικασίας της WPDL φαίνεται στο σχήμα 4. Το οργανωτικό και των πόρων μοντέλο είναι ειδικεύσεις της οντότητας ‘Workflow Participant Specification Entity Type’ και επομένως, δεν φαίνεται στο προαναφερόμενο σχήμα. Η κύρια έννοια της WPDL είναι η ‘Workflow Process Definition’ και αποτελείται από μια ή παραπάνω έννοιες ‘Workflow Process Activities’. Η σειρά των τελευταίων ενεργειών-εννοιών καθορίζεται από τα στοιχεία ‘Transition Information’, που συνδέουν απλές οντότητες. Για πιο πολύπλοκες δρομολογήσεις, μια μετάβαση (Transition) μπορεί να βασιστεί σε δεδομένα σχετικά με τη ροή εργασίας (Workflow Relevant Data), τα οποία είναι επίσης δεδομένα για συστήματα εφαρμογών που έχουν σχέση με την ακολουθία των ενεργειών.

Οι τύποι οντοτήτων της WPDL δεν είναι επεκτάσιμοι, αλλά ιδιότητες οριζόμενες από χρήστες μπορούν να προστεθούν σε απλούς τύπους οντοτήτων της WPDL. Επιπλέον, αναφορές σε εξωτερικές πηγές δεδομένων ως σημεία σύνδεσης μπορούν να υποδηλωθούν άμεσα, όπως είναι η αναφορά σε ένα εξωτερικό αποθηκευτικό χώρο ενός οργανισμού, σε συστήματα ή σε περιβαλλοντολογικά δεδομένα, ή και σε συστήματα εφαρμογών που έχουν ανακινηθεί.

Για να επιβληθεί συμμόρφωση των συστημάτων διαχείρισης ροών εργασίας, που ακολουθούν διαφορετικά παραδείγματα μοντελοποίησης, διάφορες κλάσεις συμμόρφωσης πρέπει να οριστούν. Αυτές οι κλάσεις περιορίζουν τον αριθμό των στοιχείων, που πρέπει να αναδειχθούν από ένα σύστημα διαχείρισης ροών εργασίας για την υποστήριξη συμμόρφωσης στην WPDL. Ένας συνήθης περιορισμός είναι του τύπου block για συστήματα ροών εργασίας, ο οποίος απαιτεί κάθε διχοτομία (split) μιας διαδικασίας να ακολουθείται από μια παρόμοια ένωση (join) σε ένα μετέπειτα μέρος της διαδικασίας.

3.6.5.2. Process Interchange Framework (PIF)

Το πλαίσιο ‘Process Interchange Framework’ αναπτύχθηκε ως μια πρότυπη γλώσσα για διαδικασίες από το project ‘MIT Process Handbook’ [66]. Το project ‘Process Handbook’ στοχεύει στη συλλογή αντιπροσωπευτικών επιχειρησιακών διαδικασιών και στην παρουσίαση αυτών των, ώστε να διευκολυνθεί η σύγκριση και η συλλογή εναλλακτικών διαδικασιών κάτω από πραγματικές επιχειρησιακές συνθήκες. Ο κύριος σκοπός του είναι η υποστήριξη αφενός των οργανισμών, που αναζητούν ανα-σχεδίαση υπάρχουσων διαδικασιών και αφετέρου των νέων διαδικασιών, που αναδύονται λόγω της τεχνολογικής ανάπτυξης.



Σχήμα 5 - Η ιεραρχία των μερών του πυρήνα της PIF

Μέσα από τη PIF προσέγγιση, οι διαδικασίες αναπαριστώνται σε διάφορα επίπεδα αφαίρεσης, τα οποία παράγονται από την αντικειμενοστραφή έννοια της κληρονομικότητας και από τη διαχείριση εξαρτήσεων, όπως αποτελεί η θεωρία συγχρονισμού. Οι δημιουργοί της PIF αναφέρουν ως το κύριο πλεονέκτημα της PIF είναι ότι επιτρέπει στους χρήστες να απεικονίζουν άμεσα τις ομοιότητες και τις διαφορές μεταξύ συσχετιζόμενων διαδικασιών και επιπλέον να βρίσκουν εύκολα ή να παράγουν λογικές εναλλακτικές λύσεις του τρόπου εκτέλεσης μιας διαδικασίας [67].

Όλα τα δομικά μέρη της PIF προσδιορίζονται μέσω του Knowledge Interchange Format (KIF), μια γλώσσα που έχει σχεδιαστεί για την ανταλλαγή γνώσης μεταξύ ξεχωριστών υπολογιστικών συστημάτων [68]. Η KIF επιτρέπει την επέκταση των υπάρχουσων εννοιών, πράγμα σημαντικό για τη προσθήκη επεκτάσεων οριζόμενων στη γλώσσα PIF από το χρήστη. Επιπροσθέτως, η KIF είναι ένα προτεινόμενο πρότυπο και διαθέτει καλά ορισμένη τυπική σημασιολογία, που απλοποιεί της διαδικασία ορισμού των δομικών μερών της PIF.

Μια περιγραφή διαδικασίας της PIF βασίζεται σε ένα σύνολο από ορισμούς πλαισίου (frame definitions). Κάθε ένας από τους ορισμούς πλαισίου υποδηλώνει ένα τύπο οντότητας, που μπορεί να έχει περιπτώσεις (για παράδειγμα TIMEPOINT ή ACTIVITY). Αυτοί οι τύποι υπάρχουν τακτοποιημένοι σε μια ιεραρχία. Η ιεραρχία των μερών του πυρήνα της PIF φαίνονται στο σχήμα 5. Για κάθε τύπο της PIF αντιστοιχεί ένα σύνολο από προκαθορισμένες ιδιότητες, οι οποίες ορίζουν διάφορες πλευρές της περίπτωσης αυτού του τύπου. Για παράδειγμα, ο ορισμός του 'CREATES' περιέχει μια 'ACTIVITY' και μια 'OBJECT' ιδιότητα, οι τιμές των οποίων φανερώνουν αντίστοιχα την ενέργεια που δημιουργεί ένα αντικείμενο(α) και το αντικείμενο που δημιουργείται. Οι ιδιότητες κληρονομούνται από τους υπερτύπους στους τύπους και από τύπους στις περιπτώσεις τους. Για παράδειγμα, μια περίπτωση του ορισμού κορμού της έννοιας 'ACTIVITY' περιέχει την ιδιότητα 'NAME', διότι ο τύπος 'ACTIVITY' κληρονόμησε αυτή την ιδιότητα από τον υπερτύπο 'ENTITY'. Η τιμή μιας ιδιότητας μέσα σε ένα πλαίσιο μπορεί να αναφέρεται σε ένα άλλο πλαίσιο. Με αυτό τον τρόπο, οι σχέσεις μεταξύ των περιπτώσεων των παραπάνω συσχετιζόμενων πλαισίων μπορούν να αναπαρασταθούν.

Η PIF είναι μια ισχυρή πλατφόρμα ανταλλαγής ορισμών διαδικασιών. Λόγω του διαρθρωτικού της σχεδιασμού μπορεί εύκολα να επεκταθεί, ώστε να ανταποκρίνεται στις ανάγκες της μοντελοποίησης διαδικασιών ροών εργασίας. Η ομάδα εργασίας της PIF ανταλλάσσει ιδέες με το WFMC, για να γίνουν οι γλώσσες PIF και WPDΛ διαλειτουργικές, και άρα να προετοιμάσουν το έδαφος για μια ενιαία φόρμα ανταλλαγής.

3.6.5.3. Process Specification Language (PSL)

Η γλώσσα PSL χρηματοδοτείται από το Ινστιτούτο 'National Institute of Standards and Technology' (NIST) [69]. Σκοπός αυτού του project είναι η ανάπτυξη μιας κοινής φόρμας ανταλλαγής για τις παραγωγικές επιχειρήσεις, η οποία θα είναι ανεξάρτητη από υπάρχουσες εφαρμογές και αρκετά δυνατή, ώστε να αναπαριστά τις απαραίτητες πληροφορίες μιας διαδικασίας για κάθε είδος εφαρμογής. Ο τελικός στόχος είναι η υποστήριξη της επικοινωνίας μεταξύ διαφορετικών εφαρμογών, που θα βασίζεται σε μια κοινή αντίληψη του περιβάλλοντός τους. Για τις διαδικασίες δεδομένων, η PSL έχει σχεδιαστεί με τέτοιο τρόπο ώστε να αποτελεί μια φόρμα ανταλλαγής, όπως είναι το STEP στο πεδίο των παραγωγικών δεδομένων (product data) [70].

Η κύρια έννοια της PSL για την αντιστοίχιση (mapping) μεταξύ δύο προγραμμάτων εφαρμογών είναι αρχικά η αντιστοίχιση της οντολογίας μοντελοποίησης της κάθε εφαρμογής στην οντολογία της PSL. Έπειτα, η διαδικασία της εφαρμογής πόρων (resource application process) μπορεί να απεικονιστεί με τη χρησιμοποίηση της KIF και να μεταμορφωθεί σε μια διαδικασία, που συμμορφώνεται με την οντολογία της PSL. Από αυτήν την ενδιάμεση διαδικασία, μια στοχευόμενη διαδικασία μπορεί να δημιουργηθεί με τη χρησιμοποίηση της οντολογίας της στοχευόμενης εφαρμογής και την KIF, η οποία με τη σειρά της μπορεί να εισαχθεί στη στοχευόμενη διαδικασία.

Τα βασικά μέρη της PSL είναι τα παρακάτω :

- Ενέργειες. Αυτές μπορεί να είναι χαρακτηριστικές ενέργειες, όπως deterministic ή η non-deterministic διαδικασίες, καθώς επίσης μπορεί να είναι και λειτουργίες εντολής (ordering functions), όπως σχέσεις creates ή precedes.
- Αντικείμενα. Μπορεί είτε να είναι πόροι, όπως άνθρωποι, μηχανές, ή καταστάσεις, όπως pre- και post-activity καταστάσεις.
- Σημεία χρονισμού. Αυτά μπορεί να χρησιμοποιηθούν είτε για την περιγραφή είτε χρονικών σχέσεων μεταξύ ενεργειών είτε της διάρκειας των διαδικασιών.

Όλες οι έννοιες της PSL προσδιορίζονται από την KIF, όπως γίνεται και με την PIF. Τελικά, τα projects PSL και PIF θα ενωθούν, αν και ένα συντονισμένο έγγραφο δεν είναι, ακόμη και στις μέρες, μας διαθέσιμο. Επίσης, έχει δημιουργηθεί και η αντιστοίχιση μεταξύ της PSL και της XML με το όνομα XPSL, αλλά και αυτό βρίσκεται στα πρώτα στάδια ανάπτυξής του.

3.6.5.4. Generalized Process Structure Grammars (GSPG)

Αν και οι γλώσσες WPD, PIF και PSL αναπαριστούν γλώσσες μοντελοποίησης διαδικασιών, βασισμένες στην κατηγορία IPO, η γραμματική 'Generalized Process Structure Grammars' (GSPG), όπως προτείνεται από το project 'GLANCE' [48], αναπαριστά μια προσέγγιση βασισμένη σε περιορισμούς για τη μοντελοποίηση των διαδικασιών. Αναλυτικότερα, προκειμένου να μοντελοποιηθεί μια διαδικασία κατασκευάζεται μια γραμματική, η οποία και θα περιέχει όχι μόνο τα νόμιμα στοιχεία της διαδικασίας, αλλά και τις συσχετίσεις-σχέσεις αυτών. Η γραμματική 'στενεύει' το χώρο της διαδικασίας, ώστε να περιέχει μόνο τους ζωτικούς περιορισμούς και τους κανόνες κατασκευής, ενώ κάθε τι που δεν περιορίζεται εξαρχής θεωρείται μεταβλητό κατά το χρόνο της πράξης της διαδικασίας (process enactment). Μια περίπτωση διαδικασίας στην GSPG δύναται να αποτελέσει μια νόμιμη φράση, που κατασκευάζεται με τη χρησιμοποίηση της γραμματικής του μοντέλου της εν λόγω διαδικασίας. Κάθε γραμματική GSPG περιέχει δύο είδη από κανόνες :

- Ενεργειο-κεντρικοί κανόνες (activity-centric rules), οι οποίοι διαχωρίζουν το στόχο μιας διαδικασίας σε υπο-στόχους και επισυνάπτουν περιορισμούς εκτέλεσης. Η διαδικασία αυτή είναι συγκρίσιμη με την ανάλυση των ενεργειών και του διαχωρισμού, όπως είναι γνωστή από τη Θεωρία Οργανισμών (Organizational Theory).
- Εγγραφο-κεντρικοί κανόνες, οι οποίοι περιγράφουν τα αντικείμενα δεδομένων, που χρησιμοποιούνται σε μια διαδικασία.

Η εφαρμογή της GSPG στη μοντελοποίηση διαδικασιών μπορεί να διευκρινιστεί με ένα απλό παράδειγμα. Αντιθέτως, η ακολουθία δύο ενεργειών σε μια IPO-based γλώσσα ορισμού διαδικασιών, συνήθως, υποδηλώνεται ως εξής:

B.start := A.end

μια GPSG-based γλώσσα μοντελοποίησης θα περιείχε τους εξής περιορισμούς:

B.start = A.end

B.end < deadline

B.start = B.end – B.average_duration

Επομένως, η GPSG-based ορισμοί διαδικασιών επιτρέπουν περισσότερη ευκαμψία κατά τη διάρκεια της πράξης μια διαδικασίας (process enactment). Το γεγονός αυτό οφείλεται στο ότι όταν οι διαδικασίες εκτελούνται δεν ακολουθούν ένα αυστηρά καθορισμένο σύνολο από μονοπάτια ροής ελέγχου και καταστάσεων, αλλά αναδύονται μέσα από το χώρο των διαδικασιών που ορίζονται από την αντίστοιχη γραμματική.

Ένα πρωτότυπο σύστημα με το όνομα FREEFLOW [71] έχει αναπτυχθεί με την GPSG προσέγγιση για την αναπαράσταση διαδικασιών. Η πολυπλοκότητα του μοντέλου διαδικασιών, βασιζόμενο σε περιορισμούς, οδηγεί τη μοντελοποίηση των διαδικασιών σε μια αναπαράσταση με κείμενο, που περιλαμβάνει και τους περιορισμούς. Παράλληλα, η γραφική αναπαράσταση των διαδικασιών μοιάζει δύσκολη, λόγω των πολλαπλών δυνατών μονοπατιών μιας διαδικασίας κατά το χρόνο εκτέλεσης της. Μια πρόσφορη επέκταση της GPSG για τη μοντελοποίηση διαδικασιών θα ήταν η μέριμνα για γραμματικά πρότυπα με στόχο συγκεκριμένους τύπους λειτουργιών ροών εργασίας, όπως είναι η διακλάδωση, η ανάθεση πόρων, η ανακίνηση εφαρμογών. Αν και η GPSG υπόσχεται ορισμούς μη δομημένων, κατά το χρόνο εκτέλεσης, διαδικασιών, που γίνονται αντιληπτές από το build-time, ωστόσο ο προσδιορισμός των ροών εργασίας για δοσοληψίες θα φέρει ως αποτέλεσμα πολύπλοκες γραμματικές και μάλλον θα πρέπει να γίνει αποδοτικότερος με τις IPO-based γλώσσες μοντελοποίησης διαδικασιών.

3.6.5.5. Unified Modelling Language (UML)

Αντίθετα με τις τέσσερις προηγούμενες προσεγγίσεις, οι οποίες εστιάζουν σε μια ορισμένη περιγραφή μέσω ενός κειμένου των διαδικασιών, η γλώσσα 'Unified Modelling Language' (UML) ορίζει τέσσερα διαφορετικά είδη διαγραμμάτων για τη σχεδίαση αντικειμενοστραφή συστημάτων λογισμικού [72,73,74]. Η UML σχεδιάστηκε γύρω από ένα περιορισμένο αριθμό βασικών εννοιών, που εν μέρει προϋπήρχαν πριν αυτή τυποποιηθεί το 1997. Οι έννοιες της UML μπορούν να επεκταθούν ή να ειδικευτούν από χρήστες. Επομένως, η UML επιτρέπει τον ορισμό πολύπλοκων συστημάτων και είναι ανεξάρτητος του πεδίου εφαρμογής. Οι διαφορετικοί τύποι διαγραμμάτων της UML οδηγούν στη σχεδίαση διαφορετικών όψεων ενός συστήματος.

Κατά τη διάρκεια των τελευταίων δεκαετιών, το ενδιαφέρον στην αντικειμενοστρέφεια ως ένα μέσο παροχής επεκτάσιμων, κλιμακώσιμων και καταναμημένων συστημάτων διαχείρισης ροών εργασίας έχει αυξηθεί σημαντικά [75]. Παράλληλα, ένας μεγάλος αριθμός από πρωτότυπα συστήματα, όπως το

WASA2, έχει αναπτυχθεί. Αρχίζοντας με την εσωτερική αρχιτεκτονική αυτών των συστημάτων, η ανάγκη για μια αντικειμενοστραφής μέθοδο μοντελοποίησης των μοντέλων ροών εργασίας οδήγησε στην αξιολόγηση της UML ως μια ακόμη γλώσσα μοντελοποίησης [76]. Αντίθετα με τις άλλες γλώσσες μοντελοποίησης, που έχουν προαναφερθεί, η UML προσφέρει γραφικές δηλώσεις (notations) των μοντέλων ροών εργασίας. Όμως, δεν υπάρχει μια απλή δήλωση, η οποία να μπορεί να χρησιμοποιηθεί για τη μοντελοποίηση όλων των πλευρών ενός μοντέλου ροής εργασίας. Επιπλέον, αρκετοί τύποι διαγραμμάτων πρέπει να χρησιμοποιηθούν για να μοντελοποιηθούν όλες τις πλευρές μιας διαδικασίας ροής εργασίας. Αυτοί οι τύποι διαγραμμάτων είναι οι εξής :

- Διαγράμματα περιστατικών-υποθέσεων χρήσης (use case diagrams). Τα διαγράμματα χρησιμοποιούνται για να απεικονίσουν την αλληλεπίδραση ενός συστήματος με το περιβάλλον του. Στην περίπτωση μιας διαδικασίας ροής εργασίας, οι υποθέσεις χρήσης μπορεί να χρησιμοποιηθούν για τη μοντελοποίηση της αλληλεπίδρασης μιας διαδικασίας ή μιας ενέργειας με εξωτερικούς παράγοντες-ενεργούντες (συμμετέχοντες σε ροή εργασίας, πελάτες, εξωτερικά συστήματα κ.λ.π.). Οι υποθέσεις χρήσης υποδηλώνουν μόνο τη στατική σχέση μεταξύ των ενεργούντων και της λειτουργικότητας του συστήματος. Επιπλέον, δεν περιγράφουν τη χρονική ή λογική ακολουθία των βημάτων μιας διαδικασίας. Η αποσύνθεση μιας διαδικασίας σε αρκετές υπο-διαδικασίες ή στοιχειώδεις ενέργειες μπορεί να μοντελοποιηθεί με τη χρησιμοποίηση της σχέσης ‘χρήσεων’ μεταξύ των υποθέσεων χρήσης (π.χ. ‘η διαχείριση παραγγελιών’ υπόθεση χρήσης χρησιμοποιεί τις υποθέσεις χρήσης: ‘εισήγαγε πελατειακά δεδομένα’, ‘τοποθέτησε παραγγελία’ και ‘κανόνισε την πληρωμή’).
- Διαγράμματα ακολουθίας (sequence diagrams). Τα διαγράμματα απεικονίζουν τη χρονική και λογική σειρά των ενεργειών και των εμπλεκόμενων συμμετεχόντων σε μια δήλωση της μορφής ‘swim-lane’. Αν οι διαφορετικοί ενεργούντες μέσα σε μια διαδικασία τακτοποιηθούν σε παράλληλες λωρίδες (lanes), η αλληλεπίδραση μεταξύ των συμμετεχόντων επιτρέπει μια αντιστοίχιση σε μοντέλα ροών εργασίας speech-act ή / και σε IPO-style.
- Διαγράμματα συνεργασίας (collaboration diagrams). Μέσα από τα διαγράμματα συνεργασίας περιγράφεται η αλληλεπίδραση μεταξύ των ενεργούντων και των υποθέσεων χρήσης, με βάση τα μηνύματα που στέλνονται μεταξύ των διαφορετικών στοιχείων του διαγράμματος. Αυτού του είδους τα διαγράμματα μπορεί να ειπωθούν ως μια επέκταση των διαγραμμάτων υποθέσεων χρήσης, αφού επιτρέπουν τη διάταξη των μηνυμάτων και των κατευθυνόμενων σχέσεων.
- Διαγράμματα καταστατικών χαρτών (state-chart diagrams). Ένα διάγραμμα καταστατικού χάρτη δείχνει όλες τις δυνατές καταστάσεις μιας υπόθεσης χρήσης και επιπλέον, τις μεταβάσεις αυτών των καταστάσεων. Αν αυτού του είδους το διάγραμμα ενταχθεί σε πλαίσιο διαχείρισης ροής εργασίας, τότε μπορεί να χρησιμοποιηθεί για την απεικόνιση των πιθανών σημείων έναρξης και τερματισμού ενός μοντέλου ροής εργασίας, καθώς επίσης και των νόμιμων μεταβάσεων μεταξύ καταστάσεων (π.χ. έτοιμη, άρχισε, αναστάληκε, τελείωσε, τερμάτισε κ.τ.λ.). Επιπροσθέτως, οι καταστατικοί χάρτες μπορούν να χρησιμοποιηθούν για τη περιγραφή της μεταμόρφωσης των αντικειμένων μιας διαδικασίας μέσα στα πλαίσια της δοσμένης διαδικασίας.

- Διαγράμματα ενεργειών (activity charts). Τα διαγράμματα ενεργειών είναι παραλλαγές των διαγραμμάτων καταστατικών χαρτών και παρουσιάζουν γραφικά όλα τα πιθανά μονοπάτια των δράσεων (acts) μεταξύ ενεργειών. Ενώ οι καταστατικοί χάρτες μπορεί να περιέχουν παθητικές καταστάσεις, τα διαγράμματα ενεργειών απεικονίζουν τις σχέσεις μεταξύ ενεργειών. Η μετάβαση μεταξύ δύο ενεργειών είναι ενεργή μόνο όταν η προηγούμενη ενέργεια έχει τελειώσει και ένας προαιρετικός περιορισμός φρούρησης (optional guard constraint) στη μετάβαση πάρει την τιμή true. Τα στοιχεία μοντελοποίησης επιτρέπουν παράλληλες διακλαδώσεις, καθώς επίσης και εναλλακτικές πορείες ανάμεσα σε ενέργειες. Αν και η πρωτότυπη δήλωση της UML οδηγεί σε πολύπλοκα διαγράμματα για εναλλακτικά μονοπάτια ή για υποθετικές διακλαδώσεις, η νέα έκδοση της UML παραθέτει μια πιο σύντομη δήλωση αυτών των υποθέσεων-περιπτώσεων.

Συνολικά, η UML προσφέρει ποικιλία από τύπους διαγραμμάτων και μπορεί να ανταποκριθεί στους διάφορους στόχους της μοντελοποίησης ροών εργασίας. Ωστόσο, η αναγκαιότητα για χρήση διαφορετικών τύπων διαγραμμάτων, που δεν είναι πάντοτε ορθογώνια (orthogonal), σε συνδυασμό με την έλλειψη της μοντελοποίησης πόρων μετατρέπουν την τρέχουσα έκδοση της UML δύσκολη στη χρήση των τρεχόντων εργαλείων ροών εργασίας. Πιθανώς, οι νέες εκδόσεις να προσφέρουν βελτιώσεις προς αυτήν την κατεύθυνση.

3.7. Διαλειτουργικότητα ροών εργασίας

Τα συστήματα ροών εργασίας, ανέκαθεν, μοιράζονται και κατανέμουν εργασία, και άρα πληροφορία και ευθύνη, μεταξύ διαφορετικών εργατών (workers). Όμως, αυτά τα συστήματα έχουν αυτοπεριοριστεί σε ένα μοναδικό οργανισμό με ένα μεμονωμένο σύνολο από ενδιαφέροντα (τα οποία όλοι οι εργάτες δεσμεύονται να μοιραστούν), καθώς και σε ένα μοναδικό υπολογιστικό σύστημα. Τώρα, δύο ή περισσότεροι οργανισμοί ή πρόσωπα, καθένα με τα δικά του ενδιαφέροντα, νομικά δικαιώματα, λεξιλόγιο, πρωτόκολλα και υπολογιστικά συστήματα, πρέπει να συνεργαστούν και να μοιραστούν μια μοναδική διαδικασία. Δυστυχώς, οι οργανισμοί υπό συνεργασία μπορεί να χρησιμοποιήσουν διαφορετικά προϊόντα ροών εργασίας, τα οποία είναι ποικίλα στη φύση τους και η διαβάθμισή τους ξεκινά από εκείνα που χρησιμοποιούνται για εξειδικευμένη δρομολόγηση των έργων ή των δεδομένων και ολοκληρώνεται σε εκείνα που στοχεύουν σε υψηλά κανονικοποιημένες παραγωγικές διαδικασίες. Πως κάποιος θα ολοκληρώσει διαφορετικά συστήματα ροών εργασίας, παραγόμενα από διαφορετικούς πωλητές, για να διαπεράσει τεμάχια εργασίας (work items) μονοκόμματα ανάμεσά τους; Αυτό είναι το πρόβλημα της διαλειτουργικότητας ροών εργασίας.

3.7.1. Η βάση του προβλήματος και η λύση των πρακτόρων

Βάση του προβλήματος αποτελεί η υπόθεση ότι τα συστήματα ροών εργασίας και οι πράκτορές τους δεν μοιράζονται μια συνεπή σημασιολογική αντίληψη της πληροφορίας, η οποία συναλλάσσεται. Επιπλέον, δεν υπάρχει αμοιβαία κατανόηση του τρόπου συντονισμού των διάφορων ενεργειών μιας ροής εργασίας. Στη μηχανική λογισμικού βασισμένη στους πράκτορες, οι προγραμματιστές δημιουργούν τα

προγράμματά τους ως ανεξάρτητοι πράκτορες λογισμικού, χωρίς να λαμβάνονται υπόψη σημαντικοί παράμετροι, όπως η ύπαρξη άλλων προγραμμάτων, η γνώση για τη ρύθμιση του υλικού με την οποία τα προγράμματα εκτελούνται κ.α.

Ο οργανισμός Workflow Management Coalition [] επιβάλλει τρεις απαιτήσεις στο περιβάλλον των πρακτόρων, οι οποίοι και είναι :

1. Πρέπει να υπάρχει μια κοινή αντίληψη των βασικών εννοιών του πεδίου εφαρμογής. Όλη η επικοινωνία ανάμεσα στους πράκτορες πρέπει να βασίζεται σε αυτή την κοινή αντίληψη.
2. Το σύστημα πρέπει να υποστηρίζει ένα βασικό ρεπερτόριο διαπρακτορικής επικοινωνίας.
3. Η αλληλεπίδραση ανάμεσα στους πράκτορες πρέπει να συντονιστεί, κάθε φορά που θα πρέπει να εκπληρώσουν από κοινού μια ομαδική ενέργεια.

Η ανάγκη για διασφάλιση κοινής αντίληψης της διαμοιραζόμενης πληροφορίας αντιμετωπίζεται με την ανάπτυξη αφενός των οντολογιών, που είναι ανεξάρτητες του έργου (task-independent) και αναπαριστούν τα αντικείμενα του πεδίου εφαρμογής, και αφετέρου των εργασιακών οντολογιών (task ontologies) που είναι ανεξάρτητες του πεδίου εφαρμογής και περιγράφουν μεθόδους λύσης προβλημάτων με ένα γενικότερο τρόπο. Η ανάγκη για κοινή σύλληψη της διαπρακτορικής επικοινωνίας αντιμετωπίζεται με τον ορισμό μιας γλώσσας επικοινωνίας πρακτόρων, που περιγράφει ένα ουσιώδες σύνολο από επικοινωνιακές δράσεις. Ο συντονισμός των πρακτόρων υλοποιείται με μια αρχιτεκτονική ένωσης σε ομοσπονδία (federation), με την οποία οι πράκτορες λύσεως προβλημάτων αλληλεπιδρούν μεταξύ τους μέσα από υπηρεσίες εξειδικευμένων πρακτόρων, οι οποίοι και ονομάζονται διευκολυντές (facilitators).

3.7.2. Η Γενική λύση

Σε περίπτωση που έχουμε διαφορετικές υπηρεσίες πράξης (enactment services), που υπακούουν στο Μοντέλο Αναφοράς Ροών Εργασίας (Workflow Reference Model) του WFMC και δεν βασίζονται στην τεχνολογία πρακτόρων ή σε διαφορετικές εφαρμογές ροών εργασίας, τότε ο WFMC προτείνει να υπάρχουν δύο σημαντικές όψεις για την επιτυχία της διαλειτουργικότητας ροών εργασίας :

1. Το εύρος του ποια κοινή ερμηνεία του ορισμού διαδικασίας (ή ενός υποσυνόλου της) είναι αναγκαία, παρουσιάζοντας παράλληλα δυνατότητα επίτευξης.
2. Υποστήριξη από το run-time και για την ανταλλαγή διαφόρων ειδών πληροφορίας ελέγχου και για τη μεταφορά δεδομένων, που σχετίζονται είτε με την εφαρμογή είτε με τη ροή εργασίας μεταξύ των ποικίλων υπηρεσιών πράξης.

3.7.2.1. Χρήση Ορισμού Διαδικασίας Κατά Μήκος Πολλαπλών Πεδίων Εφαρμογής

Είναι σημαντικό όταν και οι δύο υπηρεσίες πράξης είναι σε θέση να ερμηνεύσουν ένα κοινό ορισμό διαδικασίας, ο οποίος για παράδειγμα έχει παραχθεί από ένα κοινό εργαλείο κατασκευής (build tool). Τότε και τα δύο περιβάλλοντα καθιστώνται ικανά να μοιραστούν μια μοναδική όψη των αντικειμένων του ορισμού

της διαδικασίας, καθώς και τις ιδιότητές αυτών. Επίσης, οι ανεξάρτητες μηχανές ροών εργασίας καθιστώνται ικανές να μεταφέρουν την εκτέλεση των ενεργειών ή των υπό-διαδικασιών σε ετερογενής μηχανές ροών εργασίας, μέσα από το πλαίσιο ενός κοινού μοντέλου ονομασίας και αντικειμένων (naming and object model).

Όταν αυτή η κοινή επιδίωξη του ορισμού διαδικασίας δεν είναι εφικτή, τότε η εναλλακτική προσέγγιση της ‘έξαγωγής’ λεπτομερειών ενός υποσυνόλου του ορισμού διαδικασίας, ως μέρος της runtime ανταλλαγής, μπορεί να καθίσταται πιθανή.

Επιπλέον, όταν η ανταλλαγή ορισμού διαδικασίας, μέσω και των δύο παραπάνω προσεγγίσεων, δεν είναι εφικτή, η διαλειτουργικότητα περιορίζεται σε μια προσέγγιση ανοίγματος πύλης (gateway approach), στην οποία τα ονόματα αντικειμένων και ιδιοτήτων αντιστοιχίζονται μεταξύ των δύο περιβαλλόντων με την ύπαρξη μιας διαδικτυακής πύλης της εφαρμογής. Σε αυτήν την περίπτωση, οι δύο διαφορετικές υπηρεσίες πράξης χρησιμοποιούν τις δικές τους φόρμες ορισμού διαδικασίας και η πύλη χειρίζεται οποιαδήποτε αντιστοίχιση μεταξύ των δύο.

3.7.2.2 Αλληλεπιδράσεις Runtime Ελέγχου

Ο WFMC ορίζει την Προγραμματιστική Διεπαφή Πελατειακών Εφαρμογών Ροών Εργασίας (Workflow Client Application Programming Interface – WAPI), προκειμένου να επιτρέψει την υλοποίηση των εφαρμογών εμπρόσθιων άκρων (front-end), οι οποίες και χρειάζονται για την πρόσβαση στις λειτουργίες της μηχανής διαχείρισης ροών εργασίας (υπηρεσίες ροών εργασίας). Κατά το χρόνο εκτέλεσης (runtime), τα καλέσματα του WAPI χρησιμοποιούνται για τη μεταφορά του ελέγχου μεταξύ των υπηρεσιών ροών εργασίας, έτσι ώστε να ενεργοποιηθούν υπό-διαδικασίες ή ανεξάρτητες ενέργειες σε μία συγκεκριμένη υπηρεσία. Όταν και οι δύο υπηρεσίες υποστηρίζουν ταυτόχρονα ένα κοινό επίπεδο από καλέσματα του WAPI και μια κοινή όψη των αντικειμένων του ορισμού διαδικασίας (περιλαμβανομένων των ονοματικών συμβάσεων και των δεδομένων, που σχετίζονται με την εφαρμογή ή με τη ροή εργασίας), τότε το παραπάνω γεγονός θα γίνεται άμεσα από τις μηχανές ροών εργασίας – αν και αυτό θα απαιτήσει συμφωνία για ένα κοινό πρωτόκολλο υποστήριξης των στοιχειωδών WAPI μερών.

Όταν η προηγούμενη προϋπόθεση δεν είναι δυνατή, τα καλέσματα του WAPI μπορούν να χρησιμοποιηθούν για τη κατασκευή μιας υπηρεσίας πύλης (gateway function), παρέχοντας με αυτό το τρόπο διαλειτουργικότητα μεταξύ των δύο υπηρεσιών ροών εργασίας, η οποία και επιτυγχάνεται αρχικά με την αντιστοίχιση των διαφορετικών αντικειμένων και όψεων δεδομένων μεταξύ των δύο περιβαλλόντων, και ακολούθως με την υποστήριξη διαφορετικών πρωτοκόλλων για περιβάλλοντα σε κάθε υπηρεσία ροών εργασίας.

4. Το Σύστημα ARION

4.1 Εισαγωγή

Μεγάλα κομμάτια της πολύτιμης επιστημονικής κληρονομιάς υπάρχουν σε συλλογές (αποθήκες), οι οποίες αποθηκεύουν επιστημονικούς πόρους, όπως είναι τα σύνολα δεδομένων (data sets), τα προγράμματα και τα εργαλεία που σχετίζονται με επιστημονικά έργα. Τα σύνολα δεδομένων μπορεί να είναι δεδομένα, που προκύπτουν ή από μετρήσεις ή από κάποιους άλλους επιστημονικούς υπολογισμούς. Τα προγράμματα είναι συνήθως υλοποιήσεις μοντέλων προσομοίωσης / πρόβλεψης, ενώ τα εργαλεία αποτελούν είδη προγραμμάτων για (στατιστική, γεωγραφική, κ.τ.λ.) παρουσίαση και οπτικοποίηση δεδομένων.

Εδώ και αρκετό καιρό, τα επιστημονικά δεδομένα και προγράμματα χρησιμοποιούνται ως ‘ιδιωτικοί’ πόροι, που χρησιμοποιούνται μόνο από τους ανθρώπους / οργανισμούς που τα δημιούργησαν / ανέπτυξαν. Όμως, αυτή η ‘ιδιαίτερη ιδιοκτησία’ είναι, συνήθως, μια κατάσταση που δημιουργείται μάλλον από απραξία, παρά από οποιαδήποτε πολιτική περιορισμού της χρήσης των δεδομένων / πόρων. Τα δεδομένα και τα μοντέλα συλλέγονται / αναπτύσσονται μόνο ως μέρος μιας επιστημονικής μελέτης, αλλά πριν από τη μελέτη αυτή δεν είναι, εκ των προτέρων, έκδηλος ο τρόπος χειρισμού και αξιοποίησης τους. Με την απουσία μιας κατάλληλης υποδομής, πολύτιμα και μοναδικά επιστημονικά δεδομένα και μοντέλα ‘χάνονται’, διότι παρακρατούνται σε αποθήκες προσβάσιμες μόνο από το τοπικό δίκτυο και χωρίς να ιεραρχούνται μέσω μιας τυπικής μεθόδου καταλογισμού.

Διάφορες τεχνικές, βασισμένες στο δίκτυο, έχουν αναπτυχθεί για να κάνουν αυτούς τους επιστημονικούς πόρους διαθέσιμους στην ευρύτερη επιστημονική κοινότητα και, άρα, να βελτιώσουν την υπάρχουσα κατάσταση. Όμως, ακόμη και συστήματα, που είναι στη κορυφή της τεχνολογίας, τυπικά, παρουσιάζουν τέσσερις βασικά μειονεκτήματα, τα οποία και τα καθιστούν μη ελκυστικά τόσο στους προμηθευτές (επιστημονικών) πόρων όσο και στους χρήστες. Πρώτον, η διαδικασία εξαγωγής πόρων παραμένει αρκετά πολύπλοκη, περιλαμβάνοντας προγραμματιστική προσπάθεια και εξειδίκευση από τους προμηθευτές δεδομένων, οι οποίοι δεν μπορούν να αντεπεξέλθουν. Δεύτερον, προσφέρεται στους χρήστες μια απλοποιημένη ‘διεπαφή χρήση αναζήτησης πληροφορίας’, με πολύ μικρή καθοδήγηση για το πώς θα βρεθεί ή θα δημιουργηθεί η συγκεκριμένη πληροφορία. Τρίτον, μόλις βρεθεί ο πόρος, υπάρχει πολύ μικρή υποστήριξη για εύκαμπτη χρήση, δηλαδή κάποιος ή μπορεί να πάρει / χρησιμοποιήσει τον πόρο όπως ακριβώς είναι ή δεν μπορεί. Επομένως, ο δυναμικός συνδυασμός διάφορων πηγών, που ανήκουν σε διαφορετικούς προμηθευτές για τη δημιουργία νέων πηγών, είναι ουσιαστικά αδύνατος. Τελευταίο, αλλά εξίσου σημαντικό, είναι ότι οι τρέχουσες επιλύσεις δεν δουλεύουν με τις υπάρχουσες πρακτικές και μεθόδους χρηματοδότησης, που χρησιμοποιούνται στους οργανισμούς που παράγουν δεδομένα. Άρα, οι εν λόγω επιλύσεις μπορούν να θεωρηθούν ως ‘βάρος’ παρά ως ‘βοήθεια’.

Ο στόχος του **ARION** είναι να ξεπεράσει την τρέχουσα έλλειψη πρόσβασης στην κληρονομιά των συλλογών επιστημονικών δεδομένων και να προσφέρει μια νέα προσέγγιση στην προώθηση της επανάχρησής τους, η οποία επανάχρηση προς το παρόν δεν υπάρχει. Θα παρέχει μια ‘ελαφριού βάρους’ (lightweight) αρχιτεκτονική, που θα βασίζεται στη τεχνολογία του διαδικτύου, η οποία μάλιστα θα παραδίδει στους επιστήμονες και στους πολίτες μεγαλύτερη πρόσβαση στη παραγωγή χρήσιμης

πληροφορίας από ιστορικά δεδομένα και συλλογές μοντέλων. Τόσο τα δεδομένα όσο και οι συλλογές μοντέλων ακολουθούν διαφορετικές τεχνικές αρχές.

4.2. Αντικειμενικοί στόχοι του ARION

Το κύριο χαρακτηριστικό του **ARION** είναι ότι επιτρέπει τη δυναμική παραγωγή επιστημονικής πληροφορίας, πράγμα που δεν έχει προηγηθεί στο πλαίσιο μιας ψηφιακής βιβλιοθήκης. Αντίθετα, με τα παραδοσιακά ψηφιακά αντικείμενα, τα επιστημονικά αντικείμενα είναι πολύπλοκοι υποψήφιοι για επαναχρησιμοποίηση, εκτός και εάν ακριβής πληροφορία για τη χρήση τους αποθηκευθεί μέσα σε Ψηφιακή Βιβλιοθήκη. Επομένως, το σύστημα **ARION** δεν είναι μόνο μια Ψηφιακή Βιβλιοθήκη, αλλά και ένα σύστημα διαχείρισης ροών εργασίας. Ταυτόχρονα, είναι και ένα σύστημα πιστοποίησης και εξουσιοδότησης χρήστη, εφόσον θα πρέπει να διασφαλίζει ποιός και πότε θα έχει πρόσβαση στους προσφερόμενους επιστημονικούς πόρους από ανάλογους οργανισμούς. Συνεπώς, το σύστημα **ARION** χωρίζεται σε τρία υποσυστήματα, λόγω των τριών βασικών και προαναφερθέντων λειτουργιών του. Τα υποσυστήματα είναι τα ακόλουθα :

1. Αποθήκευση των επιστημονικών πόρων, με ταχεία αναζήτηση και πρόσβαση σε αυτούς.
2. Εάν κάποιος επιστημονικός πόρος δεν ευρεθεί στο σύστημα, αλλά υπάρχει περιγραφή για το πώς θα παραχθεί, τότε θα παράγεται δυναμικά στο πλαίσιο αυτής της περιγραφής.
3. Πιστοποίηση και εξουσιοδότηση χρηστών, που επιθυμούν πρόσβαση σε επιστημονικού πόρους.

Μπορεί να τεθεί το ζήτημα ότι το σύστημα **ARION** είναι απλώς μια Ψηφιακή Βιβλιοθήκη, που προσφέρει μια επιπλέον λειτουργία: τη δυναμική παραγωγή νέων δεδομένων. Μπορεί, επίσης, να τεθεί το ζήτημα ότι το σύστημα **ARION** είναι ένα σύστημα διαχείρισης ροών εργασίας, που μοντελοποιεί-προσομοιώνει τον τρόπο λειτουργίας μιας Ψηφιακής Βιβλιοθήκης μέσω κατάλληλων διαδικασιών στις οποίες λαμβάνει μέρος και ο χρήστης. Με άλλα λόγια, το σύστημα **ARION**, ως ένα σύστημα διαχείρισης ροών εργασίας, αυτοματοποιεί με συγκεκριμένο τρόπο την λειτουργία επιστημονικών οργανισμών, όσον αφορά όλες τις πιθανές αρμοδιότητες-λειτουργίες που περιλαμβάνουν όπως αποθήκευση, αναζήτηση, παραγωγή επιστημονικών δεδομένων. Αναμφισβήτητα λοιπόν, το σύστημα **ARION** προσφέρει συγκεκριμένες και βασικές λειτουργίες στους χρήστες και για αυτό κρίθηκε σκόπιμο να υπάρχουν τρία ανεξάρτητα υπο-συστήματα, που θα λειτουργούν αυτόνομα, προσφέροντας λειτουργίες-υπηρεσίες είτε το ένα στο άλλο είτε σε ένα τέταρτο μέρος-συντονιστή του συστήματος. Αυτό το τέταρτο μέρος δεν είναι τίποτε άλλο από τις διεπαφές χρήσης, που θα συντονίζουν τα τρία υποσυστήματα έτσι ώστε να προσφέρουν τις τελικές και ενοποιημένες λειτουργίες του ολικού συστήματος όταν το ζητήσει ο χρήστης, ο οποίος και θα καθοδηγείται οπτικά από αυτές τις διεπαφές. Με αυτόν τον τρόπο, η αρχιτεκτονική του συστήματος μας θα είναι διαρθρωτική (modular) και λογικά κατανοητή.

Ένα σημαντικό χαρακτηριστικό, που προέκυψε από την ανάλυση απαιτήσεων είναι ότι το σύστημα **ARION** μπορεί να λειτουργεί αυτόνομα και τοπικά σε ένα προμηθευτή επιστημονικών αντικειμένων εκτός από κεντρικά. Δηλαδή, θα υπάρχουν δύο είδη του συστήματος **ARION**. Το κεντρικό θα παρέχει όλες τις λειτουργίες και θα διαχειρίζεται όλα τα σύνολα δεδομένων και τα προγράμματα. Αντίθετα, το τοπικό,

ανάλογα με την ρύθμιση, που επιθυμεί ο κάθε προμηθευτής, θα παρέχει ένα υποσύνολο όλων των λειτουργιών και θα διαχειρίζεται μόνο τα σύνολα δεδομένων και προγράμματα αυτού του προμηθευτή. Συνεπώς, ο κάθε προμηθευτής θα έχει ένα τοπικό σύστημα **ARION**, ανάλογα με το ποιο κομμάτι-υποσύστημα θέλει να λειτουργεί και ποιοί υπολογιστικοί πόροι θα του αυτοματοποιούν την δουλειά του, ενώ, παράλληλα, θα διοχετεύει στην αγορά όλους τους πόρους του μέσω του κεντρικού συστήματος του **ARION**.

Κάθε υποσύστημα του **ARION** θα προσφέρει τις βασικές λειτουργίες, άλλες επιμέρους που θα είναι χρήσιμα κομμάτια των βασικών, καθώς και ορισμένα τεχνικά χαρακτηριστικά. Το σύνολο των λειτουργιών και των τεχνικών χαρακτηριστικών του κάθε υποσυστήματος θα υπακούει στις απαιτήσεις του συστήματος και των χρηστών. Το κάθε υποσύστημα θα πρέπει, επίσης, να παρουσιάζει συγκεκριμένα αρχιτεκτονικά χαρακτηριστικά, όπως κλιμακωσιμότητα (scalability) και κατανομή (distribution), τα οποία επαγωγικά θα ανάγονται ως χαρακτηριστικά του όλου συστήματος.

Στις επόμενες υπο-ενότητες θα αναλύσουμε τις λειτουργικές και τεχνικές απαιτήσεις του κάθε υποσυστήματος και θα περιγράψουμε τα βασικά αρχιτεκτονικά χαρακτηριστικά, που θεωρητικά θα πρέπει να είναι εφοδιασμένα.

4.2.1. Λειτουργίες και τεχνικές της Ψηφιακής Βιβλιοθήκης

Οι βασικές λειτουργίες της Ψηφιακής Βιβλιοθήκης είναι δύο :

1. Αποθήκευση επιστημονικών πόρων.
2. Γρήγορη εύρεση αυτών.

Εκτός από τις βασικές λειτουργίες, η Ψηφιακή Βιβλιοθήκη θα πρέπει να παρουσιάζει μερικές άλλες λειτουργίες, πάνω στις οποίες στηρίζουν την ύπαρξή τους τόσο οι βασικές λειτουργίες όσο και τα τεχνικά χαρακτηριστικά, που αναλύουμε παρακάτω :

- Θα πρέπει να αναπτυχθεί μια οντολογία ή και περισσότερες, ανάλογα με την τεχνική σχεδίασης, που θα καταγράφει τη γνώση του πεδίου εφαρμογής και θα χωρίζει τα μετα-δεδομένα της Ψηφιακής Βιβλιοθήκης, ιεραρχικά, σε κατηγορίες-κλάσεις. Με αυτό τον τρόπο, κάθε επιστημονικός πόρος θα είναι μια οντότητα, που θα ανήκει σε μια κατηγορία ή σε πολλές κατηγορίες μετα-δεδομένων / ιδιοτήτων, ανάλογα με τη φύση του. Επιπλέον, ορισμένες κατηγορίες μετα-δεδομένων / ιδιοτήτων μπορούν να γίνουν είτε πρότυπα είτε να αντιστοιχισθούν με πρότυπα μετα-δεδομένων. Πρέπει να σημειωθεί ότι οι οντολογίες, αποδεδειγμένα, βελτιώνουν το χρόνο αναζήτησης πόρων, εφόσον βέβαια σχεδιαστούν σωστά. Τέλος, οι οντολογίες θα βοηθήσουν στη σχεδίαση και λειτουργία των διεπαφών χρήσης, ώστε να καθοδηγούν διαισθητικά τον χρήστη στην εύρεση της σωστής πληροφορίας για αυτόν.
- Θα πρέπει να γίνεται σωστή διαχείριση των μετα-δεδομένων. Δηλαδή, κάθε φορά που εισάγεται ή διαγράφεται ένας πόρος στο σύστημα, θα πρέπει να υπάρχει ένα εργαλείο τόσο δημιουργίας των σωστών μετα-δεδομένων για αυτόν τον πόρο και βέλτιστης αποθήκευσης τους στο σύστημα (στη περίπτωση εισαγωγής) όσο και ένα εργαλείο-μηχανή αναζήτησης, που θα βρίσκει τα μετα-δεδομένα για ένα πόρο και θα τα διαγράφει από το σύστημα (στη περίπτωση διαγραφής) ή θα τα παρουσιάζει στο χρήστη (στη περίπτωση αναζήτησης). Πρέπει να σημειώσουμε ότι στη θέση των εργαλείων ή πίσω από αυτά κρύβονται ή τοποθετούνται κάποιες επιμέρους λειτουργίες.

- Η μηχανή αναζήτησης δεν θα πρέπει να απαντά μόνο σε ερωτήσεις, που αφορούν μετα-δεδομένα, αλλά θα πρέπει να απαντά και σε ερωτήσεις, που αφορούν τις κατηγορίες μετα-δεδομένων. Δηλαδή, να απαντά σε όλες τις ερωτήσεις, που σχετίζονται με το σχήμα της οντολογίας.
- Σε περίπτωση που δεν ευρεθεί ένας επιστημονικός πόρος/δεδομένο, θα πρέπει είτε η μηχανή αναζήτησης είτε η διεπαφή χρήσης, που έκανε την ερώτηση, να επικοινωνήσει με το σύστημα διαχείρισης ροών εργασίας, προκειμένου να ληφθεί η πληροφορία για τον εάν αυτό το δεδομένο μπορεί να παραχθεί δυναμικά.

4.2.2 Λειτουργίες και τεχνικές του συστήματος διαχείρισης ροών εργασίας

Ένα σύστημα διαχείρισης ροών εργασίας θα πρέπει να εμφανίζει τις εξής λειτουργίες και τεχνικά χαρακτηριστικά :

- Θα πρέπει να απαντάει στην ερώτηση αν κάποιο επιστημονικό δεδομένο, που ανήκει σε κλάσεις της οντολογίας / των οντολογιών του συστήματος μπορεί να παραχθεί δυναμικά.
- Θα πρέπει αφενός να παράγει δυναμικά επιστημονικά δεδομένα και μετα-δεδομένα για αυτά τα δεδομένα και αφετέρου να επικοινωνεί μέσω λειτουργιών ή εργαλείων με τη Ψηφιακή Βιβλιοθήκη, προκειμένου να αποθηκεύσει αυτά τα μετα-δεδομένα. Επίσης, θα πρέπει να αποθηκεύει στατιστικά στοιχεία μετά από κάθε παραγωγή δεδομένου, τα οποία και θα παρουσιάζει στο χρήστη κάποια διεπαφή χρήσης.
- Θα πρέπει να επιτρέπει την περιγραφή του επακριβούς τρόπου παραγωγής ενός δεδομένου (δηλαδή της ροής εργασίας) και να την αποθηκεύει. Η περιγραφή του τρόπου παραγωγής, συνήθως, γίνεται μέσω μιας γλώσσας περιγραφής ροών εργασίας ή μέσω ενός γραφικού κειμενογράφου ροών εργασίας, ενώ η αποθήκευση, συχνά, γίνεται με τη μορφή αρχείων, που αποθηκεύονται στο λειτουργικό σύστημα με ευφυή τρόπο ή/και με τη χρήση μιας βάσης δεδομένων.
- Θα πρέπει να εξασφαλίζει ότι οι συνδυασμοί προγραμμάτων και δεδομένων είναι εφικτοί τόσο από πλευράς διαθεσιμότητας και των δύο ειδών των επιστημονικών πόρων όσο και από πλευράς εξουσιοδοτημένης πρόσβασης του χρήστη σε αυτούς τους πόρους. Άρα, το σύστημα θα πρέπει να επικοινωνεί τόσο με τη Ψηφιακή Βιβλιοθήκη, για να εντοπίσει τους χρήστες που έχουν πρόσβαση σε κάποιο δεδομένο ή σε κάποιο πρόγραμμα, όσο και με το σύστημα ταυτοποίησης και εξουσιοδότησης, για να αποκαλύψει αν ένας ταυτοποιημένος χρήστης έχει εξουσιοδότηση να έχει πρόσβαση στους προαναφερόμενους πόρους.
- Θα πρέπει να δίνει στο χρήστη τη δυνατότητα να εποπτεύει οπτικά, αλλά και να επεμβαίνει στη ροή της εργασίας σε επίπεδο έργων (tasks) ή συνολικά (δηλαδή να την σταματήσει, να την αναστείλει κ.τ.λ.) ή όπου χρειάζεται να δοθούν παράμετροι στα προγράμματα που τρέχουν. Όλα αυτά θα γίνονται μέσω κατάλληλων οπτικών εργαλείων.
- Θα πρέπει να υποστηρίζει τη διαλειτουργικότητα των ροών εργασίας.
- Θα πρέπει να υποστηρίζει την εκτέλεση των ροών εργασίας ως δοσοληψίες, ώστε όλες οι περιγραφές ροών εργασίας που δίδει ο χρήστης για εκτέλεση, να εκτελεστούν κάποια στιγμή είτε επιτυχημένα είτε όχι, αλλά πάντοτε χωρίς να επηρεάζονται από τα διάφορα προβλήματα που πιθανώς να παρουσιάζονται στο διαδίκτυο.

4.2.3. Λειτουργίες και τεχνικές του συστήματος ταυτοποίησης και εξουσιοδότησης

Το σύστημα ταυτοποίησης και εξουσιοδότησης χρηστών, εκτός από τις βασικές λειτουργίες του που είναι :

- Ταυτοποίηση χρήστη.
- Εξουσιοδότηση κατάλληλου χρήστη.

θα πρέπει να προσφέρει και τις παρακάτω λειτουργίες :

- Δυνατότητα-λειτουργία δημιουργίας νέου χρήστη. Στο νέο χρήστη θα πρέπει να παραχωρούνται και κατάλληλα δικαιώματα στο σύστημα ανάλογα με την δικαιοδοσία του.
- Δυνατότητα-λειτουργία αλλαγής των πληροφοριών ταυτοποίησης και εξουσιοδότησης ενός χρήστη.
- Λειτουργία παραχώρησης της ιεραρχίας των ρόλων – δικαιωμάτων του συστήματος στα διάφορα υποσυστήματά του.

4.2.4. Βασικά αρχιτεκτονικά χαρακτηριστικά όλων των υποσυστημάτων

Τα βασικά αρχιτεκτονικά χαρακτηριστικά του κάθε υποσυστήματος, τα οποία και εάν ισχύουν σε κάθε ένα από τα υποσυστήματα, θα ισχύουν και για το ολικό σύστημα του **ARION**, είναι :

- Να είναι ανοικτό (open).
- Να είναι κατανεμημένο.
- Να παρουσιάζει κλιμακωσιμότητα.
- Να είναι αξιόπιστο.
- Να είναι ασφαλές τόσο όσον αφορά την εσωτερική επικοινωνία με τα διάφορα κομμάτια του όσο και όσον αφορά την εξωτερική του επικοινωνία με τα άλλα υποσυστήματα.
- Να είναι lightweight.
- Να εξασφαλίζει τη λογική και φυσική κατανομή των λειτουργιών του, αλλά και την κατανομή φόρτου στους κόμβους – υπολογιστές που καταλαμβάνει.
- Να χρειάζεται ελάχιστη διοίκηση τόσο κατά την εγκατάσταση όσο και για τη λειτουργία και συντήρησή του.

4.3. Σχεδίαση του συστήματος ARION

4.3.1. Εισαγωγή

Η σχεδίαση του συστήματος **ARION** έγινε προσεκτικά και με μόνο οδηγό την ικανοποίηση των αντικειμενικών του στόχων. Η διαδικασία της σχεδίασης χωρίστηκε σε τέσσερα ανεξάρτητα και παράλληλα κομμάτια. Με αυτό τον τρόπο, ελαχιστοποιήθηκε κατά πολύ ο χρόνος σχεδίασης του συστήματος. Το πρώτο κομμάτι αφορά την τεχνική σχεδίασης, που θα ακολουθηθεί για τη δημιουργία οντολογίας / οντολογιών για το σύστημα **ARION**. Επίσης, στο πρώτο κομμάτι αναφέρεται και το τελικό αποτέλεσμα της σχεδίασης που έγινε. Στο δεύτερο κομμάτι

αναφέρεται ο ορισμός του τι είναι η επιστημονική ροή εργασίας και η σχεδίαση / καθορισμός του μοντέλου της. Στο τρίτο κομμάτι αναφέρεται / αναλύεται η αρχιτεκτονική του συστήματος **ARION**. Τέλος, στο τέταρτο κομμάτι περιγράφονται ποιες είναι οι διεπαφές χρήσης του συστήματος και ποιος ο ρόλος τους στο σύστημα του **ARION**. Κάθε κομμάτι της σχεδίασης αναλύεται σε μια αντίστοιχη υπο-ενότητα αυτής της ενότητας.

4.3.2. Σχεδίαση Οντολογίας

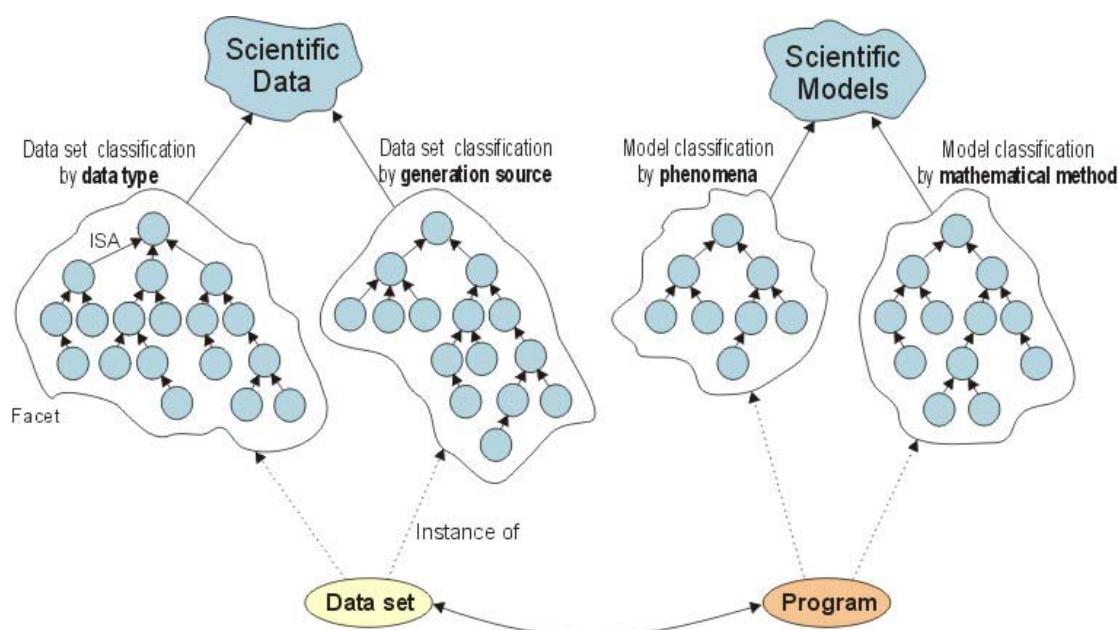
Ως μοντέλο οργάνωσης της γνώσης, η αρχιτεκτονική μιας οντολογίας εξαρτάται ουσιαστικά από το είδος της γνώσης, που θα πρέπει να αναπαρασταθεί και να γίνει σαφής. Μια ιδιαίτερα ζωτικής σημασίας αντίθεση είναι αυτή μεταξύ της ‘γενικής γνώσης’ και της ‘ορολογικής γνώσης’ (terminological knowledge) ή αλλιώς ‘της γνώσης που βασίζεται στο πεδίο’ (domain-specific knowledge). Αυτή η παράμετρος επιδρά στο βαθμό της πολυπλοκότητας της διαδικασίας της σχεδίασης οντολογιών, καθώς επίσης και στο βαθμό granularity του τύπου του συστήματος. Η ορολογική γνώση είναι συνήθως ομογενής και ρητά δομημένη, ενώ η γενική γνώση είναι ετερογενής και άρρητα ή πολύ χαλαρά δομημένη. Η γενική γνώση είναι ετερογενής, διότι αφενός διασταυρώνεται ουσιαστικά με πολλά πεδία (cross-domain) και αφετέρου είναι σε πολλές περιπτώσεις ανεξάρτητη από κάποια ιδιαίτερη λάξευση περιοχής (domain carving). Αυτό φέρνει στην επιφάνεια αρκετά πολύπλοκα ζητήματα στην επιλογή των τύπων, οι οποίοι και είναι ικανοί να ‘αιχμαλωτίσουν’ τη σχετική δομή της γνώσης με ένα βέλτιστο τρόπο. Αντίστροφα, η επιλογή των τύπων για μια οντολογία, που στοχεύουν σε συγκεκριμένη περιοχή της, μπορεί να εκμεταλλευτεί μια οργάνωση της εν λόγω περιοχής. Η οργάνωση αυτή ή διαμοιράζεται από τους αρμόδιους εμπειρογνώμονες ή αποσπάται εύκολα από την συνηθισμένη πρακτική.

Μια άλλη συσχετιζόμενη αντίθεση είναι αυτή μεταξύ των οντολογιών πολλών σκοπών (multi-purpose) και συγκεκριμένης χρήσης (specific-usage). Στη πραγματικότητα, η επιλογή της οντολογίας επηρεάζεται καθαρά από το είδος του στόχου της διαχείρισης γνώσης. Ένας συγκεκριμένος σκοπός ή εφαρμογή προκαταλαμβάνει την επιλογή ενός ιδιαίτερου συνόλου από τύπους, για να αναλύσει και να οργανώσει το πεδίο γνώσης, τονίζοντας τις συνδέσεις και τις τακτικότητες, που χρειάζονται για το δεδομένο σκοπό. Αντίθετα, η σχεδίαση μιας οντολογίας πολλαπλών σκοπών, ενώ υστερεί μιας σημαντικής καθοδήγησης που αναπαριστάται από περιορισμούς είτε της εφαρμογής είτε βασιζόμενους σε έργα, πρέπει να θεωρήσει τη στοιχειοθέτηση (versatility) του τύπου της αρχιτεκτονικής, ως ένα από τους πιο σημαντικούς στόχους προς επιτυχία.

Σήμερα, η ανάπτυξη οντολογιών, που ανταποκρίνονται επιτυχώς σε ιδιαίτερες περιοχές του συστήματος, αποτελεί κοινή πρακτική στην ανάπτυξη συστημάτων βασισμένων στο περιεχόμενο. Το πλεονέκτημα αυτών των αρκετά συγκεκριμένων οντολογιών είναι η αποδοτικότητα της αναπαράστασης τους, ενώ η σημαντικότερη αδυναμία τους είναι η σχεδόν ολοκληρωτική έλλειψη φορητότητας και ευκαμψίας κατά τις διασταυρώσεις περιοχών (cross-domain portability and flexibility). Η τελευταία αδυναμία είναι αναμενόμενη, αφού οι οντολογίες αναπαριστούν συγκεκριμένη γνώση.

Στο σύστημα **ARION**, γίνεται εστίαση στην κοινή σημασιολογία των επιστημονικών συλλογών. Σχεδιάζεται μια περιβαλλοντολογική (κυματικού κλίματος) οντολογία αποτελούμενη από συλλογή διαφορετικών πλευρών (facets). Παραδείγματος χάριν, οι πλευρές μπορεί να περιγράψουν σύνολα δεδομένων,

μεθόδους παραγωγής που περιλαμβάνουν τη μαθηματική μοντελοποίηση, παραμέτρους προς χρησιμοποίηση κ.α.. Ο λόγος για το συνδυασμό διάφορων πλευρών είναι η διάρθρωση μιας δυνητικά τεράστιας μονολιθικής οντολογίας. Επιπροσθέτως, οι πλευρές μπορεί να διατυπωθούν σύμφωνα με βασικές επερωτήσεις, που οι χρήστες ενδιαφέρονται να κάνουν-διατυπώσουν. Η μηχανική, που βασίζεται σε πλευρές (facet-based) μιας οντολογίας, κλιμακώνεται ομαλά με τεράστιες επιστημονικές οντολογίες. Νέες πληροφορίες μπορεί να προστεθούν σε συμφωνία με τις ανάγκες των χρηστών ή των προμηθευτών δεδομένων. Ο ορισμός της οντολογίας περιέχει μια ιεραρχία 'IS-A' συσχετιζόμενων εννοιών της περιοχής, σχέσεις μεταξύ εννοιών και ιδιότητες εννοιών. Υπάρχουν δύο κύριες οντότητες στην οντολογία του ARION, αποτελούμενες από διαφορετικές πλευρές, οι οποίες οντότητες περιγράφουν τα επιστημονικά δεδομένα και τα επιστημονικά μοντέλα αντίστοιχα. Αυτή η προσέγγιση παρέχει ακόμα ένα επίπεδο granularity. Η βασική δομή της οντολογίας για κύματα ωκεανού φαίνεται στο σχήμα 6.



Σχήμα 6 - Η δομή της οντολογίας για κυματικό κλίμα

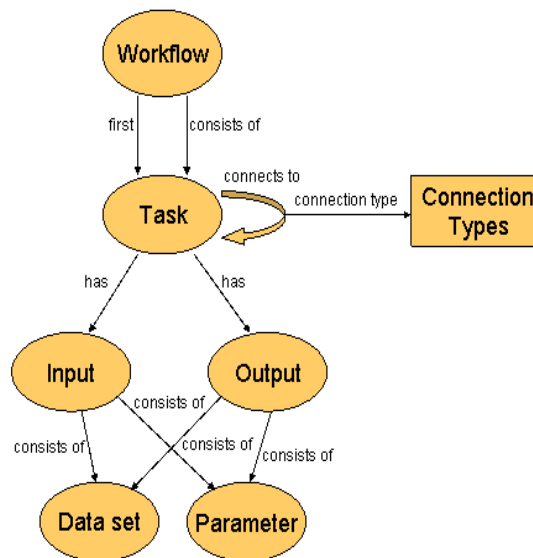
Η κατασκευή μιας γενικής οντολογίας για τη διαχείριση γνώσης επιτρέπει στην οντολογία να είναι εύκολα επαναχρησιμοποιούμενη. Επιπλέον, αυτοί που αναπτύσσουν την οντολογία δεν χρειάζεται να ακολουθούν πολύπλοκες και χρονοβόρες διαδικασίες, για να αναπροσαρμοστούν στους υπάρχοντες τύπους συστημάτων, αντιμετωπίζοντας τις νέες ανάγκες αναπαράστασης ή επεκτάσεις στην οντολογία. Επομένως, οι γενικές οντολογίες φαίνεται να προσφέρουν την προτυποποίηση και την ομοιομορφία, οι οποίες μπορούν να εγγυηθούν ένα υψηλό βαθμό διαμοιρασμού και αναχρησιμοποίησης τόσο της γνώσης όσο και των πόρων.

Στην περίπτωση του ARION, η οντολογία χρησιμοποιείται αφενός ως η βάση για τη σημασιολογική δόμηση και οργάνωση της αποθήκης της πληροφορίας των μετα-δεδομένων, και αφετέρου για να στηρίξει τη διατύπωση των επερωτήσεων. Η οντολογία χρησιμοποιείται με τρεις διαφορετικούς τρόπους :

1. Ως ένα εννοιολογικό πλαίσιο, με στόχο την υποστήριξη των χρηστών όσον αφορά την επιλογή τους για την αποθήκη της πληροφορίας των μετα-δεδομένων και τη διατύπωση των πιθανών επερωτήσεων.

2. Ως ένας οδηγός, για την κατανόηση των μετα-δεδομένων που οργανώνονται σύμφωνα με τις έννοιες της οντολογίας (ontology-driven metadata). Τα μετα-δεδομένα αυτά αποτελούν (περιγράφουν) τις ιδιότητες των εννοιών.
3. Ως ένας οδηγός στις διεπαφές χρήσης, για τη δημιουργία και τη βελτίωση των επερωτήσεων.

4.3.3. Σχεδίαση του μοντέλου της επιστημονικής ροής εργασίας



Σχήμα 7 - Το μοντέλο της επιστημονικής ροής εργασίας

Η επιστημονική εργασία-μελέτη του **ARION** μοντελοποιεί, περιγράφει, διαχειρίζεται και εκτελεί επιστημονικές ροές εργασίας. Αυτές δεν είναι τίποτε άλλο από υψηλά αυτοματοποιημένες και παραγωγικές ροές εργασίας, ενός απλού μοντέλου ροής εργασίας. Το μοντέλο, που εμφανίζεται στο σχήμα 7, εμπεριέχει εξ' ολοκλήρου το μοντέλο επιστημονικής διαδικασίας (δηλαδή μια επιστημονική ροή εργασίας ταυτίζεται με μια επιστημονική διεργασία) και ορίζει μια ροή εργασίας ως γράφο από αλληλοσυνδεόμενους κόμβους / ενέργειες που ονομάζονται έργα. Αυτά τα έργα είναι επιστημονικά προγράμματα, που εκτελούνται σύμφωνα με είσοδο δοσμένη από χρήστη / χρήστες πριν ή κατά την εκτέλεση της συγκεκριμένης ροής εργασίας, ενώ παράγουν και έξοδο. Η είσοδος αποτελείται από μηδέν ή περισσότερα σύνολα δεδομένων και από μηδέν ή περισσότερες παραμέτρους. Το ίδιο ισχύει και για την έξοδο. Ένα σύνολο δεδομένων περιέχει μετρήσεις ή αποτελέσματα μιας παρέκτασης (extrapolation). Η έξοδος ενός προγράμματος μπορεί να αποθηκευθεί στο σύστημα του **ARION** (ως ένα ενδιάμεσο ή τελικό αποτέλεσμα) ή / και να χρησιμοποιηθεί ως είσοδο στο επόμενο έργο προς εκτέλεση στη σειρά εκτέλεσης έργων. Τα έργα εκτελούνται είτε σειριακά είτε παράλληλα. Δύο έργα τρέχουν σειριακά, όταν από το πρώτο έργο φεύγει ένα βέλος που καταλήγει στο δεύτερο. Πολλά έργα τρέχουν

παράλληλα, όταν από κάποιο έργο φεύγουν βέλη προς αυτά τα έργα. Υπάρχουν δύο είδη σύνδεσης (βέλη): το απλό και το ειδικό. Το απλό είδος ήδη εξετάστηκε προηγουμένως. Το ειδικό είδος σύνδεσης επιτρέπει σε ένα τρέχων έργο, μετά την ολοκλήρωσή του, να δώσει τον έλεγχο της επανεκτέλεσης σε ένα (ή πολλά) έργα, το οποίο και έχει ήδη τρέξει, με τη σύμφωνη γνώμη του χρήστη (backtracking). Αυτό γίνεται όταν ο χρήστης δεν είναι ικανοποιημένος με τα ενδιάμεσα αποτελέσματα των έργων, που έχουν εκτελεστεί και θέλει κάποιο αρχικό έργο να επανα-εκτελεστεί με διαφορετική παράμετρο (παραμέτρους), έτσι ώστε να παραχθεί το επιθυμητό αποτέλεσμα στη συνέχεια της εκτέλεσης.

Κάθε χρήστης του **ARION** έχει διαφορετικά προνόμια στα αντικείμενα – πόρους του συστήματος. Αυτά τα αντικείμενα είναι είτε ροές εργασίας είτε προγράμματα είτε σύνολα δεδομένων. Μια ροή εργασίας προσδιορίζεται, εκ των προτέρων, από ένα χρήστη και αρχικοποιείται (initiated) από αυτόν. Ο ίδιος χρήστης έχει δικαίωμα να επεμβαίνει στην εκτέλεση αυτής της ροής εργασίας, κάθε φορά που χρειάζεται να δοθεί μια επιπλέον παράμετρος ή κάθε φορά που προκύπτουν σημεία ελέγχου της εκτέλεσης (backtrack points or arcs), που ρωτάνε τον χρήστη εάν θα πρέπει να γυρίσει πίσω και να επανα-εκτελέσει ένα έργο ή να συνεχίσει μπροστά με το επόμενο έργο προς εκτέλεση. Περισσότερα για τη σχεδίαση του μοντέλου της επιστημονικής ροής εργασίας θα αναφερθούν στο επόμενο κεφάλαιο, όπου και θα παρουσιαστεί μια γλώσσα ορισμού επιστημονικών ροών εργασίας, η οποία θα υπακούσει στο προαναφερόμενο μοντέλο.

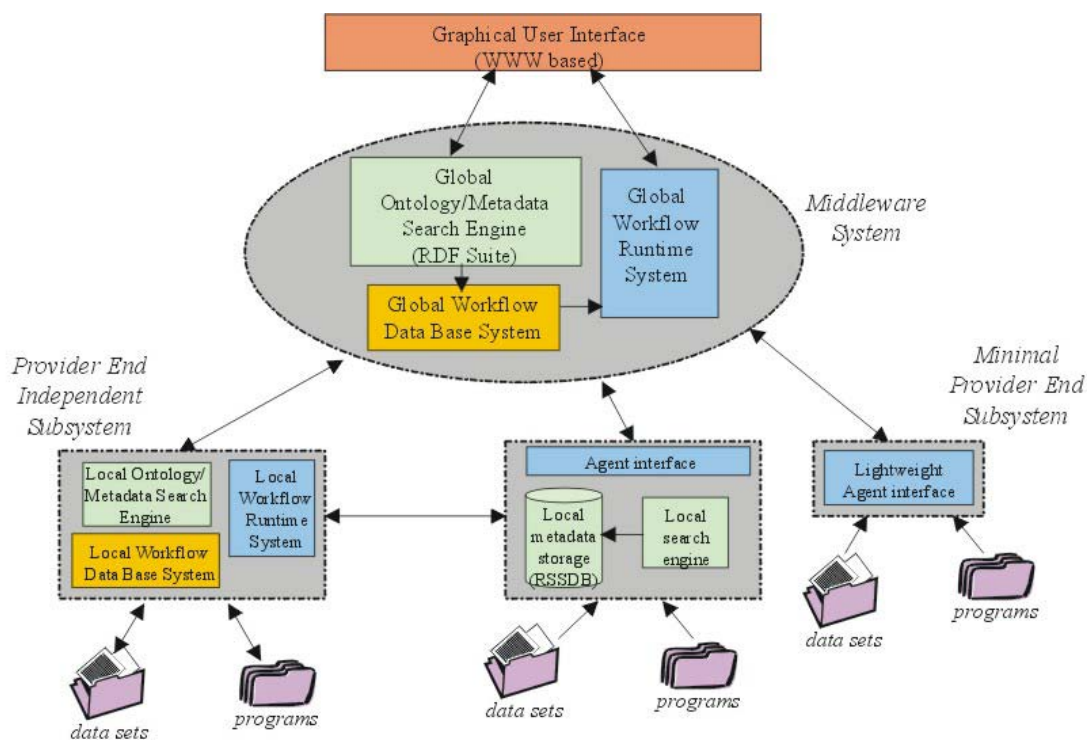
4.3.4. Η αρχιτεκτονική του συστήματος ARION

Αρχικά και σύμφωνα με τη φάση της ανάλυσης των απαιτήσεων, η αρχιτεκτονική του συστήματος **ARION** περιλάμβανε τρία ανεξάρτητα μέρη – υποσυστήματα: τη Ψηφιακή Βιβλιοθήκη, το σύστημα διαχείρισης ροών εργασίας και το σύστημα ταυτοποίησης και εξουσιοδότησης χρήστη. Τελικά, η αρχιτεκτονική του συστήματος ορίστηκε ως ακολούθως :

- Πρώτον, το υποσύστημα Ψηφιακής Βιβλιοθήκης να μετονομαστεί σε υποσύστημα Μηχανής Αναζήτησης, διότι ουσιαστικά παρέχει ακριβώς όλες τις λειτουργίες που συνθέτουν την έννοια του συστήματος μιας μηχανής αναζήτησης. Η Ψηφιακή Βιβλιοθήκη, δηλαδή, είναι γενικότερη έννοια, που περιλαμβάνει και την έννοια της Μηχανής Αναζήτησης.
- Δεύτερον, το σύστημα διαχείρισης ροών εργασίας να χωριστεί σε δύο υποσυστήματα: στο υποσύστημα βάσεων δεδομένων ροών εργασίας και στο υποσύστημα εκτέλεσης ροών εργασίας. Αυτό έγινε αφενός, για να διαχωριστούν οι πολλές λειτουργίες και τεχνικές που παρουσιάζει ένα σύστημα διαχείρισης ροών εργασίας, και αφετέρου, για να χωριστεί η σχεδίαση και η υλοποίηση σε δύο κομμάτια που θα αναληφθούν από διαφορετικά άτομα, ώστε να μειωθεί ο χρόνος κατασκευής του υποσυστήματος περίπου στο μισό. Τα δύο αυτά μέρη αντιπροσωπεύουν, αντιστοίχως, τα μέρη κατασκευής (build-time) και εκτέλεσης (run-time) ενός συστήματος διαχείρισης ροών εργασίας.

Κατά συνέπεια, το σύστημα του **ARION** αποτελείται από τέσσερα ανεξάρτητα μέρη. Η Μηχανή Αναζήτησης (Search Engine) επιτρέπει στους χρήστες να θέτουν επερωτήσεις στη γνώση του πεδίου του **ARION**. Το Σύστημα Καταναμημένης Αποθήκευσης Ροών Εργασίας Επιστημονικών Αντικειμένων (System of Distributed Workflow Storage of Scientific Objects) περιέχει τους ορισμούς ροών εργασίας και

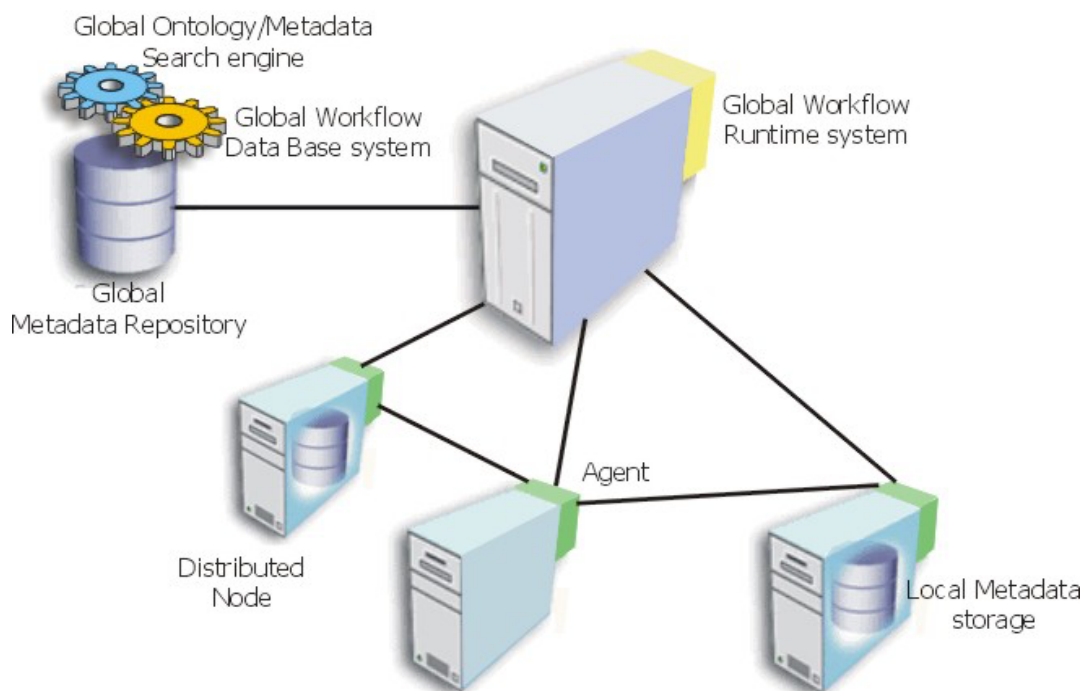
χειρίζεται την ετοιμασία των εκτελεστικών ορισμών ροών εργασίας, που θα σταλούν προς εκτέλεση στο σύστημα εκτέλεσης ροών εργασίας. Το Σύστημα Εκτέλεσης Ροών Εργασίας (Workflow Runtime System) είναι υπεύθυνο για την εκτέλεση των ροών εργασίας και τη διαχείριση της πληροφορίας, που παράγεται από τους καταναμημένους κόμβους κατά τη διάρκεια αυτής της εκτέλεσης. Το Σύστημα Ταυτοποίησης και Εξουσιοδότησης Χρηστών (User Authentication and Authorization System) είναι υπεύθυνο για την ταυτοποίηση και την εξουσιοδότηση ταυτοποιημένων χρηστών. Η αρχιτεκτονική του συστήματος φαίνεται στο σχήμα 8, όπου το κομμάτι των Διεπαφών Χρήσης (Graphical User Interface) συμβολίζει όλες τις διεπαφές χρήσης του διαδικτύου, που θα καθοδηγούν το χρήστη στη σωστή και ταχεία πρόσβαση της πληροφορίας, η οποία και τον ενδιαφέρει. Και τα τέσσερα μέρη – υποσυστήματα του **ARION** συνδέονται εννοιολογικά τόσο από μια συγκεκριμένη οντολογία, όσο και από τις αντίστοιχες ροές εργασίας, δηλαδή τις συνολικές δομικές αρχές του όλου συστήματος.



Σχήμα 8 - Αρχιτεκτονική του συστήματος ARION

Το σύστημα του **ARION** αποτελείται από ένα σύνολο καταναμημένων κόμβων, που περιέχουν διαφορετικά σύνολα δεδομένων και προγράμματα (δηλαδή επιστημονικές συλλογές). Οι κόμβοι δια-λειτουργούν μεταξύ τους με τη χρησιμοποίηση μιας πλατφόρμας πρακτόρων, ενώ ταυτόχρονα παρέχουν τις βασικές υπηρεσίες για την εκτέλεση ροών εργασίας. Οι ροές εργασίας βασίζονται, τυπικά, σε καταναμημένα και αυτόνομα έργα και ελέγχονται από ένα κεντροποιημένο υπηρετητή, τον κύριο υπηρετητή του συστήματος **ARION**, όπως φαίνεται και στο σχήμα 9. Οι πράκτορες που έχουν εγκατασταθεί σε κάθε κόμβο εκτελούν έργα ροών εργασίας, κυρίως υπολογισμούς, και παρακολουθούν τις εργασίες τους.

Middleware System



Σχήμα 9 - Το σύστημα ARION αποτελεί σύστημα middleware

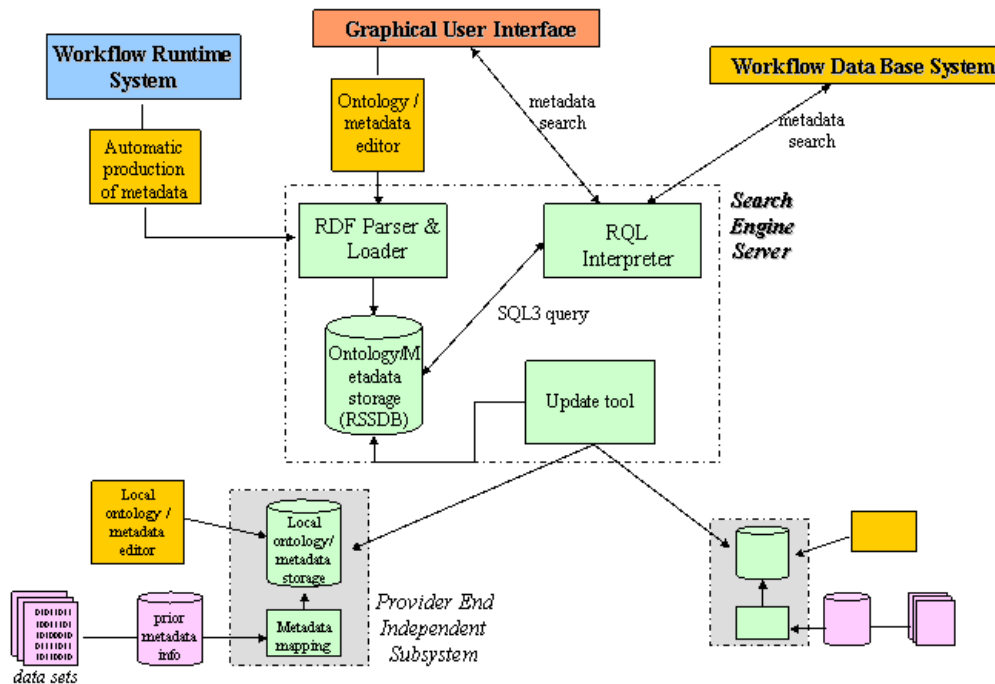
4.3.4.1. Η Μηχανή Αναζήτησης

Η Μηχανή Αναζήτησης (*Search Engine*), όπως φαίνεται και από το σχήμα 10, βασίζεται κυρίως σε μια ενιαία σειρά από εργαλεία με όνομα RDFSuite, η οποία πρόκειται να αναλυθεί παρακάτω. Ένας κειμενογράφος χρησιμοποιείται για την εισαγωγή μετα-δεδομένων και οντολογικών πληροφοριών, τα οποία εν συνεχεία φορτώνονται στο ORDBMS του συστήματος ARION. Μια καθολική αποθήκη, που βρίσκεται στον κεντρικό υπηρετητή του ARION, περιέχει όλες τις περιγραφές μετα-δεδομένων, που έχουν υποβληθεί ή δημιουργηθεί από το σύστημα είτε κεντρικά είτε τοπικά σε ένα επιστημονικό οργανισμό. Ένα εργαλείο ανανέωσης (update) είναι υπεύθυνο για τοπική συλλογή (σε κάθε ανεξάρτητο υποσύστημα κάποιου προμηθευτή του ARION) υποβληθέντων μετα-δεδομένων.

Η Μηχανή Αναζήτησης εκτελεί οποιαδήποτε απαίτηση επερώτησης, που προέρχεται από τις διεπαφές χρήσης ή από το σύστημα *Κατανεμημένης Αποθήκευσης Ροών Εργασίας Επιστημονικών Δεδομένων*. Οι επερωτήσεις, που μπορούν να τεθούν, είναι είτε για πληροφορίες μετα-δεδομένων είτε για έννοιες και ιδιότητες της οντολογίας.

Στις παρακάτω υπο-ενότητες θα αναφέρουμε πώς γίνονται οι προσδιορισμοί των μετα-δεδομένων του συστήματος και πώς γίνονται οι επερωτήσεις σε ένα πιο χαμηλό επίπεδο, συγκριτικά με εκείνο που παρουσιάζεται στο χρήστη.

ARION Search Engine Architecture



Σχήμα 10 - Αρχιτεκτονική της Μηχανής Αναζήτησης

4.3.4.1.1. Προσδιορισμοί μετα-δεδομένων

Η αναπαράσταση των μετα-δεδομένων στο σύστημα του ARION πραγματοποιείται μέσω της γλώσσας RDF (Resource Description Framework). Η RDF παρέχει ένα πλαίσιο για την αναπαράσταση περιγραφών πόρων, που είναι επεξεργάσιμοι από μηχανές. Ο προσδιορισμός του μοντέλου και της σύνταξης της RDF [77] ορίζει ένα τυπικό (formal) μοντέλο δεδομένων, που στηρίζεται στη Θεωρία Συνόλων.

Η RDF ορίζει το τρίδυμο: αντικείμενο-ιδιότητα-τιμή, ως ένα βασικό αρχέγονο (primitive) μοντελοποίησης και συστήνει μια τυποποιημένη σύνταξη για αυτό. Χρησιμοποιεί τη γλώσσα XML για την ανταλλαγή δεδομένων, που είναι δομημένα ως κατευθυνόμενοι γράφοι με ετικέτες, όπου οι κόμβοι των γράφων είναι πηγές και οι ακμές τους είναι ιδιότητες με συγκεκριμένο όνομα.

Τα σχήματα της RDF (RDFS) [78] παρέχουν ένα βασικό τύπο σχήματος για την RDF. Αντικείμενα, κλάσεις και ιδιότητες μπορούν να περιγραφθούν. Σε σχέση με τις οντολογίες, οι RDF/S προσφέρουν δύο σημαντικές συνεισφορές: μια τυποποιημένη σύνταξη για τη συγγραφή οντολογιών και ένα τυπικό σύνολο από αρχέγονα μοντελοποίησης, όπως οι σχέσεις 'instance-of' και 'subclass-of'.

Οι ικανότητες έκφρασης των RDF και RDF Schema είναι ικανοποιητικές στο πλήθος για τους στόχους του project ARION και χρησιμοποιούνται ως βάση μοντελοποίησης του πεδίου της γνώσης. Ιδιαίτερα, έχει υιοθετηθεί μια διαρθρωτική προσέγγιση για τη περιγραφή μετα-δεδομένων, η οποία καθοδηγείται από την οντολογία, με την έννοια ότι η κατασκευή των περιγραφών μετα-δεδομένων γίνεται

σε μια top-down μορφή, με το γέμισμα μιας δεδομένης οντολογίας, αντί σε μια μορφή bottom-up. Κάθε επιστημονικό αντικείμενο (σύνολο δεδομένων ή μοντέλο) περιγράφεται με μια συλλογή από ιδιότητες, που ή κληρονομούνται από την πατρική του κλάση ή είναι ιθαγενής στο συγκεκριμένο αντικείμενο.

4.3.4.1.2. *Εννοιολογικές ερωτήσεις*

Η ανάπτυξη της RDF δημιουργεί τη προοπτική μιας ευρείας αποδοχής προδιαγραφής για την αναπαράσταση της γνώσης στο διαδίκτυο. Όμως, η αναπαράσταση μόνο γνώσης και πληροφορίας δεν είναι αρκετή. Γλώσσες επερώτησης και εργαλεία χρειάζονται για να καταστήσουν ικανή τη δημιουργία εφαρμογών, που βασίζονται στη RDF. Μια τέτοια λύση είναι η RDFSuite [79].

Η RDFSuite είναι ένα σύστημα διαχείρισης (σειρά από εργαλεία) της αποθήκευσης και επερώτησης τόσο περιγραφών RDF όσο και περιγραφών RDF σχημάτων [80]. Αποτελείται από τρία βασικά μέρη: ένα εργαλείο κυρώσεως RDF (RDF validator), μια αποθηκευτική βάση δεδομένων για RDF (RSSDB) και μια γλώσσα επερωτήσεων (RQL).

Η βάση δεδομένων RDF Schema Specific Database (RSSDB) φορτώνει περιγραφές πόρων σε ένα αντικειμενο-σχεσιακό (object-relational) σύστημα διαχείρισης βάσεων δεδομένων (DBMS). Η αναπαράσταση ενός RDF σχήματος και η πληροφορία μετα-δεδομένων γίνεται ξεχωριστά, αποφεύγοντας τη προσέγγιση ενός μονολιθικού πίνακα για την αναπαράσταση RDF τριδύμων. Αυτό προσφέρει ευκαμψία για το υφιστάμενο ORDBMS και επιτρέπει ευκολότερο χειρισμό της πληροφορίας του σχήματος. Μετά την κύρωση των RDF μετα-δεδομένων και τον έλεγχο συνέπειας των περιγραφών σχήματος, ένας φορτωτής (loader) τα αποθηκεύει στη βάση δεδομένων.

Η γλώσσα επερωτήσεων RDF Query Language (RQL) [81] μπορεί να χρησιμοποιηθεί για τις ενιαίες επερωτήσεις τόσο πληροφορίας RDF μετα-δεδομένων όσο και RDF σχημάτων. Επομένως, αυτή η ικανότητα μπορεί να γίνει αντικείμενο εκμετάλλευσης για την υλοποίηση της πλοήγησης ενός σχήματος, αφού τα τεράστια RDF σχήματα μεταφέρουν πολύτιμη πληροφορία λόγω της βελτίωσης κλάσεων (class refinement).

Υπάρχουν διάφοροι ειδικοί ρόλοι, που μια οντολογία διαδραματίζει, προκειμένου να διευκολύνει τη διαδικασία των επερωτήσεων. Στην περίπτωση του ARION, η οντολογία χρησιμοποιείται ως βάση για σημασιολογική δόμηση και οργάνωση της αποθήκης των περιγραφών μετα-δεδομένων. Αυτή η δόμηση αποδίδει με τρεις διαφορετικούς τρόπους :

1. Ως ένα εννοιολογικό πλαίσιο βοήθειας προς το χρήστη, στην προσπάθειά του να καταλάβει το σχήμα της πληροφοριακής αποθήκης και να διατυπώσει επερωτήσεις.
2. Ως ένας οδηγός αντίληψης των μετα-δεδομένων, που κατασκευάζονται-καθοδηγούνται από την οντολογία.
3. Για να βοηθήσει τις διεπαφές χρήσης να δημιουργήσουν και να βελτιώσουν επερωτήσεις.

Η αναζήτηση, που βασίζεται σε οντολογίες, χρησιμοποιεί μια διαισθητική σχέση μεταξύ εννοιών, ώστε να προσφέρει ευφυή πρόσβαση σε πληροφορία. Με τη χρήση της RQL, οι επερωτήσεις μπορούν να χρησιμοποιήσουν είτε εννοιολογικές

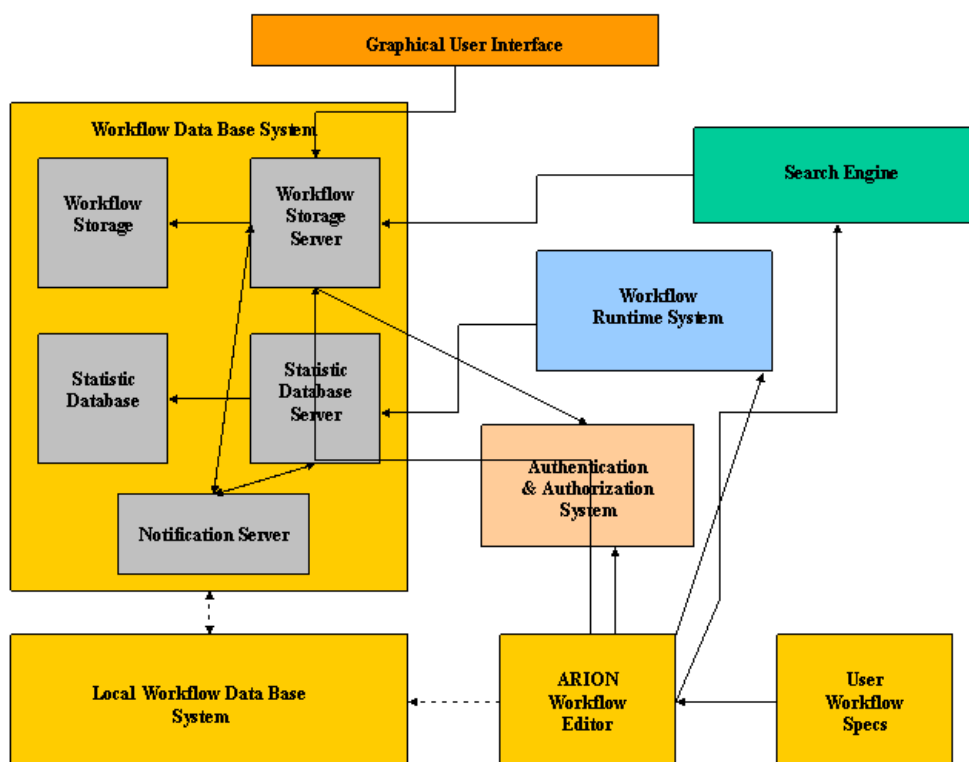
έννοιες (RDF schema) ή απλώς να είναι βασιζόμενες σε λέξεις (RDF metadata). Ο σημασιολογικός πλούτος της οντολογίας αποτελεί σημαντικό παράγοντα, αφού μια πιο πλούσια οντολογία μπορεί να βελτιώσει την αναζήτηση.

4.3.4.2. Το Σύστημα Κατανεμημένης Αποθήκευσης Ροών Εργασίας Επιστημονικών Αντικειμένων

Το υποσύστημα: Σύστημα Κατανεμημένης Αποθήκευσης Ροών Εργασίας Επιστημονικών Αντικειμένων υλοποιήθηκε στα πλαίσια της παρούσας Μεταπτυχιακής Εργασίας. Η αρχιτεκτονική του παρουσιάζεται συνοπτικά, ώστε ο αναγνώστης να έχει μια πλήρη εικόνα για το σύστημα του ARION. Μια πιο λεπτομερής περιγραφή γίνεται στο ακριβώς επόμενο κεφάλαιο.

Το Σύστημα Κατανεμημένης Αποθήκευσης Ροών Εργασίας Επιστημονικών Αντικειμένων (Workflow Data Base System), όπως φαίνεται στο σχήμα 11, αποτελείται από τα εξής κύρια μέρη: τον Κειμενογράφο Ροών Εργασίας (Workflow Editor), την Αποθήκη Ροών Εργασίας (Workflow Storage), τον Υπηρετητή της Αποθήκης Ροών Εργασίας (Workflow Storage Server), τη Στατιστική Βάση Δεδομένων (Statistical Database) και τον Υπηρετητή της Στατιστικής Βάσης Δεδομένων (Statistical Database Server).

Ο Κειμενογράφος Ροών Εργασίας (Workflow Editor) χρησιμοποιείται για τον ορισμό νέων προσδιορισμών ροών εργασίας ή για τη μετατροπή των ήδη υπαρχόντων προσδιορισμών. Οι προσδιορισμοί, αφού σωθούν, στέλνονται στον Υπηρετητή της Αποθήκης Ροών Εργασίας, ο οποίος και τους αποθηκεύει στο χώρο-αποθήκη, που έχει υπό τη προστασία του.



Σχήμα 11 - Η αρχιτεκτονική του συστήματος AWDB

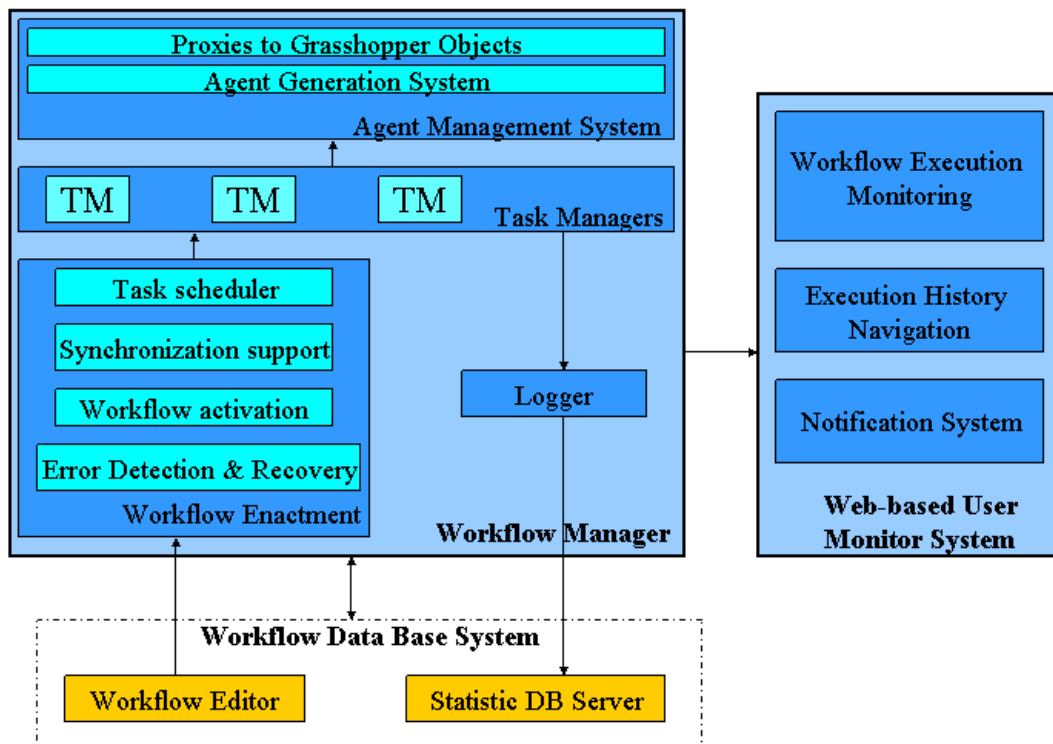
Ο Υπηρετητής της Αποθήκης Ροών Εργασίας (Workflow Storage Server) είναι το πιο σημαντικό κομμάτι του υποσυστήματος. Επικοινωνεί και με τις Γραφικές Διεπαφές Χρήσης και με τη Μηχανή Αναζήτησης, αλλά και με το Σύστημα Εκτέλεσης Ροών Εργασίας. Για τη διευκρίνιση της λειτουργικότητάς του, υποθέστε ότι ο χρήστης εκδίδει μια επερώτηση για ένα σύνολο δεδομένων. Ο υπηρετητής επικοινωνεί με τη Μηχανή Αναζήτησης, για να προσδιορίσει αν το ζητούμενο σύνολο δεδομένων είναι αποθηκευμένο στο σύστημα του **ARION**. Αν η απάντηση είναι αρνητική, ψάχνει την Αποθήκη Ροών Εργασίας για να βρει έναν προσδιορισμό ροής εργασίας, που μπορεί να παράγει το συγκεκριμένο σύνολο δεδομένων. Αν αυτή η ροή εργασίας υπάρχει, τότε τη στέλνει στο σύστημα Εκτέλεσης Ροών Εργασίας.

Όταν η διαδικασία εκτέλεσης μιας ροής εργασίας ολοκληρωθεί, το σύστημα Εκτέλεσης Ροών Εργασίας επικοινωνεί με τον Υπηρετητή της Στατιστικής Βάσης Δεδομένων, για να αποθηκεύσει, στη Στατιστική Βάση Δεδομένων, πληροφορία σχετιζόμενη με την εκτέλεση, όπως ο χρόνος εκτέλεσης, οι πόροι που δεσμεύθηκαν, οι κόμβοι δικτύου που χρησιμοποιήθηκαν, ο αριθμός των λαθών εκτέλεσης, προειδοποιήσεις κ.α..

4.3.4.3. Το Σύστημα Εκτέλεσης Ροών Εργασίας

Το Σύστημα Εκτέλεσης Ροών Εργασίας (Workflow Runtime System), που φαίνεται στο σχήμα 12, είναι αρκετά διαρθρωτικό και επιτρέπει υψηλή ευκαμψία σε σχέση με την εγκατάσταση και τη λειτουργία του. Η σχεδίασή του εστιάζεται στην αντιμετώπιση θεμάτων κλιμακωσιμότητας με τον καλύτερο δυνατό τρόπο, ενώ παράλληλα προσπαθεί να διατηρήσει την ευχρηστία του συστήματος αυτού.

Workflow Runtime System Architecture



Σχήμα 12 - Αρχιτεκτονική του Workflow Runtime System

4.3.4.3.1. Μέρη Βρισκόμενα στον Υπηρετητή

Ο Υπηρετητής (Server) του Συστήματος Εκτέλεσης Ροών Εργασίας προσφέρει την απαιτούμενη λειτουργικότητα για την 'ύστερα από ζήτηση' παραγωγή συνόλων δεδομένων, μέσω της εκτέλεσης ροών εργασίας. Λειτουργεί ως γέφυρα διαλειτουργικότητας μεταξύ πελατών, όπως είναι οι εξωτερικές εφαρμογές, οι δυναμικές σελίδες του διαδικτύου (Java Servlets) και τα Java [82] Applets, και στο υπολογιστικό πλέγμα, το οποίο θα παράγει τα απαιτούμενα σύνολα δεδομένων. Για κάθε αίτηση, που απορρέει από πελάτες και έχει ως αποτέλεσμα την εκτέλεση ροών εργασίας, μια Μηχανή Ροών Εργασίας (Workflow Engine) δημιουργείται. Η Μηχανή Ροών Εργασίας φροντίζει, αποκλειστικά, την εκτέλεση των αντίστοιχων ροών εργασίας. Ο Υπηρετητής αποτελείται από διάφορα συστατικά μέρη, τα οποία περιγράφονται παρακάτω :

Σύστημα Διοίκησης Πρακτόρων (Agent Management System) : Είναι το μέρος, που συνδέει τον Υπηρετητή με την πλατφόρμα των πρακτόρων. Παρέχει το μηχανισμό δημιουργίας πρακτόρων, καθώς επίσης και παραγγελιοδόχα αντικείμενα (proxy objects), που χρησιμοποιούνται για την απευθείας πρόσβαση στα αντικείμενα της πλατφόρμας. Επιπλέον, υλοποιεί ένα μηχανισμό επικοινωνίας, που επιτρέπει αμφίδρομη επικοινωνία μεταξύ των απομακρυσμένων πρακτόρων, έτσι ώστε να διευκολύνει την εκτέλεση των ροών εργασίας.

Χειριστής Δικτύου (Net Handler) : Είναι το μέρος που χειρίζεται την επικοινωνία μεταξύ των πελατών και των άλλων συνεργατικών υπηρετητών εκτέλεσης. Οι αιτήσεις από τους πελάτες επεξεργάζονται και αργότερα προωθούνται στην κατάλληλη μηχανή ροών εργασίας για επιπλέον επεξεργασία. Οι μηχανές ροών εργασίας επικοινωνούν μέσω αυτού του συστατικού μέρους με άλλους υπηρετητές, για να συντονίσουν την εκτέλεση μιας ροής εργασίας και για να εξασφαλίσουν ότι όλοι οι σχετικοί υπηρετητές διατηρούν συνεπή δεδομένα, που έχουν σχέση με τη τρέχουσα κατάσταση της ροής εργασίας.

Logger : Αυτό το συστατικό μέρος καταγράφει όλες εκείνες τις δραστηριότητες, που συμβαίνουν μέσα στον υπηρετητή. Η ακριβή ώρα, μια σύντομη περιγραφή του γεγονότος, καθώς επίσης και η πηγή του, διατηρούνται-κρατούνται σε μόνιμη αποθήκη.

Παρακολούθηση (Monitoring) : Αυτό το μέρος επιτρέπει τη συνεχή παρακολούθηση της κατάστασης του υπηρετητή και των μηχανών ροών εργασίας του. Θα υπάρχει αφενός ένα εργαλείο γραμμής εντολών (command line tool) και αφετέρου μια γραφική διεπαφή χρήσης βασισμένη στο διαδίκτυο που θα επιτρέπουν την πλήρη πρόσβαση στις λειτουργίες του υπηρετητή τόσο για τους χρήστες όσο και για τον διοικητή του υπηρετητή (server administrator). Επίσης, το εν λόγω συστατικό μέρος περιέχει ένα σύστημα ειδοποίησης, που ενημερώνει τους χρήστες με τη βοήθεια μιας κατάλληλης μεθόδου (π.χ. real-time, e-mail, SMS).

Μια μηχανή ροών εργασίας δημιουργείται από τον υπηρετητή, κάθε φορά που υπάρχει ανάγκη για την εκτέλεση μιας ροής εργασίας. Αποτελείται από τα εξής υπομέρη, όπως φαίνονται και στην εικόνα :

Προγραμματιστής Έργων (Task Scheduler) : Αυτό το υπομέρος διαβάζει τον προσδιορισμό της ροής εργασίας, που θα εκτελεστεί, και καθορίζει τα έργα που πρέπει να δρομολογηθούν για εκτέλεση. Ζητάει ένα *Διαχειριστή Έργου (Task Manager)* από τη *Δεξαμενή Διαχειριστών Έργου (Task Managers Pool)* και του αναθέτει την εκτέλεση ενός έτοιμου έργου. Για κάθε έτοιμο έργο, ένα νέο νήμα

εκτέλεσης δημιουργείται για τον αντίστοιχο Διαχειριστή Έργων, επιτρέποντας με αυτό τον τρόπο πολλούς Διαχειριστές Έργων να τρέξουν παράλληλα. Ο Προγραμματιστής ενημερώνεται για τη πρόοδο των εκτελούμενων έργων από τον/τους Διαχειριστές Έργων του.

Δεξαμενή Διαχειριστών Έργου (Task Managers Pool) : Αυτό το υπομέρος περιέχει τους *Διαχειριστές Έργου*, που είναι διαθέσιμοι για χρήση από τον *Προγραμματιστή Έργων*. Μετράει τους απασχολημένους και τους διαθέσιμους *Διαχειριστές Έργου* και μπορεί να δημιουργήσει νέους εάν προκύψει αντίστοιχη ανάγκη.

Διαχειριστής Έργου (Task Manager) : Ο Διαχειριστής Έργου χειρίζεται την εκτέλεση ενός έργου, που του έχει ανατεθεί. Διαβάζει το προσδιορισμό του έργου και, κατόπιν, ετοιμάζει όλες τις προετοιμασίες για την εκτέλεση του. Αν χρειαστεί, συντονίζει τη μεταφορά αρχείων με δεδομένα στον υπολογιστή του έργου και ζητάει τις απαιτούμενες παραμέτρους αρχικοποίησης για το έργο από το χρήστη, με τη βοήθεια του συστήματος ειδοποίησης του Υπηρετητή. Όταν, όλες οι απαιτούμενες παράμετροι γίνουν διαθέσιμες, το *Σύστημα Διοίκησης Πρακτόρων* καλείται και ένας Πράκτορας Εκτέλεσης Έργου δημιουργείται, ο οποίος και μεταναστεύει στο στοχευόμενο υπολογιστή, για να εκτελέσει το έργο. Ο *Διαχειριστής Έργου* ενημερώνεται για την πρόοδο της εκτέλεσης του έργου από τον αντίστοιχο πράκτορα μέσω ενός αντικειμένου επικοινωνίας, το οποίο έχει ανατεθεί για να διευκολύνει την επικοινωνία μεταξύ των δύο συμβαλλόμενων. Ο *Διαχειριστής Έργου* ενημερώνει τον *Προγραμματιστή Έργων* για τις αλλαγές που συντελούνται στη δική του κατάσταση.

Κατάσταση (State) : Κρατάει την τρέχουσα κατάσταση της μηχανής ροών εργασίας.

4.3.4.3.2. Μέρη βρισκόμενα στον πελάτη

Ο χρήστης μπορεί να αλληλεπιδράσει με τον υπηρετητή, μέσω παρεχόμενων διεπαφών χρήσης του διαδικτύου. Υπάρχουν δύο είδη διεπαφών χρήσης :

1. Δυναμικά δημιουργημένες σελίδες του διαδικτύου, που βασίζονται στην τεχνολογία των Java Servlets.
2. Java Applets.

Μια πιο λεπτομερής περιγραφή των διεπαφών χρήσης του διαδικτύου παρέχεται σε μια προσεχή ενότητα.

Οι διεπαφές χρήσης του διαδικτύου προωθούν τις αιτήσεις των χρηστών στον Υπηρετητή και ενημερώνουν το χρήστη για το αποτέλεσμα των λειτουργιών, που τελούνται στο συγκεκριμένο Υπηρετητή. Κάθε αίτηση από το χρήστη έχει ως αποτέλεσμα το άνοιγμα μιας υποδοχής (socket) σύνδεσης με τον Υπηρετητή.

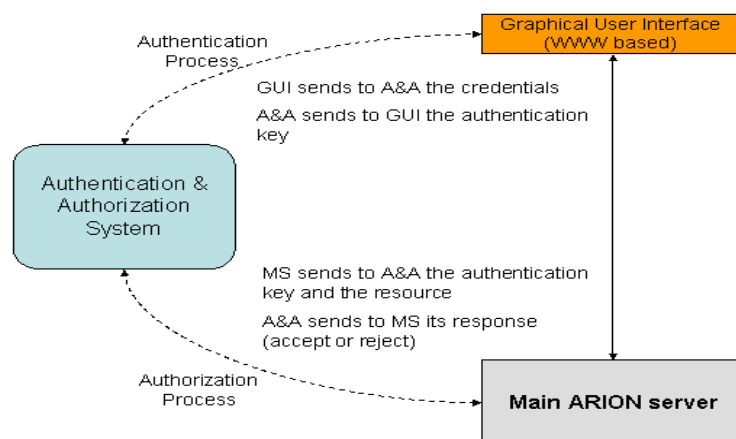
Ένας Υπηρετητής μπορεί να ενεργεί ως πελάτης για έναν άλλο Υπηρετητή. Ιδιαίτερα, οι συνεργατικοί Υπηρετητές έχουν τη δυνατότητα να ζητήσουν ο ένας από τον άλλο μια δοσοληψία, προκειμένου να εξασφαλιστεί η διατήρηση της έγκυρης κατάστασης του συστήματος.

4.3.4.4. Το Σύστημα Ταυτοποίησης και Εξουσιοδότησης Χρήστη

Η διαδικασία ταυτοποίησης, κατά τη διάρκεια της οποίας ο χρήστης αποδεικνύει την ταυτότητά του, υλοποιείται από ένα μηχανισμό ταυτοποίησης. Συνεπώς, ένας μηχανισμός ταυτοποίησης χορηγεί ή αρνείται την άδεια πρόσβασης σε ψηφιακά αντικείμενα (δεδομένα ή πόροι). Για να οριστούν οι εν λόγω οι άδειες, έχει υιοθετηθεί ένας έλεγχος πρόσβασης βασισμένος σε ρόλους (role-based access control) [83] για τη διαχείριση χρηστών και αντικειμένων. Αν ένας ρόλος ανατεθεί σε ένα χρήστη, τότε αυτός ο χρήστης αποκτά όλα τα δικαιώματα πρόσβασης σε αυτόν το ρόλο. Οι ρόλοι διαμοιράζονται στο κατακευματισμένο σύστημά μας με τη χρησιμοποίηση μιας ιεραρχικής δομής (εξαρτώμενη στις περιοχές εμπιστοσύνης (trust domains)). Όταν ένας χρήστης ξεκινήσει μια ροή εργασίας, το Σύστημα Εκτέλεσης Ροών Εργασίας ενεργεί εκ μέρους του και διεκπεραιώνει όλες τις απαραίτητες εξουσιοδοτήσεις με το πέρασμα ενός πιστοποιητικού (certificate), που δημιουργείται από το μηχανισμό ταυτοποίησης.

4.3.4.4.1. Η διαδικασία Ταυτοποίησης και Εξουσιοδότησης

Η διαδικασία ταυτοποίησης χρήστη ξεκινά από μια γραφική διεπαφή χρήσης του διαδικτύου, όπως φαίνεται και στο σχήμα 13. Μια διεπαφή χρήσης λειτουργίας συνθήματος (login module interface) ανακινείται και η ταυτοποίηση παίρνει μέρος με το πέρασμα των διαπιστευτηρίων (credentials), που χρειάζονται από τη μέθοδο του συνθήματος (login method). Ο χρήστης παρέχει αυτά τα διαπιστευτήρια ταυτοποίησης, που στην περίπτωση του **ARION** είναι δεδομένα υπογεγραμμένα από αυτόν, τα οποία περνάνε στο Σύστημα Ταυτοποίησης και Εξουσιοδότησης. Τελικώς, το σύστημα θα προσδιορίσει εάν είναι σωστά για τη χορηγία πρόσβασης.



Σχήμα 13 - Η διαδικασία ταυτοποίησης χρήστη

Αν η διαδικασία ταυτοποίησης ολοκληρωθεί με επιτυχία, ένα μοναδικό κλειδί στέλνεται πίσω στο χρήστη (που είναι υπογεγραμμένο από το σύστημα ταυτοποίησης), το οποίο περιέχει πληροφορίες (principals), όπως το όνομα του χρήστη, την έγκυρη περίοδο χρήσης, τους ρόλους για αυτόν τον χρήστη κ.α.. Αυτή η πληροφορία είναι το πιστοποιητικό, που φανερώνει ότι ο χρήστης έχει ταυτοποιηθεί. Επίσης, η πληροφορία αυτή θα χρησιμοποιηθεί και για την περαιτέρω διαδικασία εξουσιοδότησης.

Το πιστοποιητικό ταυτοποίησης αποτελείται από ένα σύνολο από principals, που φανερώνουν ρόλους στη περίπτωση μας. Για κάθε μεταγενέστερη λειτουργία, που έχει ελεγχθεί για ασφάλεια, ο μηχανισμός εξουσιοδότησης θα προσδιορίζει αυτόματα αν αυτή η λειτουργία χορηγεί την απαιτούμενη άδεια για ένα συγκεκριμένο principal (ρόλο). Αν ισχύει η τελευταία περίπτωση και αν το πιστοποιητικό του χρήστη, που σχετίζεται με τη συγκεκριμένη λειτουργία περιέχει τον επιθυμητό ρόλο, τότε η λειτουργία θα επιτραπεί.

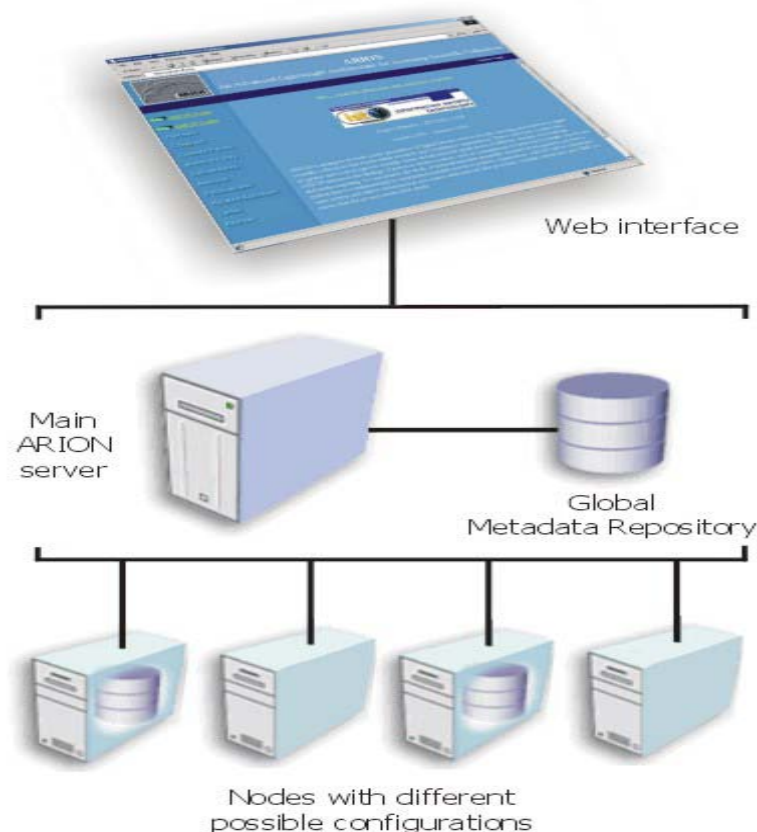
Η διαδικασία εξουσιοδότησης γίνεται σε σύνδεση με τον Κύριο Υπηρετητή του συστήματος του **ARION** και πιο ειδικά με το Σύστημα Κατανεμημένης Αποθήκευσης Ροών Εργασίας Επιστημονικών Αντικειμένων. Το πιστοποιητικό ταυτοποίησης, μαζί με τους πόρους, που θα προσπελαστούν, στέλνεται στο Σύστημα Κατανεμημένης Αποθήκευσης Ροών Εργασίας Επιστημονικών Αντικειμένων. Η άδεια δεν χορηγείται ή χορηγείται ανάλογα με την πληροφορία εξουσιοδότησης, που είναι ενσωματωμένη στο σύστημα. Αν η ροή εργασίας προς εκτέλεση επιβάλλει τη χρήση ενός ή περισσότερων προγραμμάτων/συνόλων δεδομένων, τότε πρέπει να επαληθευθεί εάν ο χρήστης έχει πράγματι πρόσβαση σε όλους αυτούς τους μεμονωμένους πόρους.

Εκτός από τη διαδικασία ταυτοποίησης και εξουσιοδότησης, υπάρχει η ανάγκη για τη διαδικασία της αμοιβαίας ταυτοποίησης, ανάμεσα στο περιβάλλον του Συστήματος Κατανεμημένης Αποθήκευσης Ροών Εργασίας Επιστημονικών Αντικειμένων και σε αυτό του Συστήματος Εκτέλεσης Ροών Εργασίας. Αυτό γίνεται με μια σειρά από υπογεγραμμένες ερωτήσεις και απαντήσεις μεταξύ των δύο συμβαλλόμενων, για να επιβεβαιώσουν / επαληθεύσουν ο ένας τον άλλον.

4.3.4.5.Ειδικές λεπτομέρειες-χαρακτηριστικά

4.3.4.5.1. Ειδικές Λεπτομέρειες

Η κατανεμημένη αρχιτεκτονική τριών δεσμών (three-tier) του συστήματος του **ARION** προσδίδει ένα υψηλό επίπεδο αξιοπιστίας και κλιμακωσιμότητας, όπως παρατηρείται και στην εικόνα 14. Δεν χρειάζεται όλοι οι κόμβοι (των προμηθευτών) να έχουν τοπικά εγκατεστημένα και τα τέσσερα υποσυστήματα του **ARION**. Εξαιτίας της ευκαμψίας της αρχιτεκτονικής του **ARION**, πολλαπλοί τρόποι σύνταξης (deployment) είναι διαθέσιμοι. Μέσω της ανάλυσης του περιβάλλοντος του κάθε προμηθευτή και του προσδιορισμού των γενικών διοικητικών αναγκών και πόρων, μια επιλογή σύνταξης, που προσαρμόζεται βέλτιστα στον κάθε προμηθευτή, μπορεί να διαλεχθεί με αποδοτικότερο τρόπο.



Εικόνα 14 - Κατανεμημένη αρχιτεκτονική τριών δεσμών του συστήματος ARION

Σε μια ελάχιστη ρύθμιση σύνταξης, μόνο η πλατφόρμα των πρακτόρων συντάσσεται στην πλευρά του τοπικού υπηρετητή κόμβου του προμηθευτή. Ούτε η Μηχανή Αναζήτησης (δεν υπάρχει μια τοπική αποθήκη μετα-δεδομένων), ούτε το Σύστημα Κατανεμημένης Αποθήκευσης Ροών Εργασίας Επιστημονικών Αντικειμένων εγκαθίστανται. Σε αυτήν τη ρύθμιση, ο κύριος υπηρετητής του **ARION** συνδέεται απευθείας με τον προαναφερόμενο κόμβο, αξιοποιώντας πράκτορες λογισμικού, και εκτελεί έργα ροών εργασίας. Η πλατφόρμα των πρακτόρων μπορεί να εγκατασταθεί σε κάθε υπολογιστή (ανεξάρτητα του λειτουργικού συστήματος που χρησιμοποιεί αυτός), εφόσον η γλώσσα προγραμματισμού Java έχει εγκατασταθεί και διευθετηθεί μέσα σε αυτόν.

Οι άλλες δύο ρυθμίσεις, που φαίνονται στο σχήμα 14, παρουσιάζουν μια περισσότερο αυτόνομη και αυτό-διοικητική σύνταξη. Τοπικές επερωτήσεις μπορούν να διεκπεραιωθούν. Επιπλέον, η εκτέλεση τελείως τοπικών ροών εργασίας είναι καθίσταται εφικτή. Η αρχιτεκτονική του **ARION** μπορεί εύκολα να επεκταθεί, ώστε να μπορεί να διαχειριστεί όχι μόνο οποιοδήποτε αριθμό πελατών, αλλά και κόμβους υπηρετητών, που είναι τοπικοί στους προμηθευτές επιστημονικών πόρων.

4.3.4.5.2. Ειδικά χαρακτηριστικά

Το σύστημα του **ARION** παρέχει κλιμακωσιμότητα, τόσο σε επίπεδο σημασιολογίας όσο και σε επίπεδο αρχιτεκτονικής του συστήματος. Στο επίπεδο της

σημασιολογίας, η κλιμάκωση αφορά τις επιστημονικές περιοχές, που η οντολογία του **ARION** καλύπτει. Λόγω της προσεγμένης σχεδίασής της, η οντολογία του **ARION** μπορεί εύκολα να επεκταθεί, με τον ίδιο όμως τρόπο σχεδίασης, για να περιλάβει και άλλες επιστημονικές περιοχές. Στο επίπεδο της αρχιτεκτονικής, επαγωγικά, μπορεί να αποδειχθεί ότι η αρχιτεκτονική του **ARION** είναι κλιμακώσιμη, αφού το κάθε υποσύστημά του παρουσιάζει μια κλιμακώσιμη αρχιτεκτονική.

Επιπροσθέτως, χαρακτηριστικά, που έχουν ενσωματωθεί στο σύστημα του **ARION**, περιλαμβάνουν την αυτόματη παραγωγή μετα-δεδομένων για σύνολα δεδομένων, που παράγονται κατόπιν αιτήσεως από κάποιο χρήστη (on-demand). Αυτό το χαρακτηριστικό έχει αποδειχθεί ότι είναι πολύ χρήσιμο για τους προμηθευτές, αφού η εγγραφή μετα-δεδομένων για μια προοδευτικά αυξανόμενη ποσότητα από παραγόμενα σύνολα, μπορεί να αποτελέσει ένα επαναληπτικό και πληκτικό έργο.

Το πρόβλημα της ολοκλήρωσης και της διαλειτουργικότητας ροών εργασίας αντιμετωπίστηκε με την ικανοποίηση των απαιτήσεων του οργανισμού WFMC. Ο WFMC επιβάλλει τρεις απαιτήσεις στο περιβάλλον των πρακτόρων, που είναι :

1. Πρέπει να υπάρχει μια κοινή αντίληψη των βασικών εννοιών της περιοχής. Η όλη επικοινωνία μεταξύ των πρακτόρων οφείλει να βασίζεται σε αυτήν την κοινή αντίληψη.
2. Το σύστημα πρέπει να υποστηρίζει ένα βασικό ρεπερτόριο από δια-πρακτορική επικοινωνία.
3. Η αλληλεπίδραση μεταξύ των πρακτόρων πρέπει να δύναται συντονισμού, όποτε πρέπει να διεκπεραιώσουν δραστηριότητα μέσω συνεργασίας.

Η απαίτηση για απόκτηση κοινής αντίληψης της διαμοιραζόμενης πληροφορίας έχει διευθετηθεί αφενός με την ανάπτυξη οντολογιών περιοχής, που είναι ανεξάρτητες έργων και αναπαριστούν τα αντικείμενα της περιοχής, και αφετέρου με την ανάπτυξη οντολογιών έργων, που είναι ανεξάρτητες της περιοχής. Η απαίτηση για κοινή αντίληψη της δια-πρακτορικής επικοινωνίας έχει διευθετηθεί με τον ορισμό μιας γλώσσας επικοινωνίας πρακτόρων, που δηλωτικά περιγράφει ένα κεντρικό σύνολο από πράξεις επικοινωνίας. Ο συντονισμός των πρακτόρων υλοποιείται από την ομόσπονδη αρχιτεκτονική του συστήματος του **ARION**.

4.3.5. Διεπαφές Χρήσης

Η διεπαφή χρήσης ενός συστήματος, όπως αυτό του **ARION**, έχει τρεις κύριες ευθύνες. Πρώτον, πρέπει να παρέχει μηχανισμούς πρόσβασης στην οντολογία της περιοχής και στην πληροφορία των μετα-δεδομένων, να τα δημοσιεύει, να τα διαχειρίζεται και φυσικά να είναι ικανή να τα επερωτά. Οι διεπαφές χρήσης, που βασίζονται στο διαδίκτυο, επιτρέπουν στους χρήστες να πλοηγηθούν μέσα από μια οντολογία αποδοτικά και εύκολα (π.χ. με τη χρήση υπερβολικών δένδρων). Αυτό παρέχει ένα δυναμικό στυλ αλληλεπίδρασης 'see and go', που παρακρατεί την απλότητα του 'point and click'. Ένας Κειμενογράφος Μετα-δεδομένων (Metadata Editor) εισάγει νέα πληροφορία για μετα-δεδομένα μέσα στο σύστημα ή τροποποιεί την ήδη υπάρχουσα. Η οντολογία μπορεί, επίσης, να χρησιμοποιηθεί από τη διεπαφή χρήσης της *Μηχανής Αναζήτησης* για να προσφέρει καθοδήγηση, όταν διατυπώνονται επερωτήσεις. Αυτό έχει αποδειχθεί να είναι ο πιο αποτελεσματικός τρόπος ανάκτησης ακριβών αποτελεσμάτων αναζήτησης από μια απλή επερώτηση μέσω μιας λέξης κλειδί.

Δεύτερον, πρέπει να οπτικοποιεί τα παραγόμενα ή τα ανακτώμενα σύνολα δεδομένων. Στη περίπτωση μας, αυτές είναι κυρίως γεωγραφικές πληροφορίες, οπότε και πρέπει να παρέχει δυνατότητες Συστήματος Γεωγραφικών Πληροφοριών (Geographical Information System – GIS [84]), προκειμένου να τα οπτικοποιεί πάνω σε χάρτες. Για παράδειγμα, τα αποτελέσματα της εκτέλεσης μιας ροής εργασίας μπορεί να παρουσιαστούν γραφικά στην οθόνη. Στη πραγματικότητα, τα Java Applets μπορεί να χρησιμοποιηθούν για την οπτικοποίηση αυτού του είδους της πληροφορίας.

Τρίτον, πρέπει να παρέχει τρόπους αλληλεπίδρασης και ελέγχου της εκτέλεσης ροών εργασίας. Έχουν υιοθετηθεί τρεις διαφορετικές προσεγγίσεις στη σχεδίαση των διεπαφών χρήσης για αυτούς τους σκοπούς. Στη προσέγγιση της Σύνοψης Ροών Εργασίας (Workflow Overview), ο χρήστης συλλαμβάνει την όλη διαδικασία της εκτέλεσης μιας ροής εργασίας και είναι ενήμερος για τα διάφορα βήματα, που περιλαμβάνουν την ροή εργασίας. Ο χρήστης μπορεί να αλληλεπιδράσει με το σύστημα εκτέλεσης μέσω μιας διεπαφής χρήσης, που βασίζεται στο διαδίκτυο. Αυτή υλοποιείται με δυναμικά παραγόμενες σελίδες διαδικτύου, που στηρίζονται με τη σειρά τους στα Java Servlets.

Σε αυτήν την περίπτωση, ο χρήστης μπορεί να επιλέξει :

- Να παρακολουθήσει την εκτέλεση ροής εργασίας και να ανακτήσει στατιστικές και άλλες πληροφορίες σχετικά με την εκτέλεση, τόσο της τρέχουσας όσο και των προηγούμενων φάσεων.
- Να αναστείλει ή να ξαναρχίσει την εκτέλεση μιας ροής εργασίας.
- Να παραδώσει τις παραμέτρους εισόδου, που χρειάζονται για ένα έργο, ώστε να συνεχιστεί η εκτέλεσή του.
- Σε συγκεκριμένα σημεία (checkpoints) να επιλέξει αν η ροή εργασίας θα συνεχίσει την εκτέλεσή της ή θα επιστρέψει σε ένα προηγούμενο στάδιο.

Στην προσέγγιση της Εικόνας Εφαρμογής (Application View), ο χρήστης παρακολουθεί την εκτέλεση μιας ροής εργασίας σαν μια εφαρμογή, που έχει το χαρακτήρα stand-alone και κρύβει τη πολυπλοκότητα της εργασίας με αρκετές και διαφορετικές διαδικασίες. Αυτή η εφαρμογή υλοποιείται σαν ένα Java Applet. Ο χρήστης μπορεί να προγραμματίσει την εκτέλεση της όλης διαδικασίας-ροής εργασίας και να αποκτήσει το τελικό αποτέλεσμα. Αυτό υπονοεί ότι οι παράμετροι εισόδου στα έργα θα πρέπει να είναι εκ των προτέρων προσδιορισμένες και οφείλουν να θεωρηθούν ως παράμετροι εισόδου στην εφαρμογή. Έπεται ότι η συγκεκριμένη προσέγγιση θα χρησιμοποιηθεί από εκείνους, που επιθυμούν να προγραμματίσουν ένα μεγάλο αριθμό από εκτελέσεις ροών εργασίας (batch execution) και να αποκτήσουν το τελικό αποτέλεσμα, χωρίς παρέμβαση κατά τη διάρκεια της εκτέλεσής αυτών των ροών εργασίας.

Η τρίτη προσέγγιση είναι ένας συνδυασμός των δύο προηγούμενων. Ο χρήστης θα είναι ικανός να προγραμματίσει την εκτέλεση ενός αριθμού από ροές εργασίας, παρέχοντας όλες τις παραμέτρους εισόδου από την αρχή. Όμως, το σύστημα θα απαιτεί επαλήθευση (μέσω μιας διεπαφής χρήσης, βασισμένης στο διαδίκτυο), όταν παραδείγματος χάριν μια παράμετρος χρησιμοποιείται.

Το σύστημα του **ARION** περιλαμβάνει, επίσης, μια συλλογή από εργαλεία, που απευθύνονται σε εμπειρογνώμονες χρήστες (π.χ. τους προμηθευτές). Εργαλεία, όπως το Protégé 2000 [85], μπορούν να χρησιμοποιηθούν από εμπειρογνώμονες περιοχών για την ανάπτυξη οντολογιών. Επιπλέον, ένας Κειμενογράφος Ροών Εργασίας (Workflow Editor) πρόκειται να υλοποιηθεί, ο οποίος και θα επιτρέπει στους χρήστες να δημιουργούν, να τροποποιούν και να διαγράφουν γραφικά προσδιορισμούς ροών εργασίας.

4.4. Συμπεράσματα για το σύστημα ARION

Το σύστημα **ARION** κατασκευάζεται χρησιμοποιώντας αποτελέσματα από δύο πρόσφατες προόδους: το Σημασιολογικό Δίκτυο (Semantic Web) και το Σημασιολογικό Πλέγμα (Semantic Grid). Επιπροσθέτως, οι προσδιορισμοί προτύπων, όπως τα πρότυπα μετα-δεδομένων και διαλειτουργικότητας (π.χ. OpenGIS) [86], αποτελούν ευθύς και πολύτιμες συνεισφορές στις υλοποιήσεις βασικών υπηρεσιών Ψηφιακών Βιβλιοθηκών επιστημονικού περιεχομένου. Οι Ψηφιακές Βιβλιοθήκες, όπως εκείνη του ARION, περιέχουν κατάλληλη υποδομή για την Ηλεκτρονική Επιστήμη (e-Science), που μπορεί να ειπωθεί ως ένα σύνολο από υπηρεσίες. Το υπολογιστικό πλέγμα που χρησιμοποιείται στο σύστημα ARION αποτελείται από γεωγραφικά κατανεμημένους πόρους, όπως υπολογιστές, δίκτυα, αποθήκες δεδομένων και οθόνες οπτικοποίησης πλοηγών διαδικτύου (we browser visualization displays). Οι πόροι αυτοί βρίσκονται στα μέλη οργανισμών Ψηφιακής Βιβλιοθήκης, τα οποία και παρέχουν το επιστημονικό περιεχόμενο και τους κατάλληλους πόρους.

Το σύστημα **ARION** παρέχει το middleware [1,2], που είναι αναγκαίο για την ολοκλήρωση αυτού του πλαισίου (ensemble), έτσι ώστε τα πολλά και ποικίλα κομμάτια του να λειτουργήσουν ως ενιαίο σύνολο. Παράλληλα, οργανώνει και ολοκληρώνει υπολογιστικές ευχέρειες (facilities) σε μια ολότητα, που εμφανίζει ειρμό. Επιπροσθέτως, ενεργεί ως ένα middleware για τη διασύνδεση μεταξύ: των χρηστών-των δεδομένων που θέλουν οι χρήστες να επεξεργαστούν-των υπολογιστικών πόρων, που απαιτούνται για αυτήν την επεξεργασία. Κύριο ρόλο σε αυτές τις αλληλεπιδράσεις διαδραματίζουν οι οντολογίες και οι ροές εργασίας, που καταγράφουν αφενός τη γνώση των εμπειρογνομώνων μέσω κατάλληλων εργαλείων δημοσίευσης/τροποποίησης για την αναπαράσταση/πλοήγηση της πληροφορίας, και αφετέρου τους υπολογισμούς πάνω από το πλέγμα (grid).

Συνοπτικά, το σύστημα **ARION** προσφέρει τη βασική υποδομή για πρόσβαση και παραγωγή επιστημονικής πληροφορίας σε ένα ανοικτό, κατανεμημένο και ομόσπονδο σύστημα. Πιο προηγμένες ηλεκτρονικές υπηρεσίες, που εξαρτώνται από το επιστημονικό περιεχόμενο της βιβλιοθήκης, μπορούν να δημιουργηθούν πάνω στην υποδομή του **ARION**. Τέτοιου είδους υπηρεσίες είναι η λήψη αποφάσεων για παροχή υπηρεσιών, ο έλεγχος, ο συντονισμός και η ασφάλεια. Το σύστημα **ARION** είναι μια εφαρμογή της Ηλεκτρονικής Επιστήμης, που παρέχει ένα σύνολο από υπηρεσίες, το οποίο μελλοντικά δύναται να επεκταθεί, έτσι ώστε να αποτελέσει μια πλήρη υπολογιστική εφαρμογή.

5. Το Σύστημα Κατανεμημένης Αποθήκευσης Ροών Εργασίας Επιστημονικών Αντικειμένων

5.1. Εισαγωγή

Ένα σύστημα διαχείρισης ροών εργασίας αποτελείται από δύο βασικά μέρη: το κατασκευαστικό (build-time component) και το εκτελεστικό (run-time component), καθώς και από μια βάση δεδομένων. Στο κατασκευαστικό μέρος ορίζεται το μοντέλο της ροής εργασίας. Επίσης, ορίζονται, με βάση αυτό το μοντέλο, οι ροές εργασίας, οι οποίες και αποθηκεύονται στη βάση δεδομένων. Το κατασκευαστικό μέρος, συνήθως, εκδηλώνεται με δύο τρόπους. Ο πρώτος τρόπος είναι ο ορισμός της πληροφορίας μιας ροής εργασίας μέσω μιας γραφικής διεπαφής χρήσης. Ο άλλος τρόπος είναι μέσω μιας κατάλληλης ή πρότυπης γλώσσας ορισμού ροών εργασίας. Στο εκτελεστικό μέρος, ο χρήστης δημιουργεί την περιγραφή μιας περίπτωσης ροής εργασίας, από τον ορισμό αυτής της ροής εργασίας, και το μέρος-σύστημα αναλαμβάνει να την εκτελέσει. Όταν τελειώσει η εκτέλεση, στατιστικές πληροφορίες, που αφορούν τόσο την περίπτωση ροής εργασίας όσο και την ίδια τη ροή εργασία, αποθηκεύονται στη βάση δεδομένων.

Το σύστημα 'Κατανεμημένης Αποθήκευσης Ροών Εργασίας Επιστημονικών Αντικειμένων' είναι ένα πρωτότυπο σύστημα, που προσομοιώνει το κατασκευαστικό μέρος ενός συστήματος διαχείρισης ροών εργασίας, ενώ παράλληλα εγγυάται την αποδοτική, αξιόπιστη και ασφαλή αποθήκευση τόσο των ροών εργασίας όσο και των στατιστικών πληροφοριών, που σχετίζονται με την εκτέλεση των ροών. Επιπλέον, αφαιρεί-κλέβει λειτουργικότητα από το παραδοσιακό εκτελεστικό μέρος ενός συστήματος διαχείρισης ροών εργασίας, με το να ορίζει περιπτώσεις ροών εργασίας και, ακολούθως, να τις στέλνει στο εκτελεστικό μέρος.

Στόχος του συστήματος είναι να αποτελέσει ένα ανεξάρτητο σύστημα, που θα προσφέρει αφενός, λειτουργίες ορισμού επιστημονικών ροών εργασίας και περιπτώσεων αυτών, και αφετέρου, λειτουργίες αναζήτησης και πρόσβασης σε πληροφορίες, που αφορούν ροές εργασίας από μια (κατανεμημένη) βάση δεδομένων. Επομένως, με την κατασκευή / προσθήκη ενός συστήματος εκτέλεσης ροών εργασίας, που απαιτεί ως είσοδο τον ορισμό είτε μιας επιστημονικής ροής εργασίας είτε την περίπτωση αυτής, θα διαπερατωθεί ένα ολοκληρωμένο σύστημα διαχείρισης επιστημονικών ροών εργασίας. Συνεπώς, το σύστημά μας θα αποτελέσει μια έτοιμη λύση-κομμάτι για την κατασκευή συστημάτων διαχείρισης επιστημονικών ροών εργασίας.

Στις ακόλουθες ενότητες αυτού του κεφαλαίου θα αναλυθούν οι στόχοι του συστήματος, θα τεκμηριωθεί η μεθοδολογία της σχεδίασής του και θα παρουσιαστούν οι λεπτομέρειες μιας πρώτης προσπάθειας υλοποίησής του. Στο επόμενο κεφάλαιο θα συζητηθούν συγκεκριμένες επεκτάσεις, τόσο σε επίπεδο σχεδίασης όσο και σε επίπεδο υλοποίησης.

5.2 Στόχοι του συστήματος

Η φάση της ανάλυσης απαιτήσεων αυτού του συστήματος αποτέλεσε μέρος της ανάλυσης απαιτήσεων του συστήματος ARION. Στηρίχθηκε τόσο στις αρχικές προδιαγραφές του συστήματος ARION όσο και σε συζητήσεις-συνεντεύξεις, που έγιναν με τους εμπειρογνώμων χρήστες του συστήματος, δηλαδή τους προμηθευτές

επιστημονικών δεδομένων. Προτού ξεκινήσουμε να αναφέρουμε τα αποτελέσματα της ανάλυσης απαιτήσεων, θα πρέπει να αναφέρουμε ότι τόσο το σύστημα **ARION** όσο και το Υποσύστημά του, στο οποίο είναι αφιερωμένο αυτό το κεφάλαιο, είναι πρωτότυπα, ακόμη εξελισσόμενα και όχι τελικά προϊόντα.

Σύμφωνα, λοιπόν, με τα αποτελέσματα της ανάλυσης απαιτήσεων, το σύστημά μας θα πρέπει να προσφέρει βασικές λειτουργίες και να εκδηλώνει συγκεκριμένα τεχνικά χαρακτηριστικά. Επίσης, θα πρέπει να στηρίζεται σε καθορισμένη αρχιτεκτονική.

Οι βασικές λειτουργίες, που προσφέρει το σύστημά μας, διακρίνονται σε δύο βασικές κατηγορίες: στις λειτουργίες διαχείρισης ροών εργασίας και περιπτώσεων αυτών και στις λειτουργίες διαχείρισης στατιστικών πληροφοριών για (περιπτώσεις) ροές εργασίας.

Οι λειτουργίες διαχείρισης ροών εργασίας και περιπτώσεων ροών εργασίας είναι οι εξής :

1. Δημιουργία νέων ορισμών ροών εργασίας.
2. Τροποποίηση ήδη υπαρχόντων ορισμών ροών εργασίας.
3. Αποθήκευση ορισμών ροών εργασίας.
4. Γραφική παρουσίαση ορισμού ροών εργασίας.
5. Αναζήτηση ορισμών ροών εργασίας με βάση την κλάση-οντότητα της οντολογίας στην οποία ανήκει κάποιο σύνολο δεδομένο, που παράγεται από αυτές τις ροές εργασίας.
6. Δημιουργία ορισμού και αρχείου εκτέλεσης για περίπτωση ροής εργασίας από τον ορισμό αυτής της ροής εργασίας.
7. Προώθηση ορισμού ή / και αρχείου εκτέλεσης περίπτωσης ροής εργασίας για εκτέλεση στο Σύστημα Εκτέλεσης Ροών Εργασίας.
8. Αποθήκευση ορισμού περίπτωσης ροής εργασίας και του αντίστοιχου αρχείου εκτέλεσης αν υπάρχει.
9. Γραφική παρουσίαση ορισμού περίπτωσης ροής εργασίας.

Οι λειτουργίες 1,2,3,4,6,7,8,9 μπορούν να υλοποιηθούν από ένα κειμενογράφο ροών εργασίας, που θα ορίζει και θα αποθηκεύει τόσο ροές εργασίας όσο και περιπτώσεις αυτών των ροών εργασίας. Οι λειτουργίες 5,6,7,8,9 μπορεί να υλοποιηθούν από ένα κειμενογράφο περιπτώσεων ροών εργασίας, ο οποίος μπορεί να έχει υλοποιηθεί επί τω πλείστο ως κομμάτι του κειμενογράφου ροών εργασίας. Και οι ορισμοί ροών εργασίας, αλλά και οι ορισμοί περιπτώσεων ροών εργασίας θα πρέπει να υπακούουν α) σε μια γλώσσα μοντελοποίησης και β) σε ορισμό ροών εργασίας. Επιπλέον, θα πρέπει να αποθηκεύονται (αρχικά ή τελικά) σε μορφή αρχείων. Οπότε, αν τα αρχεία ορισμού ροών εργασίας ή τα αρχεία ορισμού περιπτώσεων ροών εργασίας ή τα αρχεία εκτέλεσης είναι έτοιμα και περάσουν μέσα από ένα εργαλείο επαλήθευσης-εγκυρότητας-αποθήκευσης της μορφής τους, τότε οι λειτουργίες 1,2,3 ή 6,7,8 θα εκτελούνται αντίστοιχα από το εν λόγω εργαλείο. Σε αυτή την περίπτωση οι λειτουργίες 5 ή 9 θα γίνονται / επιτελούνται από κάποιες διεπαφές χρήσης. Θα πρέπει να τονίσουμε εδώ, ότι δεν διευκρινίστηκε ακόμα ο τρόπος αποθήκευσης των ορισμών των ροών εργασίας ή των περιπτώσεών τους.

Οι λειτουργίες της διαχείρισης στατιστικών πληροφοριών για ροές εργασίας ή περιπτώσεις τους είναι οι εξής :

1. Αποθήκευση στατιστικής πληροφορίας, προερχόμενης από το Σύστημα Εκτέλεσης Ροών Εργασίας, για μια συγκεκριμένη περίπτωση ροής εργασίας σε βάση δεδομένων. Η αποθήκευση αυτής της πληροφορίας μπορεί να προκαλέσει

την αποθήκευση ή την αλλαγή της πληροφορίας, που βρίσκεται στη βάση δεδομένων, για τη ροή εργασίας στην οποία ανήκει η συγκεκριμένη περίπτωση.

2. Αναζήτηση της στατιστικής πληροφορίας για μια ροή εργασίας, η οποία μπορεί να είναι η συνέχεια της λειτουργίας 5 από το προηγούμενο σύνολο λειτουργιών.
3. Γραφική παρουσίαση, τόσο της στατιστικής πληροφορίας μιας ροής εργασίας όσο και των στατιστικών πληροφοριών όλων των περιπτώσεών της.

Η στατιστική πληροφορία για μια ροή εργασίας μπορεί να περιλαμβάνει τις ιδιότητες: μέσος όρος χρόνου εκτέλεσης, μέσος όρος αξίας, μέσος όρος καταλαμβανόμενων πόρων, μέσος όρος κόμβων δικτύου που χρειάστηκαν ανά εκτέλεση, αξία εκτέλεσης περίπτωσης ανά χρόνο εκτέλεσης, ανά αριθμό καταλαμβανόμενων πόρων κ.τ.λ. Αντιθέτως, η στατιστική πληροφορία για μια περίπτωση ροής εργασίας είναι ο χρόνος εκτέλεσης, ο αριθμός καταλαμβανόμενων πόρων, ο αριθμός των κόμβων δικτύου που χρειάστηκαν, ο αριθμός των προειδοποιήσεων, ο αριθμός των λαθών, η πραγματική αξία εκτέλεσης κ.τ.λ. Η λειτουργία 1 θα γίνεται πιθανώς από έναν υπηρετητή, που θα περιμένει από το Σύστημα Εκτέλεσης Ροών Εργασίας να τον ειδοποιήσει. Οι λειτουργίες 2,3 θα επιτελούνται από διεπαφές χρήσης ή / και από τους προαναφερόμενους κειμενογράφους.

Θα πρέπει να αναφέρουμε ότι όλες οι παραπάνω μη γραφικές λειτουργίες θα υλοποιούνται από κάποια είδη υπηρετητών και έπειτα τα γραφικά εργαλεία (κειμενογράφοι και διεπαφές χρήσης) θα τις χρησιμοποιούν, επικοινωνώντας με τους υπηρετητές, για να προσφέρουν το κατάλληλο οπτικό ή μη αποτέλεσμα. Είναι προφανές, ότι τα γραφικά εργαλεία θα ελέγχουν τη σωστή ακολουθία των λειτουργιών, έτσι ώστε να επιτευχθεί κάποιος ή κάποιοι στόχοι του χρήστη, οδηγώντας τον σε επιτρεπτές και, άρα, σωστές κινήσεις.

Τα τεχνικά χαρακτηριστικά, που θα πρέπει να εκδηλώνει το σύστημά μας είναι τα εξής :

- Να βασίζεται και να χρησιμοποιεί την τεχνολογία του διαδικτύου.
- Να προσφέρει διαλειτουργικότητα ροών εργασίας.
- Να συμμορφώνεται με διεθνή πρότυπα.
- Να χρησιμοποιεί εξεζητημένα εργαλεία παραγωγής λογισμικού, όπως είναι η Java.
- Θα πρέπει να οριστεί μια γλώσσα ορισμού: ροών εργασίας και περιπτώσεων ροών εργασίας, που θα έχει αυστηρό λεξιλόγιο, δομή και σημασιολογία.
- Οι ορισμοί ροών εργασίας (και περιπτώσεων) θα γίνονται σύμφωνα με την παραπάνω γλώσσα και θα αποθηκεύονται σε φόρμες δεδομένων, που είναι δομημένες, εύκολα ανταλλάξιμες και έχουν προτυποποιηθεί (όπως είναι η XML [87]).
- Θα πρέπει να δημιουργηθεί ένας κειμενογράφος ροών εργασίας, που θα ορίζει γραφικά ροές εργασίας.
- Θα πρέπει να δημιουργηθεί ένας κειμενογράφος περιπτώσεων ροών εργασίας, που θα ορίζει γραφικά περιπτώσεις ροών εργασίας και θα αποτελεί λειτουργικό κομμάτι του κειμενογράφου ροών εργασίας.
- Και οι δύο παραπάνω κειμενογράφοι θα μπορούν να λειτουργούν off-line ή θα είναι σε μορφή Java Applets σε σελίδες του διαδικτύου.
- Ο κάθε προμηθευτής επιστημονικών αντικειμένων θα έχει δικαίωμα να εγκαταστήσει τοπικά στο δικό του σύστημα ρυθμίσεις-υλοποιήσεις του

συστήματος Κατανεμημένης Αποθήκευσης Ροών Εργασίας Επιστημονικών Αντικειμένων.

- Να χρειάζεται ελάχιστη διοίκηση, τόσο κατά την εγκατάσταση όσο και για τη σωστή λειτουργία και διατήρησή του.

Τέλος, η αρχιτεκτονική πάνω στην οποία θα στηρίζεται η όλη λειτουργία του συστήματός μας θα πρέπει να παρουσιάζει ιδιαίτερα και επιθυμητά χαρακτηριστικά, όπως :

- Να είναι ανοικτή (open).
- Να είναι κατανεμημένη.
- Να είναι κλιμακώσιμη.
- Να είναι αξιόπιστη.
- Να είναι ασφαλής.
- Να είναι lightweight.

5.3. Σχεδίαση του συστήματος

Η σχεδίαση του συστήματος είναι από τις πιο σημαντικές διαδικασίες κατασκευής ενός συστήματος. Η ολοκλήρωση της διαδικασίας της σχεδίασης καθορίζει ολόκληρη τη διαδικασία της υλοποίησης. Αν γίνουν λάθη στη σχεδίαση του συστήματος, θα διογκωθούν στην υλοποίηση και το σύστημά μας θα παρουσιάζει προβλήματα και δυσλειτουργίες. Οπότε, και η μόνη λύση θα είναι να ανασχεδιαστεί το σύστημα και να ξανα-υλοποιηθεί, πράγμα που συνεπάγεται χρονικές και οικονομικές επιβαρύνσεις.

Σύμφωνα, λοιπόν, με τα παραπάνω, η σχεδίαση του συστήματος έγινε προσεχτικά και με βάση τα αποτελέσματα της φάσης της ανάλυσης απαιτήσεων. Περιελάμβανε αφενός τη σχεδίαση μιας γλώσσας ορισμού ροών εργασίας και περιπτώσεων ροών εργασίας, και αφετέρου τον καθορισμό της αρχιτεκτονικής του συστήματος. Προφανώς, ο καθορισμός της αρχιτεκτονικής ενός συστήματος είναι η πεμπτούσια της σχεδίασης του συστήματος αυτού. Αλλά και η σχεδίαση μιας γλώσσας ορισμού ροών εργασίας, που είναι κατανοητή και εύχρηστη, θα βοηθήσει τους εμπειρογνώμονες χρήστες να διατυπώσουν τον ορισμό των επιστημονικών τους ροών εργασίας. Στις παρακάτω δύο υποενότητες θα περιγραφθούν και θα αναλυθούν τα δύο μέρη της σχεδίασής μας.

5.3.1. Σχεδίαση της γλώσσας ορισμού ροών εργασίας

5.3.1.1. Ορισμός μοντέλου ροής εργασίας

Αρχικά θα πρέπει να υπενθυμίσουμε στον αναγνώστη το μοντέλο της επιστημονικής εργασίας, όπως αυτό ορίστηκε στη φάση της σχεδίασης του συστήματος **ARION**, το οποίο φαίνεται και στο σχήμα 7. Σύμφωνα με αυτό το μοντέλο, η επιστημονική ροή εργασίας είναι ένας κατευθυνόμενος γράφος από αλληλοσυνδεόμενα έργα. Υπάρχουν δύο είδη σύνδεσης (κατευθυνόμενα βέλη): το απλό και το ειδικό. Το απλό καθορίζει αν κάποια έργα τρέχουν σειριακά ή παράλληλα. Δύο έργα τρέχουν σειριακά, όταν από το πρώτο έργο, που θα εκτελεστεί πρώτα, φεύγει ένα βέλος προς το δεύτερο έργο, που θα εκτελεστεί αμέσως μετά.

Πολλά έργα τρέχουν παράλληλα, όταν από κάποιο έργο, που εκτελείται νωρίτερα, φύγουν τόσα βέλη όσα είναι και ο αριθμός των παράλληλων έργων προς αυτά τα παράλληλα έργα. Το άλλο είδος σύνδεσης καθορίζει ότι μόλις ένα έργο εκτελεστεί (από αυτό φεύγει το ειδικό βέλος) θα γυρίσουμε πίσω στην ιστορία εκτέλεσης και θα επαναεκτελέσουμε ένα (ή πολλά) έργο (σε αυτό-ά καταλήγει το ειδικό βέλος). Κάθε έργο απαιτεί είσοδο και παράγει έξοδο, σύμφωνα με τις απαιτήσεις / προδιαγραφές του προγράμματος, που θα εκτελέσει. Τόσο η είσοδος όσο και η έξοδος αποτελούνται από μηδέν ή παραπάνω σύνολα δεδομένων και από μηδέν ή παραπάνω παραμέτρους.

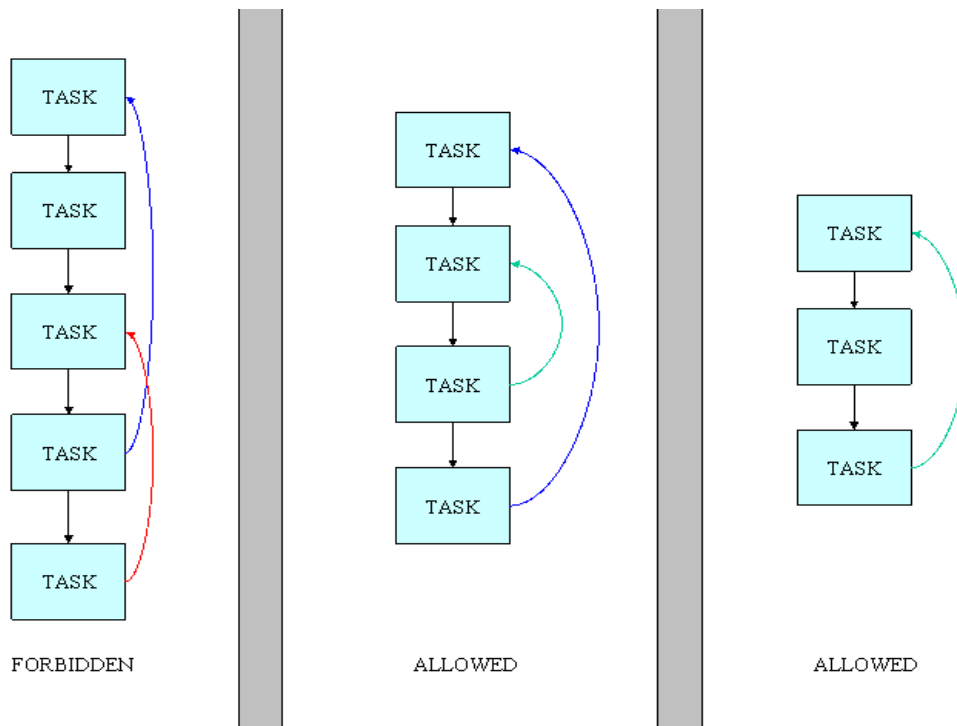
Προφανώς, ο γράφος της ροής εκτέλεσης περιέχει κύκλους, λόγω των ειδικών συνδέσεων. Αλλά τα έργα, που συνδέονται με απλό τύπο σύνδεσης, δεν θα πρέπει να σχηματίζουν κύκλο, διότι τότε η εκτέλεση της ροής εργασίας μπορεί να μην τελειώσει ποτέ. Άρα, το μοντέλο της επιστημονικής ροής εργασίας το κατατάσσουμε στην κατηγορία IPO-based, όπου, όμως, η ροή εργασίας είναι ένας κυκλικός γράφος αλληλοσυνδεόμενων έργων. Πρέπει να σημειωθεί ότι ορισμένα από τα στοιχεία του μοντέλου δεν είναι ικανοποιητικά.

Καταρχάς, η ροή εργασίας είναι ένας γράφος με δύο είδη σύνδεσης. Αν μια ροή εργασίας είναι πολύ μεγάλη, τότε η περιγραφή της θα ήταν μακροσκελής και σημασιολογικά / εννοιολογικά δυσνόητη. Είναι δύσκολο να κατανοηθεί από ποιο έργο αρχίζει η ροή εκτέλεσης και σε ποιο καταλήγει, ποια είναι ακριβώς η ροή εκτέλεσης και πότε ένα έργο προηγείται σίγουρα ενός άλλου. Θα ήταν καλό να υπήρχε μια ιεραρχία από ειδικά δομικά συστατικά / στοιχεία με ειδικά ονόματα, όπως ακολουθία, συγχρονισμένη παραλληλισμός, επανάληψη υπό όρους, που να ομαδοποιούν τα έργα και να μαρτυρούν τον τρόπο εκτέλεσής τους. Με αυτό τον τρόπο, οι μακροσκελείς περιγραφές ροών εργασίας θα ήταν πιο κατανοητές από τους χρήστες. Επιπροσθέτως, θα μοντελοποιούσαμε και κομμάτια της εκτέλεσης, που υπό όρους θα ήταν επαναληπτικά. Συνεπώς, το μοντέλο μας θα είχε περισσότερη δύναμη εκφραστικότητας.

Το δεύτερο στοιχείο, που δεν είναι ικανοποιητικό, είναι η ύπαρξη κύκλων μέσω του ειδικού βέλους (backtracking), που υποδηλώνουν ότι ο χρήστης δεν είναι ικανοποιημένος με κάποια αποτελέσματα και γυρίζει πίσω στην ιστορία της εκτέλεσης, για να επανα-ξεκινήσει ένα ήδη εκτελεσμένο έργο με διαφορετική είσοδο αυτή τη φορά. Γενικώς, οι κύκλοι πρέπει να αποφεύγονται, διότι εάν δεν εφαρμοστούν σωστά σε ένα γράφο μπορεί να προκαλέσουν αρκετά προβλήματα (που αρχίζουν από τη μη κατανόηση της ροής εκτέλεσης της ροής εργασίας) και καταλήγουν στην ατέρμονη εκτέλεση της ροής εργασίας. Για το λόγο αυτό, περιορίσαμε τη χρήση αυτής της ειδικής σύνδεσης με τον εξής τρόπο :

- Η ιστορία εκτέλεσης έργων μιας ροής εργασίας αποτελείται από ένα σύνολο έργων, στην πιο απλή περίπτωση. Το φαινόμενο του backtracking προκαλεί την επιστροφή από ένα πρόσφατο έργο σε ένα πιο παλιό, στην ιστορία εκτέλεσης. Το σύνολο των έργων, που βρίσκονται μεταξύ του πρόσφατου και του παλιού έργου στην ιστορία εκτέλεσης μαζί με αυτά τα δύο ακριανά έργα, το ονομάζουμε **backtracking set**. Το backtracking set, πρέπει να μην συνυπάρχει σε ένα γράφο με άλλα τέτοιου είδους σύνολα, που παρουσιάζουν κοινά μέλη με αυτό, εκτός και αν το ένα είναι υποσύνολο του άλλου. Με άλλα λόγια, θεωρούμε ότι τα σύνολα backtracking sets ενός γράφου δεν πρέπει να τέμνονται μεταξύ τους, αλλά μπορούν να είναι υποσύνολα το ένα του άλλου. Για να γίνουμε πιο σαφείς, θεωρούμε ότι ένα έργο δεν μπορεί να γυρίσει πίσω σε ένα πιο παλιό έργο σε εκτέλεση σε περίπτωση που ανήκει ήδη σε ένα backtracking set. Το παλιό αυτό έργο είναι το παλαιότερο από όλα τα έργα, που συμμετέχουν στο προαναφερόμενο backtracking set. Από την άλλη πλευρά, επιτρέπουμε σε ένα

έργο να γυρίσει πίσω σε ένα πιο παλιό έργο από αυτό, είτε όταν και τα δύο δεν ανήκουν σε ένα backtracking set είτε όταν και τα δύο ανήκουν στο ίδιο backtracking set. Οι τρεις περιπτώσεις που αναφέραμε επιδεικνύονται στο σχήμα 15.



Σχήμα 15 - Οι τρεις περιπτώσεις backtracking

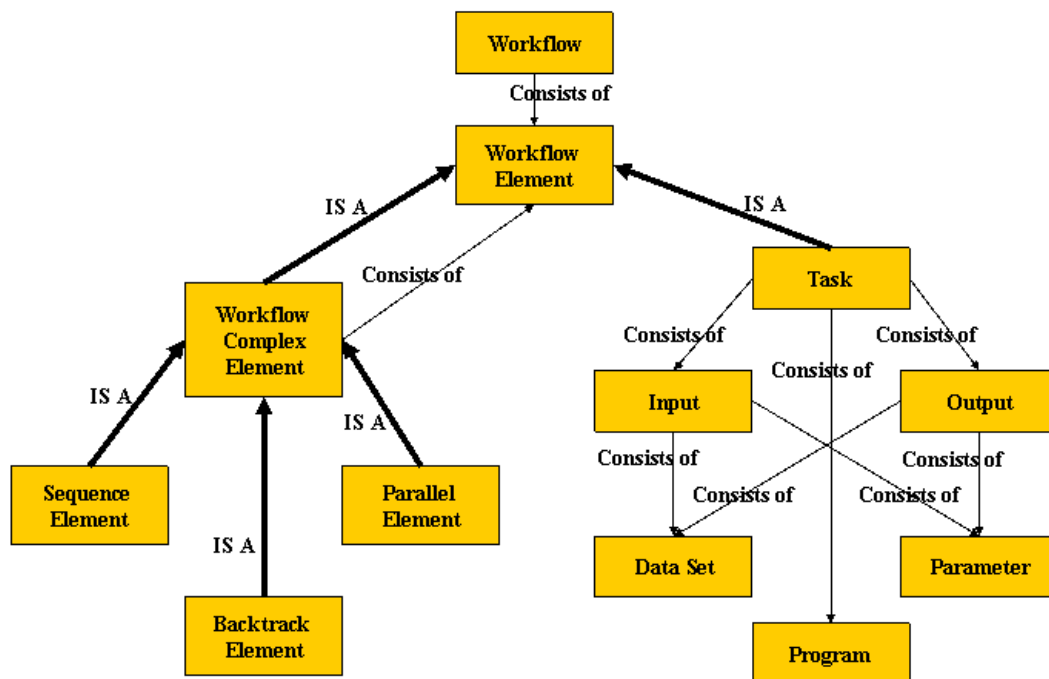
Συνεπώς, περιορίσαμε τη χρήση του backtracking, έτσι ώστε να εξασφαλίσουμε όχι μόνο ότι τα ενδιάμεσα αποτελέσματα ενός συνόλου από εργασίες, που θα τρέξουν σειριακά, είναι αυτά που οδηγούν το χρήστη να γυρίσει πίσω, αλλά και ότι αυτά δεν επηρεάζονται άμεσα από άλλα ενδιάμεσα αποτελέσματα. Μάλιστα το φαινόμενο του backtracking μπορεί να εφαρμοστεί στην προηγούμενη λύση του προβλήματος της εκφραστικότητας, με την προσθήκη ενός ακόμη δομικού στοιχείου με το όνομα backtrack, ενώ ο περιορισμός που επιβάλλαμε λύνεται μέσω της ιεραρχίας των δομικών στοιχείων, τα οποία μπορούν να εκφράσουν μια ροή εργασίας.

Ένας τρίτος λόγος για τη μη χρησιμοποίηση ενός γράφου είναι η δύσκολη επαλήθευσή του. Είναι δύσκολο να επαληθευθεί γρήγορα αν ένας γράφος δεν έχει κύκλους από έργα, που συνδέονται με απλές συνδέσεις, ή αν έχει κύκλους λόγω των ειδικών συνδέσεων.

Ύστερα από έρευνα πάνω σε διάφορες γλώσσες και μοντέλα ροών εργασίας καταλήξαμε στο μοντέλο ροής εργασίας της γλώσσας XRL. Σύμφωνα με αυτήν, μια ροή εργασίας αποτελείται από ένα μόνο στοιχείο ροής εργασίας. Αυτό μπορεί να είναι είτε ένα έργο, είτε ένα στοιχείο ακολουθίας, είτε ένα στοιχείο παραλληλισμού, είτε ένα στοιχείο επανάληψης υπό περιορισμό κ.τ.λ. Το κάθε στοιχείο ροής εργασίας, εκτός από το ίδιο το στοιχείο του έργου, μπορεί να αποτελείται από ένα ή περισσότερα έργα ανάλογα με τη σημασιολογία του. Όμως, τόσο ο ορισμός του τι είναι έργο όσο και η ύπαρξη γεγονότων, δεν ανήκουν στο μοντέλο της επιστημονική εργασίας. Εμείς κρατάμε τον παραπάνω ορισμό μέχρι το επίπεδο του ορισμού έργου και προσθέτουμε όλα αυτά, που ισχύουν για ένα επιστημονικό έργο. Το αποτέλεσμα φαίνεται καθαρά από την οντολογία του σχήματος 16.

Συνεπώς, μια ροή εργασίας είναι ένα δένδρο που έχει παιδί / αποτελείται από ένα στοιχείο ροής εργασίας. Το στοιχείο ροής εργασίας μπορεί να είναι είτε ένα έργο είτε κάτι πιο σύνθετο. Τα σύνθετα στοιχεία μπορεί να έχουν ένα ή παραπάνω στοιχεία ως παιδιά, ανάλογα με το τι σημασιολογία έχουν. Παραδείγματος χάριν, το σύνθετο στοιχείο της ακολουθίας υποδηλώνει ότι τα παιδιά του θα τρέξουν το ένα μετά το άλλο, από το πιο αριστερό στο πιο δεξιό στοιχείο. Το κάθε έργο ουσιαστικά αντιπροσωπεύει ένα πρόγραμμα, που έχει είσοδο και έξοδο μηδέν ή περισσότερα σύνολα δεδομένων και μηδέν ή περισσότερες παραμέτρους.

Scientific Workflow Model of ARION



Σχήμα 16 - Το νέο μοντέλο της επιστημονικής ροής εργασίας

5.3.1.2. Σχεδίαση γλώσσας ροής εργασίας

Η γλώσσα ορισμού επιστημονικών ροών εργασίας (ARION Workflow Language – AWL) βασίστηκε στη γλώσσα XRL. Ο κυριότερος λόγος είναι ότι το μοντέλο επιστημονικής ροής εργασίας ταίριαζε αρκετά με το μοντέλο της ροής εργασίας για τη XRL. Η γλώσσα XRL παρουσιάζει βασικά πλεονεκτήματα, που την κάνουν περιζήτητη στις σημερινές γλώσσες ορισμού ροών εργασίας. Τα πλεονεκτήματα αυτά είναι τα ακόλουθα :

- Είναι πλούσια σε δομικά στοιχεία (building blocks) και, άρα, έχει ισχυρή δύναμη εκφραστικότητας παρόμοια με αυτή των γλωσσών προγραμματισμού.
- Στηρίζεται στην τεχνολογία της XML. Με αυτό τον τρόπο η γλώσσα θα εκφράζεται μέσω ενός Document Type Definition - DTD και θα έχει αυστηρό

λεξιλόγιο, δομή και σημασιολογία. Οι ορισμοί ροών εργασίας θα γίνονται σε αρχεία XML, που θα υπακούουν στο DTD της γλώσσας.

- Εκφράζει και τις περιπτώσεις ροών εργασίας. Αυτό σημαίνει ότι μπορούμε να επεμβαίνουμε στο σχήμα, ακόμα και όταν οι περιπτώσεις του σχήματος τρέχουν. Επιπλέον, σημαίνει ότι μπορούμε να ορίζουμε αμέσως περιπτώσεις ροών εργασίας και ακολούθως να τις γενικεύουμε.
- Μπορούμε να την αντιστοιχίσουμε στα δίκτυα Petri Nets. Τα οφέλη από αυτήν την αντιστοίχιση είναι τα εξής:
 - *Τυπική σημασιολογία*: Η συμπεριφορά μιας ροής εργασίας, η οποία και ορίζεται από την XRL, προσδιορίζεται πλήρως. Επιπλέον, κάθε μορφής αμφισημία εξαλείφεται.
 - *Διαθεσιμότητα τεχνικών ανάλυσης και εργαλείων*: Με την αντιστοίχιση της XRL στα Petri Nets μπορούμε να χρησιμοποιήσουμε διάφορες τεχνικές και εργαλεία που έχουν αναπτυχθεί για αυτά. Υπάρχουν τρία είδη ανάλυσης:
 1. *Επικύρωση (validation)*: Έλεγχος αν η συμπεριφορά μιας ροής εργασίας είναι αναμενόμενη μέσω αλληλεπιδραστικής προσομοίωσης (interactive simulation).
 2. *Επαλήθευση (verification)*: Εδραίωση της ορθότητας μιας ροής εργασίας. Τεχνικές γραμμικής άλγεβρας μπορούν να χρησιμοποιηθούν για την επαλήθευση ιδιοτήτων, όπως place and transition invariants και reachability. Δυστυχώς, η τελευταία ιδιότητα δεν μπορεί να επαληθευθεί πλήρως για τη γλώσσα AWL, λόγω της ύπαρξης του δομικού στοιχείου 'backtrack'. Ο λόγος είναι ότι η χρήση αυτού του στοιχείου παρέχει τη δυνατότητα ατέρμονης εκτέλεσης κύκλου από έργα. Όμως, η ατέρμονη εκτέλεση μπορεί να προκληθεί μόνο με την παρέμβαση του χρήστη. Συνεπώς, δεν είναι βέβαιο αν η ροή εκτέλεσης μιας ροής εργασίας θα προχωρήσει ή θα 'κολλήσει' σε ένα επαναλαμβανόμενο κύκλο, εφόσον χρησιμοποιείται το δομικό στοιχείο 'backtrack' σε αυτήν.
 3. *Ανάλυση απόδοσης*: Επαλήθευση της ικανότητας για ικανοποίηση απαιτήσεων αναφορικά με το χρόνο απόκρισης, τα επίπεδα υπηρεσιών και τη χρησιμοποίηση πόρων. Ανάλυση της coverability ενός γράφου, έλεγχος μοντέλου και τεχνικές αναγωγής (reduction techniques) μπορούν να χρησιμοποιηθούν για την ανάλυση της δυναμικής συμπεριφοράς των Petri Nets. Ανάλυση προσομοίωσης και αλυσίδων Markov μπορούν να χρησιμοποιηθούν για την επαλήθευση της απόδοσης των Petri Nets.
 - *Ένα έτοιμο προς χρήση μοντέλο υλοποίησης*: Είναι προσιτή η υλοποίηση μιας μηχανής ροών εργασίας, η οποία και θα βασίζεται στις ήδη υπάρχουσες μηχανές των Petri Net εργαλείων προσομοίωσης και enactment.
 - *Γνώση για τη δύναμη εκφραστικότητας της XRL*: Η δύναμη εκφραστικότητας της XRL (πλην των ιδιοτήτων των δεδομένων) συγκρίνεται με την αντίστοιχη των Petri Nets. Συνεπώς, είναι προσιτή η προσομοίωση αφενός μηχανών κατάστασης (state machines) και αφετέρου μαρκαρισμένων γράφων (marked graphs). Επιπροσθέτως, με τη χρήση ενός μηχανισμού γεγονότων, είναι δυνατή η δημιουργία non-free δομικών στοιχείων.

Οι επεμβάσεις, που κάναμε στο τρόπο έκφρασης της γλώσσας XRL (στο DTD), ήταν λίγες και δεν επηρέασαν τα γενικά χαρακτηριστικά της. Και αυτό, διότι δεν αλλάξαμε τα κύρια αντικείμενα της μοντελοποίησης και τις σχέσεις, που έχουν μεταξύ τους. Πρέπει να σημειωθεί, ότι για τη νέα γλώσσα δεν έχει ακόμα διευκρινισθεί αν μπορεί να περιγράψει περιπτώσεις ροών εργασίας. Συνεπώς, η

πλειοψηφία αυτών των χαρακτηριστικών ανάγεται και στη γλώσσα ορισμού επιστημονικών ροών εργασίας. Το αρχικό DTD της γλώσσας της XRL βρίσκεται στο παράρτημα Α. Το τελικό DTD της δικής μας γλώσσας AWL βρίσκεται στο παράρτημα Β. Οι επεμβάσεις αφορούσαν τα εξής :

- Προσθήκη ενός δομικού στοιχείου ‘backtrack’ για την περιγραφή του φαινομένου του backtracking.
- Προσθήκη ιδιοτήτων στο στοιχείο ‘route’, το οποίο και μετονομάζεται σε ‘workflow’, και περιέχει τις γενικές ιδιότητες μιας ροής εργασίας.
- Μετονομασία του στοιχείου ‘route_element’ σε ‘workflow_element’.
- Προσθήκη και αφαίρεση ιδιοτήτων του στοιχείου ‘έργο’ και περαιτέρω ορισμός του, ώστε να ταιριάζει καλύτερα στον αντίστοιχο ορισμό του επιστημονικού έργου.

Στο υπόλοιπο αυτής της ενότητας θα περιγραφούν τα ποικίλα δομικά μέρη δρομολόγησης (routing constructs), που υπηρετούν ως κατασκευαστικά στηρίγματα (building blocks), οι ιδιότητές τους, καθώς και άλλα αντικείμενα μοντελοποίησης, όπως το έργο μαζί με τις ιδιότητές του. Στο τέλος αυτής της ενότητας θα αποδειχθεί ότι η γλώσσα AWL μπορεί να περιγράψει περιπτώσεις επιστημονικών ροών εργασίας.

Προτού ξεκινήσει η περιγραφή των δομικών μερών, θα πρέπει να τονιστεί ότι – όπως και σε κάθε XML περιγραφή – κάθε στοιχείο αρχίζει με ένα δομικό μέρος, όπως είναι το <element1> (start tag) και τελειώνει με </element1> (end tag), όπου το *element1* είναι το όνομα του στοιχείου.

• **Workflow (Ροή Εργασίας)**

Το σχήμα πρέπει να αρχίσει με ένα <workflow> start-tag και να τελειώσει με ένα </workflow> end-tag. Το workflow (ροή εργασίας) αποτελείται από ένα παιδί <workflow_element>, που θα περιγραφθεί ακριβώς μετά. Το στοιχείο workflow περιέχει τις εξής ιδιότητες :

ID: Το μοναδικό ID της ροής εργασίας.

Name: Το όνομα της ροής εργασίας.

Created_by: Ο δημιουργός του προσδιορισμού της ροής εργασίας.

Date: Η ημερομηνία της δημιουργίας του προσδιορισμού.

Description: Μια μικρή περιγραφή της ροής εργασίας.

Roles: Οι ρόλοι, που θα έχουν πρόσβαση σε αυτή τη ροή εργασίας.

Rights: Τα δικαιώματα, που θα έχουν οι προαναφερόμενοι ρόλοι.

Πρέπει να τονιστεί ότι εάν μια ιδιότητα μπορεί να πάρει πολλαπλές τιμές, τότε αυτές οι τιμές θα έχουν την μορφή : ‘τιμή1, τιμή2 , τιμή3 ... , τιμήN’.

• **Workflow_element (Στοιχείο ροής εργασίας)**

Το *workflow_element* ορίζεται ως εξής :

```
<!ENTITY % workflow_element "task | sequence |  
any_sequence | choice | condition | back_track |  
parallel_sync | parallel_no_sync | parallel_part_sync |  
wait_all | wait_any | while_do | stop | terminate">
```

Συνεπώς, το *workflow_element* μπορεί να είναι :

1. Task (Έργο): Το έργο που θα εκτελεστεί.
2. Sequence (Ακολουθία): Ένα σύνολο από έργα, που πρέπει να εκτελεστούν με μια ακολουθιακή σειρά, σχηματίζουν μια *sequence*.
3. Any_sequence (Οποιαδήποτε Ακολουθία): Ένα σύνολο από έργα, που πρέπει να εκτελεστούν σε οποιαδήποτε σειρά, σχηματίζουν μια *any_sequence*.
4. Choice (Επιλογή): Οποιαδήποτε έργο, που ανήκει σε ένα σύνολο έργων, ονομάζεται *choice*.
5. Condition (Συνθήκη): Έλεγχος μιας συνθήκης και προσδιορισμός του επόμενου βήματος – έργου, βάση του αποτελέσματος.
6. Parallel_sync (Συγχρονισμένος παραλληλισμός): Δημιουργία πολλαπλών *workflow_element* και αργότερα ένωσή τους.
7. Parallel_no_sync (Μη συγχρονισμένος παραλληλισμός): Δημιουργία πολλαπλών *workflow_element* και συνέχιση της ροής εκτέλεσης.
8. Parallel_part_sync (Μερικώς συγχρονισμένος παραλληλισμός): Δημιουργία πολλαπλών *workflow_element*, κάποια από τα οποία έπειτα πρέπει να ενωθούν.
9. Wait_all (Αναμονή Όλων): Εισαγωγή ενός βήματος αναμονής για την ολοκλήρωση μιας ομάδας από γεγονότα.
10. Wait_any (Αναμονή Οποιοδήποτε): Εισαγωγή ενός βήματος αναμονής για την πρώτη ολοκλήρωση οποιουδήποτε από τα γεγονότα μιας συγκεκριμένης ομάδας.
11. While_do (Ανακύκλωση Μέχρι): Δίνει τη δυνατότητα για ανακύκλωση ενός *workflow_element* όσο μια συνθήκη αποτιμάται ως αληθινή.
12. Back_track (Πισωγύρισμα): Το τελευταίο έργο ενός *workflow_element* μπορεί να οδηγήσει την εκτέλεση της ροής εργασίας στο πρώτο έργο του, κατόπιν παρεμβάσεως του χρήστη.
13. Stop (Διακοπή): Τέλος της εκτέλεσης ενός συγκεκριμένου μονοπατιού μιας περίπτωσης της ροής εργασίας.
14. Terminate (Τερματισμός): Τερματισμός εκτέλεσης αυτής της περίπτωσης ροής εργασίας.

- **Task (Έργο)**

Το *task* είναι ένας γενικός όρος για ένα έργο, βήμα ή δράση. Οι επόμενες ιδιότητες συσχετίζονται με το *task* :

ID: Το μοναδικό ID του έργου.

AgencyPort: Το port όπου ένας πράκτορας έχει τοποθετηθεί (είναι πληροφορία που σχετίζεται με τη μηχανή ροής εργασίας).

Result: Το αποτέλεσμα ενός έργου ως τιμή ενός αλφαριθμητικού.

Status: Η κατάσταση του έργου (ready – έτοιμο, running – εκτελείται, enabled – ενεργοποιημένο, disabled – απενεργοποιημένο, aborted – ματαιωμένο)

Start_time: Η ακριβής ημερομηνία έναρξης του έργου.

End_time: Η ακριβής ημερομηνία περάτωσης του έργου.

Max_run_time: Ο μέγιστος χρόνος εκτέλεσης ενός έργου, που προκύπτει έπειτα από πολλαπλές εκτελέσεις.

Notify: Διευθύνσεις ηλεκτρονικού ταχυδρομείου για συμβαλλόμενους, που πρέπει να ενημερωθούν μετά την περάτωση του έργου.

Το έργο, όπως ορίζει το μοντέλο της επιστημονικής ροής εργασίας, έχει ένα <input> και ένα <output> στοιχείο. Και τα δυο στοιχεία δεν κατέχουν κάποια ιδιότητα, αλλά περιέχουν μηδέν ή περισσότερα <data-set> στοιχεία και μηδέν ή περισσότερα <parameter> στοιχεία. Το έργο, επίσης, έχει ένα στοιχείο <program>, το οποίο περιγράφει το πραγματικό πρόγραμμα, που θα εκτελεστεί ως ένα βήμα μιας ροής εργασίας.

- **Data-set (Σύνολο Δεδομένων)**

Το στοιχείο *data-set* αναφέρεται σε ένα συγκεκριμένο αρχείο στο λειτουργικό σύστημα ενός προμηθευτή συνόλων δεδομένων (αν ορίζουμε μια περίπτωση ροής εργασίας) ή σε ένα οποιαδήποτε σύνολο δεδομένων κάποιου τύπου (αν ορίζουμε μια ροή εργασίας), το οποίο μπορεί να χρησιμοποιηθεί ως είσοδος ή έξοδος σε ένα πρόγραμμα. Αυτό έχει τις εξής ιδιότητες :

ID: Το μοναδικό ID του συνόλου δεδομένων.

Ref: Το ID ενός συνόλου δεδομένων, που έχει ήδη οριστεί στο XML αρχείο.

Name: Το όνομα του συνόλου δεδομένων.

URI: Το Unique Resource Identifier ενός συγκεκριμένου συνόλου δεδομένων.

Το URI παίρνει τιμή αν αναφερόμαστε στη διεύθυνση ενός πραγματικού συνόλου δεδομένων και όχι αν αναφερόμαστε σε ένα γενικό σύνολο δεδομένων κάποιου τύπου. Το URI μπορεί να είναι το Unique Resource Locator – URL ή ένα μονοπάτι στο λειτουργικό σύστημα του υπολογιστή, στον οποίο βρίσκεται αυτό το σύνολο δεδομένων.

Hostname: Το όνομα του υπολογιστή στον οποίο είναι αποθηκευμένο ένα συγκεκριμένο σύνολο δεδομένων.

Type: Ο τύπος του συνόλου δεδομένων. Ένα σύνολο δεδομένων μπορεί να είναι γενικό (άρα αναφέρεται σε κλάση από σύνολα δεδομένων και αντιστοιχεί στην οντολογία του ARION) ή ειδικό (άρα αναφέρεται σε περίπτωση της κλάσης του και, συνεπώς, ορίζεται η τοποθεσία που είναι αποθηκευμένο), όταν χρησιμοποιείται ως είσοδος σε ένα πρόγραμμα. Μπορεί, επίσης, να είναι παραγόμενο ή παραγόμενο-αποθηκευμένο (ορίζεται η τοποθεσία της αποθήκευσης του), στην περίπτωση που χρησιμοποιείται ως έξοδος σε ένα πρόγραμμα.

Ontology_name: Το όνομα της οντολογία της περιοχής, όπου ανήκει η κλάση του συνόλου δεδομένων.

Ontology_uri: Το URI της παραπάνω οντολογίας.

Classname: Το όνομα της κλάσης / οντότητας στην οποία ανήκει το σύνολο δεδομένων.

- **Parameter (Παράμετρος)**

Το στοιχείο *parameter* αναφέρεται στην παράμετρο του προγράμματος ενός έργου. Περιλαμβάνει τις ακόλουθες ιδιότητες :

Name: Το όνομα της παραμέτρου.

Type: Ο τύπος της τιμής της παραμέτρου (integer – ακέραιος , double – δεκαδικός διπλής ακρίβειας, string – αλφαριθμητικό, boolean , no-value – που δεν παίρνει τιμή, special – ειδική). Οι παράμετροι no-value δεν παίρνουν καθόλου τιμές. Οι ειδικές παράμετροι είναι υποχρεωτικές παράμετροι, τύπου τιμών αλφαριθμητικού, και δείχνουν, μέσω της ιδιότητας Ref, το παραγόμενο σύνολο δεδομένων του

προγράμματος στο οποίο ανήκουν. Χρησιμοποιούμε τις ειδικές παραμέτρους, όταν ένα hard-wired πρόγραμμα αποθηκεύει ένα σύνολο δεδομένων με συγκεκριμένο όνομα αρχείου. Ο χρήστης καλείται να δώσει ένα διαφορετικό όνομα στο αρχείο αυτό, ώστε να επιτρέψει την παράλληλη εκτέλεση του συγκεκριμένου προγράμματος και από άλλες περιπτώσεις ροών εργασίας άλλων χρηστών.

Ref: είναι το ID του παραγόμενου συνόλου δεδομένων εκείνου του προγράμματος στο οποίο ανήκει η συγκεκριμένη παράμετρος. Το σύνολο δεδομένων αποθηκεύεται πάντοτε με ένα συγκεκριμένο όνομα αρχείου.

Syntax: Ο τρόπος σύνταξης της παραμέτρου στη γραμμή εντολών.

Default_Value: Η αρχική τιμή της παραμέτρου.

Description: Μια σύντομη περιγραφή του λόγου ύπαρξης της παραμέτρου.

Optional: Αν είναι προαιρετική ή υποχρεωτική. Οι υποχρεωτικές παράμετροι πρέπει να παίρνουν τιμές από τους χρήστες, κατά τη διάρκεια της εκτέλεσης του προγράμματος στο οποίο ανήκουν.

Type_of_value_constraint: Ο τύπος περιορισμού τιμών, που ισχύει για τη παράμετρο. Αν είναι free, τότε παίρνει οποιαδήποτε τιμή. Αν είναι range, τότε παίρνει τιμές από ένα κλειστό σύνολο της μορφής [α,β]. Αν είναι discrete, τότε παίρνει ακριβώς ορισμένες τιμές και όχι άλλες. Και τα τρία είδη περιορισμών μπορούν να εφαρμοστούν σε ακεραίους και δεκαδικούς διπλής ακρίβειας. Το πρώτο και το τρίτο ισχύει για τα αλφαριθμητικά. Οι boolean παράμετροι μπορούν να πάρουν τις εξής τιμές: yes, no, true, false. Οι ειδικές παράμετροι είναι free of value.

Value_constraint: Είναι η τιμή του περιορισμού τιμών. Αν ο περιορισμός τιμών είναι range, τότε η τιμή της ιδιότητας αυτής πρέπει να είναι της μορφής [α,β]. Αν ο περιορισμός τιμών είναι discrete, τότε η τιμή της ιδιότητας αυτής πρέπει να είναι αποκλειστικά της μορφής 'τιμή1, τιμή2, ..., τιμήN'.

- **Program (Το πρόγραμμα)**

Το στοιχείο *program* αναφέρεται στο αληθινό πρόγραμμα, που θα εκτελεστεί ως ένα βήμα / έργο μιας ροής εργασίας. Όλοι οι ορισμοί των ιδιοτήτων του είναι οι παρακάτω :

Name: Το όνομα του προγράμματος.

ID: Το ID του προγράμματος.

Ref: Αν αυτό το πρόγραμμα ανάγεται σε ένα άλλο πρόγραμμα, που έχει ήδη οριστεί, τότε αναφέρει το ID του ήδη ορισμένου προγράμματος.

Sync: Αν αυτό το πρόγραμμα μπορεί να εκτελεστεί μόνο σε μια περίπτωση ροής εργασίας και όχι σε πολλές παράλληλα. Δηλαδή, το πρόγραμμα αυτό εκτελείται κάθε φορά από ένα μόνο χρήστη.

Hostname: Το όνομα του υπολογιστή στον οποίο είναι αποθηκευμένο το εκτελέσιμο αρχείο του προγράμματος.

URI: είναι ή το μονοπάτι του λειτουργικού συστήματος του υπολογιστή στο οποίο βρίσκεται το πρόγραμμα ή το URL του προγράμματος.

Heavy: Αν αυτό το πρόγραμμα χρειάζεται, γενικά, αρκετό χρόνο για να ολοκληρωθεί, τότε η τιμή αυτής της ιδιότητας είναι 'Y'.

Mobile: Αν αυτό το πρόγραμμα βρίσκεται σε άλλο υπολογιστή από αυτόν του πράκτορα εκτέλεσης, τότε η τιμή αυτής της ιδιότητας είναι 'Y'.

Simulation: Αν αυτό το πρόγραμμα χρησιμοποιείται ως εργαλείο προσομοίωσης, τότε η τιμή αυτής της ιδιότητας είναι 'Y'.

Visualization: Αν αυτό το πρόγραμμα χρησιμοποιείται για την οπτικοποίηση αποτελεσμάτων, τότε η τιμή αυτής της ιδιότητας είναι 'Y'.

Statistical: Αν αυτό το πρόγραμμα κάνει μια στατιστική ανάλυση σε κάποιο φαινόμενο, τότε η τιμή αυτής της ιδιότητας είναι 'Y'.

Validation: Αν αυτό το πρόγραμμα επαληθεύει κάτι, τότε η τιμή αυτής της ιδιότητας είναι 'Y'.

Database_retrieval: Αν αυτό το πρόγραμμα χρησιμοποιείται για την εξόρυξη πληροφορίας από μια βάση δεδομένων, τότε η τιμή αυτής της ιδιότητας είναι 'Y'.

Output_filter: Αν αυτό το πρόγραμμα είναι ένα φίλτρο εξόδου, τότε η τιμή αυτής της ιδιότητας είναι 'Y'.

Transformation: Αν αυτό το πρόγραμμα εκτελεί κάποιου είδους μεταμόρφωση, τότε η τιμή αυτής της ιδιότητας είναι 'Y'.

- **Sequence (Ακολουθία)**

Το στοιχείο *sequence* αντιπροσωπεύει το πιο απλό σχέδιο δρομολόγησης. Είναι ένας τρόπος ενοποίησης πολλαπλών βημάτων μέσα σε ένα σύμπλεγμα. Παρακάτω παραθέτουμε ένα απλό παράδειγμα μια ακολουθιακής δρομολόγησης με τη χρησιμοποίηση της AWL :

```
<sequence>
  <task name="task1" ...>
    ...
  </task>
  <task name="task2" ...>
    ...
  </task>
  <task name="task3" ...>
    ...
  </task>
</sequence>
```

Σε αυτό το παράδειγμα, τρία έργα παρατάσσονται μαζί μέσα σε ένα στοιχείο *sequence* . Συνεπώς, το στοιχείο *sequence* είναι ένας λογικός τρόπος συγκρότησης / ομαδοποίησης ακολουθιακών έργων.

- **Parallel constructs (Παράλληλα δομικά στοιχεία)**

Υπάρχουν τρεις τύποι δομικών στοιχείων, ο οποίοι σχετίζονται μεταξύ τους και είναι οι εξής : *parallel_sync*, *parallel_no_sync*, *parallel_part_sync*.

Σε όλα αυτά τα δομικά στοιχεία, η κύρια πλευρά είναι ότι επιτρέπουν τη δημιουργία υπο-ροών εργασίας, οι οποίες μπορούν να προχωρήσουν ανεξάρτητα. Το στοιχείο *parallel_sync* αντιστοιχεί στην παραδοσιακή ιδέα της παράλληλης δρομολόγησης. Αρχικά, ένας αριθμός από κλαδιά ενεργοποιούνται (το ονομαζόμενο AND-split). Έπειτα, τα παράλληλα κλαδιά εκτελούνται ανεξάρτητα. Στο τέλος, τα παράλληλα κλαδιά συγχρονίζονται (το ονομαζόμενο AND-join). Σε ένα στοιχείο *parallel_sync*, οι υπο-ροές εργασίας πρέπει να συνδυαστούν μαζί στο τέλος αυτού του στοιχείου.

Από την άλλη πλευρά, σε ένα *parallel_no_sync* στοιχείο, οι υπο-ροές εργασίας δεν χρειάζεται να συνδυαστούν μαζί στο τέλος αυτού του στοιχείου. Εναλλακτικά, σε ένα *parallel_part_sync* στοιχείο, μόνο κάποιες υπο-ροές εργασίας πρέπει να συνδυαστούν, ο αριθμός των οποίων ορίζεται ως τιμή της ιδιότητας 'number'.

Παρακάτω παραθέτουμε ένα παράδειγμα ενός *parallel_part_sync* στοιχείου :

```
<parallel_part_sync number="2">
  <sequence>
    <task name="send-to-vp1" ...>
      <input>
        <data-set URI="invoice.doc" .../>
      </input>
    </task>
    <task name="vp1" ...>
      <input>
        <data-set URI="invoice.doc" .../>
      </input>
    </task>
  </sequence>
  <sequence>
    <task name="send-to-vp2" ...>
      <input>
        <data-set URI="invoice.doc" .../>
      </input>
    </task>
    <task name="vp2" ...>
      <input>
        <data-set URI="invoice.doc" .../>
      </input>
    </task>
  </sequence>
  <sequence>
    <task name="send-to-vp3" ...>
      <input>
        <data-set URI="invoice.doc" .../>
      </input>
    </task>
    <task name="vp3" ...>
      <input>
        <data-set URI="invoice.doc" .../>
      </input>
    </task>
  </sequence>
</parallel_part_sync>
```

Σε αυτό το παράδειγμα, μόνο δύο από τους τρεις αντιπροέδρους μιας εταιρείας απαιτούνται για να εγκριθεί ένα τιμολόγιο. Όταν εγκριθεί, η ροή εργασίας μπορεί να συνεχίσει στην επόμενη φάση. Δεν έχει σημασία ποιος από τους αντιπροέδρους έδωσε την έγκριση. Για να δοθεί έμφαση στο φώλιασμα των δομικών στοιχείων, τα έργα έγκρισης: vp1, vp2, vp3 προηγούνται από τα έργα αποστολής τιμολογίου: send-

to-vp1, send-to-vp2, send-to-vp3. Πρέπει να τονιστεί, ότι αν και η ροή εργασίας μπορεί να προχωρήσει στην επόμενη φάση, η τρίτη έγκριση μπορεί να εκτελεστεί.

- **Condition checking and looping (Ελεγχος Συνθηκών και Ανακύκλωση)**

Το στοιχείο *condition* χρησιμοποιείται για τον έλεγχο μιας γενικής δήλωσης συνθήκης και για να καθοδηγήσει τη ροή εργασίας, βασιζόμενο στο boolean αποτέλεσμα. Αυτό το στοιχείο έχει μια ιδιότητα, που ονομάζεται 'condition' και η οποία περιγράφει τη συνθήκη που πρόκειται να ελεγχθεί. Η δομή του στοιχείου αυτού φαίνεται ακολούθως :

```
<condition condition="result='ok' ">
  <true>
    <task name="send_confirmation" ...>
      ...
    </task>
  </true>
  <false>
    <task name="notify_customer" ...>
      ...
    </task>
  </false>
</condition>
```

Όπως φαίνεται παραπάνω, τα στοιχεία *true* και *false* συσχετίζονται σε σύνδεση με το στοιχείο *condition* και τουλάχιστον ένα από αυτά πρέπει να είναι παρόν.

Το δομικό στοιχείο *while_do* εισάγεται, για να επιτρέψει την επανάληψη στοιχείων δρομολόγησης ή βημάτων / έργων. Περιλαμβάνει μια ιδιότητα με όνομα 'condition', η οποία περιγράφει μια συνθήκη. Τα στοιχεία, που εσωκλείονται μέσα στο στοιχείο *while_do*, εκτελούνται συνεχώς όσο αυτή η συνθήκη αποτιμάται σε αληθινή.

Παρακάτω παραθέτουμε ένα παράδειγμα :

```
<while_do condition="vp3.result='Null' ">
  <task name="vp3" notify=vp3@xyz.com ...>
    ...
  </task>
</while_do>
```

- **Back_track (Πισωγύρισμα)**

Το στοιχείο *back_track* χρησιμοποιείται, για να ενώσει το τελευταίο με το πρώτο έργο μιας ακολουθίας έργων, με την έννοια ότι ο χρήστης μπορεί να παρέμβει όταν τελειώσει το τελευταίο έργο και να οδηγήσει τη ροή της εκτέλεσης πάλι πίσω στο πρώτο έργο της ακολουθίας. Παραθέτουμε ένα παράδειγμα :

```
<back_track>
  <sequence>
```

```

    <task name="Prepare Program Input" ...>
    ...
  </task>
  <task name="Execute Program" ...>
  ...
  </task>
</sequence>
</back_track>

```

Σύμφωνα με το παραπάνω παράδειγμα, αν ο χρήστης, μόλις τελειώσει το δεύτερο έργο που ουσιαστικά εκτελεί ένα πρόγραμμα, δεν είναι ικανοποιημένος με το αποτέλεσμα, τότε μπορεί να γυρίσω πίσω σε εκείνο το έργο, που ετοιμάζει την είσοδο στο πρόγραμμα του τελευταίου έργου. Μάλιστα, αυτή η διαδικασία μπορεί να συνεχίζεται επί άπειρον, αν ο χρήστης δεν είναι ικανοποιημένος με τα αποτελέσματα του προγράμματος του δεύτερου έργου. Για αυτό τον λόγο, και το φαινόμενο του backtracking μοιάζει με αυτό της ανακύκλωσης, διότι έχουν ακριβώς το ίδιο αποτέλεσμα. Μόνο, που η ανακύκλωση γίνεται αυτόματα, ενώ στο backtracking πρέπει να επέμβει ο χρήστης και να ζητήσει να επανα-εκτελεστεί ένα κομμάτι της ροής εργασίας.

Για να ολοκληρωθεί η ανάλυση της σχεδίασης της γλώσσας επιστημονικών ροών εργασίας, θα πρέπει να αποδειχθεί ότι η γλώσσα αυτή μπορεί να χρησιμοποιηθεί και για τον ορισμό περιπτώσεων ροών εργασίας. Μια επιστημονική ροή εργασίας θα περιέχει αφηρημένα σύνολα δεδομένων, ως είσοδο στα προγράμματα των έργων της. Με τον όρο αφηρημένο σύνολο δεδομένων εννοούμε μια ομάδα από σύνολα δεδομένων, που το καθένα από αυτά είναι περίπτωση μιας οντότητας / έννοιας της επιστημονικής οντολογίας. Επομένως, για τα αφηρημένα σύνολα δεδομένων δεν διευκρινίζεται που βρίσκονται. Απεναντίας, μια περίπτωση ροής εργασίας θα πρέπει να έχει όλα τα σύνολα δεδομένων, που είναι είσοδος σε συγκεκριμένα προγράμματα έργων. Με τον όρο συγκεκριμένο σύνολο δεδομένων εννοούμε ένα σύνολο δεδομένων, που είναι περίπτωση μιας οντότητας της επιστημονικής οντολογίας και γνωρίζουμε με ακρίβεια που έχει γίνει η αποθήκευση. Η γλώσσα AWL προβλέπει την ύπαρξη τόσο οντολογικών πληροφοριών όσο και πληροφοριών τοποθεσίας (υπολογιστή) στον ορισμό ενός συνόλου δεδομένων. Στον ορισμό μιας ροής εργασίας, κάποιοι ορισμοί συνόλων δεδομένων, τα οποία και χρησιμοποιούνται ως είσοδος σε προγράμματα, δεν θα έχουν συμπληρωμένες τις πληροφορίες τοποθεσίας. Αντίθετα, στον ορισμό μιας περίπτωσης ροής εργασίας, οι ορισμοί όλων των συνόλων δεδομένων εισόδου θα πρέπει να έχουν συμπληρωμένες τις πληροφορίες τοποθεσίας. Εν κατακλείδι, αποδείξαμε ότι η γλώσσα περιγραφής επιστημονικών ροών εργασίας μπορεί να χρησιμοποιηθεί και για τον ορισμό περιπτώσεων επιστημονικών ροών εργασίας.

5.3.2. Αρχιτεκτονική του συστήματος

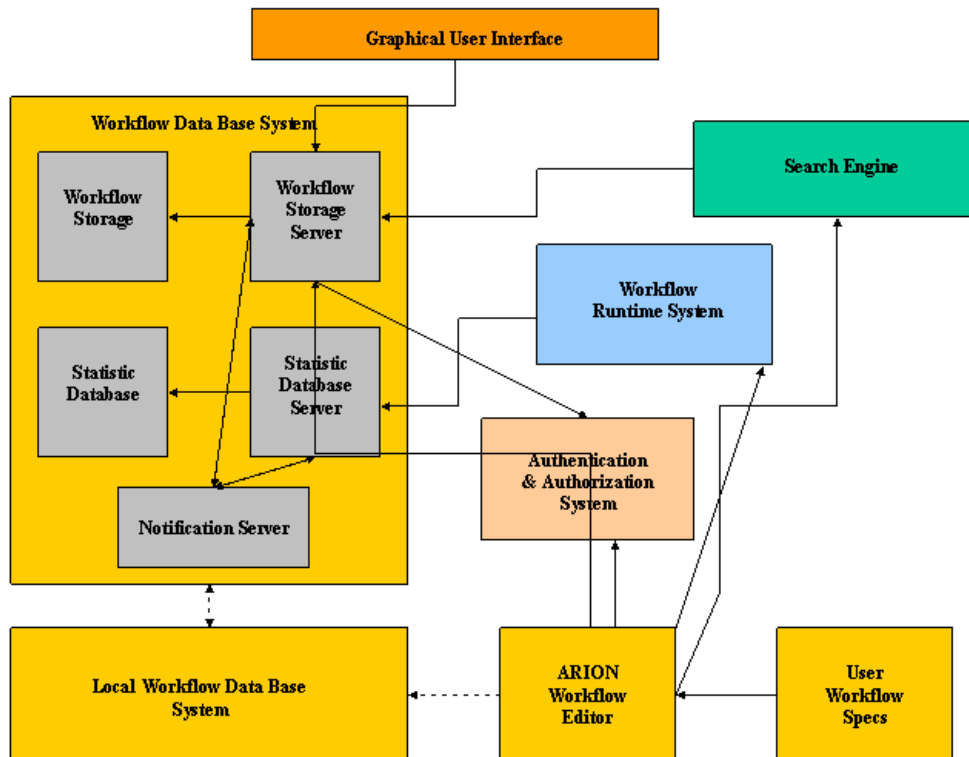
Η σχεδίαση της αρχιτεκτονικής ενός συστήματος είναι μια πολύ σημαντική διαδικασία. Οι λόγοι είναι δύο. Πρώτον, διότι διαχωρίζει τις λειτουργίες ενός συστήματος και τις αναθέτει σε διάφορα συστατικά μέρη (components). Δεύτερον, διότι ο σωστός διαχωρισμός των λειτουργιών του συστήματος, θα έχει ευεργετικό αντίκτυπο τόσο στην απόδοση, αξιοπιστία και ασφάλεια του συστήματος όσο και στη διαδικασία της υλοποίησής του. Ο πρόχειρος σχεδιασμός της αρχιτεκτονικής ενός

συστήματος, από την άλλη πλευρά, θα δημιουργήσει σημαντικά προβλήματα στην υλοποίηση του και θα μετατρέψει το σύστημα σε ένα μη αποδοτικό, αναξιόπιστο και ανασφαλές προϊόν.

Το Σύστημα Κατανεμημένης Αποθήκευσης Ροών Εργασίας Επιστημονικών Αντικειμένων - **Κ.Α.Ρ.Ε.Ε.Α** - μοιάζει σε λειτουργικότητα με το κατασκευαστικό μέρος (build-time component) ενός συστήματος διαχείρισης ροών εργασίας. Όμως, είναι και κάτι παραπάνω από αυτό. Πιο συγκεκριμένα, περιέχει μια βάση δεδομένων μέσα στην οποία αποθηκεύονται ορισμοί ροών εργασίας και ορισμοί περιπτώσεων ροών εργασίας, καθώς και στατιστικά δεδομένα, που αφορούν την εκτέλεση μιας περίπτωσης ροής εργασίας. Επιπλέον, θέτει σε εφαρμογή την εκτέλεση μιας περίπτωσης ροής εργασίας – αυτή η πράξη καθοδηγείται από ένα χρήστη, ο οποίος μετατρέπει τον ορισμό ροής εργασίας σε ορισμό περίπτωσης ροής εργασίας με τη βοήθεια φορμών του διαδικτύου (web forms) – μέσω της αποστολής του ορισμού της συγκεκριμένης περίπτωσης ροής εργασίας στο **Σύστημα Εκτέλεσης Ροών Εργασίας**. Η επιπλέον λειτουργικότητα, που εκθέτει το σύστημά μας, το αναδεικνύει ως ένα κρίσιμο μέρος του συστήματος **ARION**.

Προτού αρχίσει η περιγραφή της αρχιτεκτονικής του συστήματός μας, πρέπει να τονιστεί ότι το σύστημά μας διαχωρίζεται σε ένα **Κεντρικό Σύστημα Κ.Α.Ρ.Ε.Ε.Α**, που είναι τοποθετημένο στο κύριο σύστημα **ARION**, και σε ένα **Τοπικό Σύστημα Κ.Α.Ρ.Ε.Ε.Α**, που είναι τοποθετημένο στον τελικό κόμβο κάθε προμηθευτή. Το **Κεντρικό Σύστημα Κ.Α.Ρ.Ε.Ε.Α** αποθηκεύει όλους τους ορισμούς ροών εργασίας και περιπτώσεων ροών εργασίας, που διατυπώνονται από τους εμπειρογνώμονες. Το **Τοπικό Σύστημα Κ.Α.Ρ.Ε.Ε.Α** είναι υπεύθυνο για την αποθήκευση των ορισμών ροών εργασίας και περιπτώσεων ροών εργασίας, που διατυπώνονται από τον εμπειρογνώμονα – προμηθευτή στο τοπικό του μηχάνημα. Η ύπαρξη του **Τοπικού Συστήματος Κ.Α.Ρ.Ε.Ε.Α** βασίζεται στο γεγονός, ότι το σύστημα **ARION** μπορεί να εγκατασταθεί και να λειτουργήσει τοπικά στο σύστημα ενός προμηθευτή επιστημονικών αντικειμένων. Συνεπώς, το **Υποσύστημα ARION** του κάθε προμηθευτή λειτουργεί εσωτερικά, απαντώντας σε τοπικές επερωτήσεις και παράγοντας εσωτερικά σύνολα δεδομένων. Επίσης, επικοινωνεί με το κεντρικό σύστημα του **ARION** μόνο όταν άλλα επιστημονικά σύνολα δεδομένων (ξένα σε σχέση με αυτά που πραγματεύεται ο προμηθευτής) πρέπει να ευρεθούν ή να παραχθούν.

Το Σύστημα **Κ.Α.Ρ.Ε.Ε.Α** αποτελείται από τα εξής συστατικά μέρη : **Κειμενογράφο Ροών Εργασίας (Workflow Editor)**, **Αποθήκη Ροών Εργασίας (Workflow Storage)**, **Υπηρετητή Αποθήκης Ροών Εργασίας (Workflow Storage Server)**, **Στατιστική Βάση Δεδομένων (Statistical Database)**, **Υπηρετητή Στατιστικής Βάσης Δεδομένων (Statistical Database Server)**, **Υπηρετητή Ειδοποίησης (Notification Server)** και το **Εργαλείο Ανανέωσης (Update Tool)**. Το σχήμα 17 εικονογραφεί την διαρρύθμιση αυτών των συστατικών μερών, τις αλληλοσυνδέσεις, που έχουν μεταξύ τους, και τις σχέσεις, που έχουν αυτά με τα άλλα υποσυστήματα του συστήματος **ARION**. Συγκεκριμένα, κάθε κουτί του σχήματος υποδηλώνει ένα συστατικό μέρος του συστήματος μας (αν έχει χρώμα κίτρινο) ή κάποιο άλλο υποσύστημα. Τα βέλη διπλής κατεύθυνσης υποδηλώνουν ότι η σύνδεση μεταξύ δύο μερών ή υποσυστημάτων είναι αμφίδρομη (παρέχονται υπηρεσίες και από τις δύο μεριές), ενώ τα βέλη μονής κατεύθυνσης υποδηλώνουν ότι το ένα συστατικό μέρος ή υποσύστημα (από το οποίο αρχίζει το βέλος σύνδεσης) προσδοκά τις υπηρεσίες του άλλου.



Σχήμα 17 - Η κεντρική αρχιτεκτονική του συστήματος Κ.Α.Ρ.Ε.Ε.Α

Ο **Κειμενογράφος Ροών Εργασίας (Workflow Editor)** θα είναι ένας off-line ή ένας βασιζόμενος στο διαδίκτυο γραφικός κειμενογράφος. Θα είναι off-line εργαλείο, που προσφέρεται ως μέρος του **Τοπικού Συστήματος Κ.Α.Ρ.Ε.Ε.Α**. Ενώ θα είναι βασιζόμενος στο δίκτυο με τη μορφή ενός Java Applet για τη χρήση του από κάθε χρήστη του συστήματος **ARION**. Θα χρησιμοποιείται από εξειδικευμένους επιστήμονες / χρήστες για τον προσδιορισμό ροών εργασίας ή περιπτώσεων ροών εργασίας. Ο χρήστης του συστήματος **ARION** θα είναι σε θέση να ορίσει ένα νέο προσδιορισμό ροών εργασίας ή περιπτώσεων ροών εργασίας ή να τροποποιήσει ένα ήδη υπάρχοντα προσδιορισμό. Επιπροσθέτως, θα είναι σε θέση να παρέχει επιπλέον πληροφορία για το πού θα αποθηκευθεί ο προσδιορισμός (στην τοπική **Αποθήκη Ροών Εργασίας** ή σε μια απομακρυσμένη τοποθεσία). Όπου και να είναι αυτή η τοποθεσία αποθήκευσης, ο κάθε προσδιορισμός θα αποθηκεύεται και στο **Κεντρικό Σύστημα Κ.Α.Ρ.Ε.Ε.Α** (κεντρική **Αποθήκη Ροών Εργασίας**). Εκτός από όλα αυτά, ο χρήστης θα μπορεί να ξεκινήσει τη διαδικασία εκτέλεσης μιας περίπτωσης ροής εργασίας (να κάνει τεστ εκτέλεσης). Ο **Κειμενογράφος Ροών Εργασίας** θα επικοινωνεί με τον **Υπηρετητή Αποθήκης Ροών Εργασίας**, για να μεταφέρει το προσδιορισμό ροής εργασίας ή περίπτωσης ροής εργασίας στην **Αποθήκη Ροών Εργασίας** ή για να 'φορτώσει' ένα αντίστοιχο ορισμό. Επίσης, θα επικοινωνεί με τη **Μηχανή Αναζήτησης**, για να καθοδηγεί διαισθητικά τον εξειδικευμένο χρήστη να επιλέξει τις σωστές έννοιες / κλάσεις των συνόλων δεδομένων εισόδου ή εξόδου των έργων του προσδιορισμού μιας ροής εργασίας. Επιπροσθέτως, ο **Κειμενογράφος Ροών Εργασίας** θα είναι συνδεδεμένος με το **Σύστημα Ταυτοποίησης και Εξουσιοδότησης Χρήστη** με το σκοπό να ταυτοποιεί τους εξειδικευμένους χρήστες και να εξουσιοδοτεί τη χρήση των ροών εργασίας, των προγραμμάτων και των συνόλων δεδομένων.

Το κομμάτι του **Κειμενογράφου Ροών Εργασίας**, που ασχολείται αποκλειστικά με τον ορισμό περιπτώσεων ροών εργασίας και την προώθησή τους στο **Σύστημα Εκτέλεσης Ροών Εργασίας**, ονομάζεται **Κειμενογράφος Περιπτώσεων Ροών Εργασίας (Workflow Instance Editor)**. Αυτό το κομμάτι θα λειτουργεί και αυτόνομα στο διαδίκτυο με τη μορφή Java Applet. Θα ανακινείται από τις **Γραφικές Διεπαφές Χρήσης**, κάθε φορά που γίνεται αναζήτηση για κάποιο είδος συνόλων δεδομένων και θα επερωτά τον **Υπηρετητή της Αποθήκης Ροών Εργασίας** αν υπάρχουν ορισμοί ροών εργασίας, οι οποίες παράγουν αυτού του είδους συνόλων δεδομένων. Έπειτα, ο **Κειμενογράφος Περιπτώσεων Ροών Εργασίας** παρουσιάζει γραφικά στο χρήστη την απάντηση και τον καθοδηγεί στο να επιλέξει τον ορισμό ροής εργασίας, που τον ενδιαφέρει. Επακολούθως, τον καθοδηγεί διαισθητικά στο να δημιουργήσει ένα ορισμό περίπτωσης της παραπάνω ροής εργασίας και να τον εκτελέσει. Ο εν λόγω κειμενογράφος συνεργάζεται με όλα εκείνα τα συστατικά μέρη του συστήματος **Κ.Α.Ρ.Ε.Ε.Α** και τα υποσυστήματα **ARION** με τα οποία συνεργάζεται ο **Κειμενογράφος Ροών Εργασίας**.

Η **Αποθήκη Ροών Εργασίας** θα προσομοιώνεται είτε από το σύστημα αρχείων ενός λειτουργικού συστήματος (αφού κάθε προσδιορισμός αποθηκεύεται σε ένα αρχείο λειτουργικού συστήματος) είτε από μια XML βάση δεδομένων (το κάθε αρχείο αποθήκευσης προσδιορισμού θα είναι σε φόρμα XML). Στην Αποθήκη Ροών Εργασίας θα αποθηκεύονται και προσδιορισμοί περιπτώσεων ροών εργασίας.

Ο **Υπηρετητής Αποθήκης Ροών Εργασίας** είναι το πιο σημαντικό μέρος του **Συστήματος Κ.Α.Ρ.Ε.Ε.Α**. Προσφέρει τις λειτουργίες αποθήκευσης και επερώτησης είτε προσδιορισμών ροών εργασίας είτε προσδιορισμών περιπτώσεων ροών εργασίας. Η αποθήκευση ενός προσδιορισμού γίνεται άμεσα στην **Αποθήκη Ροών Εργασίας**, κατόπιν αιτήσεως είτε του **Κειμενογράφου Ροών Εργασίας** είτε των **Γραφικών Διεπαφών Χρήσης**. Ένας τοπικός **Υπηρετητής Αποθήκης Ροών Εργασίας** οφείλει όχι μόνο να αποθηκεύσει ένα προσδιορισμό στην τοπική του **Αποθήκη Ροών Εργασίας**, αλλά και να στείλει αυτόν τον προσδιορισμό προς αποθήκευση σε ένα κεντρικό **Υπηρετητή Αποθήκευσης Ροών Εργασίας**. Η λειτουργία της επερώτησης προσδιορισμού ροής εργασίας, η οποία ροή εργασίας παράγει σύνολο δεδομένων συγκεκριμένου τύπου, επιτελείται σε ένα κεντρικό **Υπηρετητή** με την κατανομή-αποστολή της επερώτησης σε ένα τοπικό **Υπηρετητή** κάθε τοπικού συστήματος **Κ.Α.Ρ.Ε.Ε.Α** που ανήκει σε ένα προμηθευτή επιστημονικών αντικειμένων και με την συλλογή των τοπικών αποτελεσμάτων που στέλνονται πίσω. Στις περιπτώσεις που κανένας τοπικός **Υπηρετητής** ενός τοπικού συστήματος **Κ.Α.Ρ.Ε.Ε.Α** ενός προμηθευτή δεν είναι διαθέσιμος ή που κάποιος προμηθευτής δεν έχει εγκαταστήσει ένα τοπικό σύστημα **Κ.Α.Ρ.Ε.Ε.Α**, τότε η αναζήτηση για προσδιορισμούς ροών εργασίας αυτών των προμηθευτών θα γίνεται κεντρικά και τα αποτελέσματα θα ενώνονται με αυτά που στέλνονται πίσω από τους τοπικούς **Υπηρετητές** που ρωτήθηκαν. Επομένως, η λειτουργία της επερώτησης / αναζήτησης προσδιορισμών ροών εργασίας επιτελείται, εν μέρει, με ένα κατανομημένο τρόπο. Στην περάτωση αυτής της λειτουργίας συμβάλει αποφασιστικά η ενημέρωση, που έχει ο κάθε κεντρικός **Υπηρετητής Αποθήκης Ροών Εργασίας** από τον **Υπηρετητή Ειδοποίησης**, η οποία αναλύεται παρακάτω. Η λειτουργία της επερώτησης προσδιορισμών ροών εργασίας επιτελείται από ένα τοπικό **Υπηρετητή**, αρχικά, με την αναζήτηση προσδιορισμών στην τοπική του **Αποθήκη Ροών Εργασίας**. Αν το αποτέλεσμα είναι αρνητικό, τότε η επερώτηση προωθείται σε ένα κεντρικό **υπηρετητή**, ο οποίος φυσικά και δεν θα ξαναρωτήσει τον τοπικό **υπηρετητή**, όταν θα κατανείμει την επερώτηση. Σε αυτό το σημείο θα πρέπει να τονιστεί, προς αποφυγή παρεξηγήσεων, ότι όλοι οι κεντρικοί **υπηρετητές** είναι υπεύθυνοι για την μια και μοναδική κεντρική **Αποθήκη**

Ροών Εργασίας και ότι όλοι οι τοπικοί υπηρετητές, ενός τοπικού συστήματος, είναι υπεύθυνοι για τη μια και μοναδική τοπική **Αποθήκη Ροών Εργασίας**.

Το **Σύστημα Εκτέλεσης Ροών Εργασίας** επικοινωνεί με τον **Υπηρετητή Στατιστικής Βάσης Δεδομένων**, για να μεταβιβάσει στατιστική πληροφορία που αφορά την εκτέλεση μιας περίπτωσης ροής εργασίας, όπως τη διάρκεια της εκτέλεσης, τον αριθμό των πόρων που δεσμεύθηκαν, τον αριθμό των κόμβων δικτύου που χρησιμοποιήθηκαν, τον αριθμό των λαθών εκτέλεσης, τις προειδοποιήσεις κ.τ.λ.. Έπειτα, ο **Υπηρετητής της Στατιστικής Βάσης Δεδομένων** αποθηκεύει αυτή την πληροφορία σε μια κατάλληλη σχεσιακή φόρμα μέσα στη **Στατιστική Βάση Δεδομένων**, η οποία και είναι μια σχεσιακή βάση δεδομένων (όπως είναι η Oracle). Ο **Υπηρετητής Στατιστικής Βάσης Δεδομένων** είναι, επίσης, υπεύθυνος για την άντληση στατιστικής πληροφορίας τόσο για μια ροή εργασίας όσο και για τις περιπτώσεις αυτής της ροής εργασίας από τη **Στατιστική Βάση Δεδομένων**. Η λειτουργία της άντλησης στατιστικής πληροφορίας θα μπορούσε να μοιάζει, διαδικαστικά, με τη λειτουργία της επερώτησης προσδιορισμών ροών εργασίας. Κάτι τέτοιο θα ήταν, όμως, παρακινδυνευμένο, λαμβάνοντας υπόψιν ότι πολλοί είναι οι προμηθευτές, που δεν θα ήθελαν να εγκαταστήσουν το στατιστικό κομμάτι (**Υπηρετητή** και τη **Στατιστική Βάση Δεδομένων**) στο τοπικό τους σύστημα **Κ.Α.Ρ.Ε.Ε.Α** και επειδή υπάρχουν και κάποιοι άλλοι, που δεν θα ήθελαν καν να εγκαταστήσουν ένα τοπικό σύστημα **Κ.Α.Ρ.Ε.Ε.Α**. Συνεπώς, η λειτουργία της άντλησης στατιστικής πληροφορίας επιτελείται αυστηρώς κεντρικά και όχι καταναμημένα.

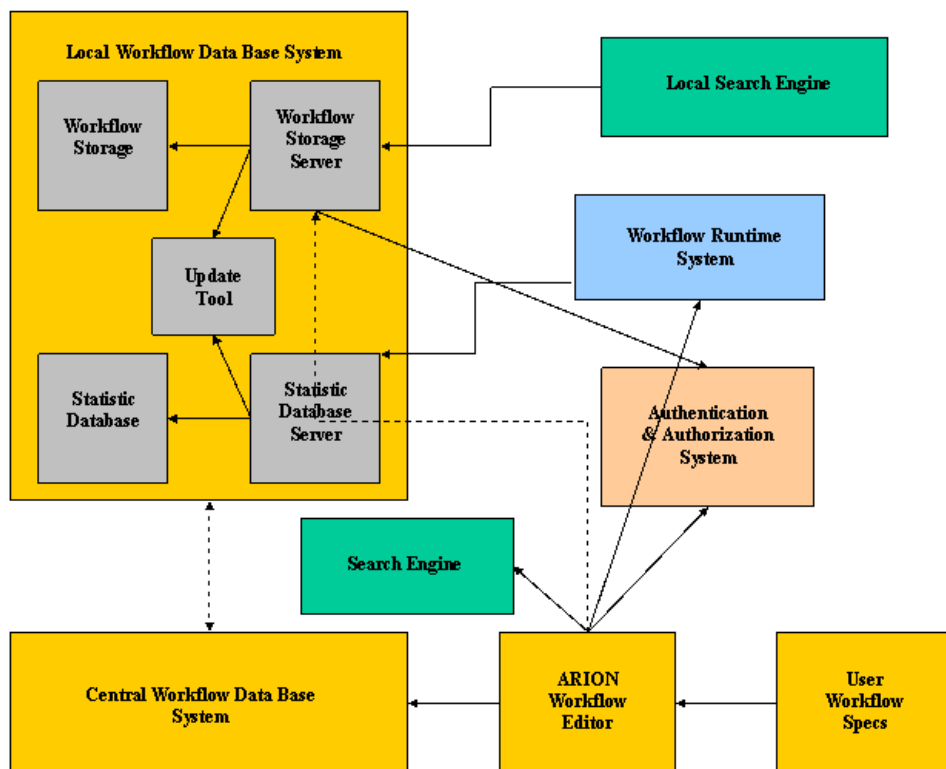
Ο **Υπηρετητής Ειδοποίησης (Notification Server)** είναι υπεύθυνος για την σωστή εικόνα, που θα πρέπει να έχει ένας **Υπηρετητής της Αποθήκης Ροών Εργασίας** σχετικά με το ποιοι άλλοι Υπηρετητές Αποθηκών Ροών Εργασίας υπάρχουν στο σύστημα. Ο τρόπος, που γίνεται αυτό, είναι: ο κάθε Υπηρετητής Αποθήκης Ροών Εργασίας αρχικά εγγράφεται στον Υπηρετητή Ειδοποίησης με την αποστολή αντίστοιχου μηνύματος. Έπειτα, ανά τακτά χρονικά συστήματα στέλνει ένα μήνυμα ότι είναι 'ζωντανός' ('I am alive' message). Με αυτό τον τρόπο, ο Υπηρετητής Ειδοποίησης γνωρίζει ανά πάσα στιγμή ποιος Υπηρετητής Αποθήκης Ροών Εργασίας είναι 'ζωντανός' και ποιος έχει 'πεθάνει'. Όποτε συμβεί κάποια αλλαγή, που αφορά ένα **Τοπικό Υπηρετητή Αποθήκης Ροών Εργασίας**, ενημερώνει τους **Κεντρικούς Υπηρετητές Αποθηκών Ροών Εργασίας** για αυτήν την αλλαγή. Στην περίπτωση, που συμβεί αλλαγή με ένα Κεντρικό Υπηρετητή Αποθήκης Ροών Εργασίας, ενημερώνει όλους τους Τοπικούς Υπηρετητές Αποθηκών Ροών Εργασίας για αυτή την αλλαγή. Πρέπει εδώ να σημειωθεί ότι: όταν αναφέρονται οι Τοπικοί Υπηρετητές Αποθηκών Ροών Εργασίας εννοούνται τα προγράμματα, που υλοποιούν το μοντέλο του **Κεντρικού Υπηρετητή Αποθήκης Ροών Εργασίας**. Το γεγονός ότι τα προγράμματα αυτά είναι πολλά σε πλήθος οφείλεται στο ότι πρέπει να επιτευχθεί κατανομή του φόρτου παροχής υπηρεσιών για το κάθε πρόγραμμα αυτού του είδους. Η ίδια σημείωση ισχύει και για τους **Τοπικούς Υπηρετητές Αποθηκών Ροών Εργασίας**. Επίσης, πρέπει να σημειωθεί ότι η λειτουργία της ειδοποίησης μπορεί, κάλλιστα, να προσφερθεί και για τους **Υπηρετητές Στατιστικών Βάσεων Δεδομένων**.

Η λειτουργία του **Υπηρετητή Ειδοποίησης** είναι κρίσιμη για το σύστημά μας για τρεις λόγους. Πρώτον, βοηθάει / ενισχύει την επικοινωνία μεταξύ των τοπικών και κεντρικών **Υπηρετητών Αποθήκης Ροών Εργασίας**. Έτσι κάθε τοπικός Υπηρετητής Αποθήκης Ροών Εργασίας μπορεί να ενημερωθεί για το ποιοι αντίστοιχοι κεντρικοί Υπηρετητές είναι 'ζωντανοί', κάθε φορά που χάνει την επικοινωνία του με κάποιον κεντρικό υπηρετητή, και να διαλέξει έναν από αυτούς.

Αν δεν υπάρχει κανένας ‘ζωντανός’, τότε θα ρωτάει ανά τακτά χρονικά διαστήματα τον Υπηρετητή Ειδοποίησης αν κάποιος κεντρικός υπηρετητής επαναλειτούργησε ή μόλις μπήκε σε λειτουργία μέχρι να πάρει μια θετική απάντηση και να αποκαταστήσει την ‘κεντρική’ επικοινωνία που είχε. Το ίδιο ισχύει και για τους κεντρικούς Υπηρετητές με μια μικρή διαφορά. Ο κάθε κεντρικός Υπηρετητής θα περιμένει έναν από τους τοπικούς υπηρετητές του τοπικού συστήματος ενός προμηθευτή επιστημονικών αντικειμένων, να επανατεθεί σε λειτουργία, προκειμένου να αποκαταστήσει την επικοινωνία που είχε με το τοπικό σύστημα αυτού του προμηθευτή. Η τελευταία λεπτομέρεια προσδίδει τη βάση, για να εξηγήσουμε και το δεύτερο λόγο της κρισιμότητας της λειτουργίας του Υπηρετητή Ειδοποίησης. Ο κάθε κεντρικός Υπηρετητής Αποθήκης Ροών Εργασίας, όταν του έρθει μια ερώτηση από κάποιον χρήστη, θα καταναίμει την ερώτηση αυτή σε κάθε τοπικό υπηρετητή με τον οποίο έχει επικοινωνία σε κάθε τοπικό σύστημα ενός προμηθευτή. Άρα, για εκείνα τα τοπικά συστήματα προμηθευτών στα οποία έχει χαθεί η επικοινωνία και για εκείνους τους προμηθευτές, που δεν έχουν εγκαταστήσει το τοπικό σύστημα **Κ.Α.Ρ.Ε.Ε.Α**, η ερώτηση θα εξεταστεί από το κεντρικό σύστημα. Συνεπώς, η κάθε ερώτηση θα είναι μερικώς κατανεμημένη ανάλογα με την εικόνα, που παρουσιάζει το σύστημά μας. Επομένως, η λειτουργία του Υπηρετητή Ειδοποίησης συνεισφέρει στη σωστή κατανομή των ερωτήσεων, που διατυπώνονται στους κεντρικούς Υπηρετητές Αποθηκών Ροών Εργασίας. Ο τρίτος λόγος της κρισιμότητας της λειτουργίας αυτού του συστατικού μέρους είναι η συνεισφορά του στη κατανομή φορτίου. Η κατανομή φορτίου επιτυγχάνεται με τη προσθήκη ενός ονόματος, που αντιστοιχεί σε πολλές διευθύνσεις IP εκείνων των προγραμμάτων, που υλοποιούν το μοντέλο του Υπηρετητή Αποθήκης Ροών Εργασίας στον Υπηρετητή DNS, όπως θα αναφερθεί και παρακάτω. Κάθε φορά, που θα χάνεται ένα πρόγραμμα ή θα προστίθεται / επανέρχεται, ο Υπηρετητής Ειδοποίησης θα ειδοποιεί τον Υπηρετητή DNS να αφαιρέσει ή να προσθέσει μια αντιστοίχιση ονόματος με IP αντίστοιχα. Λόγω της κρισιμότητας της ύπαρξης του Υπηρετητή Ειδοποίησης και λόγω του ότι είναι ένα κεντροποιημένο συστατικό μέρος, το οποίο ενδέχεται να έχει βλάβες ή να απομονωθεί λόγω διαμερισμού του δικτύου, προβλέπεται η ύπαρξη ενός **Βοηθητικού Υπηρετητή Ειδοποίησης**. Αυτός ο βοηθητικός υπηρετητής θα ενημερώνεται από τον κύριο Υπηρετητή Ειδοποίησης για όλες τις αλλαγές στην εικόνα του συστήματος. Μόλις ανιχνεύσει ότι ο κεντρικός Υπηρετητής Ειδοποίησης έχει υποστεί βλάβη, αναλαμβάνει αυτός τα ηνία, γίνεται κεντρικός Υπηρετητής Ειδοποίησης και ενημερώνει όλους τους ενδιαφερόμενους ειδοποίησης (Υπηρετητές Αποθηκών Ροών Εργασίας και Υπηρετητές DNS) για αυτήν την αλλαγή. Όταν ο ‘νεκρός’ κύριος Υπηρετητής επανέλθει, τότε θα αναλάβει τα καθήκοντα του Βοηθητικού Υπηρετητή Ειδοποίησης.

Το **Εργαλείο Ανανέωσης** είναι ένα συστατικό μέρος ενός **Τοπικού Συστήματος Κ.Α.Ρ.Ε.Ε.Α**. Είναι υπεύθυνο για την καταγραφή των αλλαγών, που επιτελούνται στη **Τοπική Αποθήκη Ροών Εργασίας**, όταν η επικοινωνία με κάθε **Κεντρικό Υπηρετητή Αποθήκης Ροών Εργασίας** χαθεί είτε λόγω δυσλειτουργίας των κεντρικών υπηρετητών είτε λόγω διαμερισμού του δικτύου. Μόλις η επικοινωνία με κάποιον **Κεντρικό Υπηρετητή Αποθήκης Ροών Εργασίας** αποκατασταθεί, το εργαλείο αυτό προωθεί στη **Κεντρική Αποθήκη Ροών Εργασίας** τις αλλαγές, που επιτελέστηκαν στο ‘νεκρό’ διάστημα της επικοινωνίας, μέσω του κεντρικού υπηρετητή που επανήλθε σε λειτουργία. Επειδή το **Εργαλείο Ανανέωσης** είναι ένα κρίσιμο συστατικό μέρος ενός τοπικού συστήματος **Κ.Α.Ρ.Ε.Ε.Α** και επειδή πρέπει να υλοποιείται από μόνο ένα πρόγραμμα (είναι κεντροποιημένο), θα πρέπει να

ακολουθηθεί η ίδια τακτική όσον αφορά το back up, όπως αποτελεί η αντίστοιχη του Υπηρετητή Ειδοποίησης.

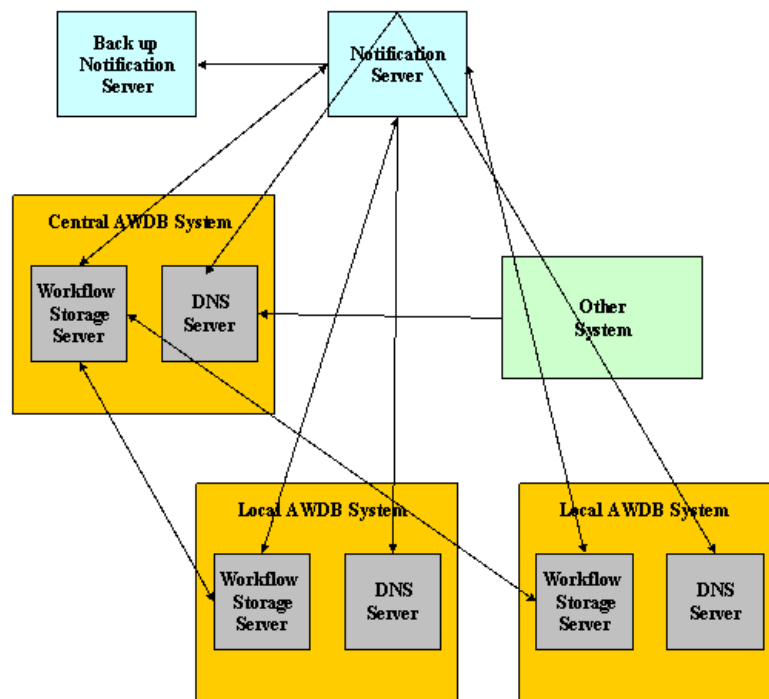


Σχήμα 18 - Η αρχιτεκτονική ενός τοπικού συστήματος Κ.Α.Ρ.Ε.Ε.Α

Υπάρχουν προφανείς διαφορές στην τοπολογία και στη λειτουργικότητα των παραπάνω συστατικών μερών του **Τοπικού Συστήματος Κ.Α.Ρ.Ε.Ε.Α**, όπως προκύπτει από το σχήμα 18. Λόγω των διαφορετικών δυνατών ρυθμίσεων παράταξης του συστήματος **ARION**, ο **Τοπικός Υπηρετητής της Αποθήκης Ροών Εργασίας** είναι συνδεδεμένος με το υποσύστημα της **Τοπικής Μηχανής Αναζήτησης**, αν αυτή υπάρχει, ή με το υποσύστημα της **Κεντρικής Μηχανής Αναζήτησης**. Η επικοινωνία του **Τοπικού Υπηρετητή της Αποθήκης Ροών Εργασίας** τόσο με το **Σύστημα Εκτέλεσης Ροών Εργασίας** όσο και με το **Σύστημα Ταυτοποίησης και Εξουσιοδότησης Χρήστη** παραμένει η ίδια, αφού αυτά τα δύο υποσυστήματα βρίσκονται μόνο στο κεντρικό σύστημα **ARION**. Η **Τοπική Στατιστική Βάση Δεδομένων** και ο αντίστοιχος υπηρετητής της είναι προαιρετικά συστατικά μέρη ενός **Τοπικού Συστήματος Κ.Α.Ρ.Ε.Ε.Α**, αφού ένας προμηθευτής του **ARION** μπορεί να μην ενδιαφέρεται να αποθηκεύσει στατιστική πληροφορία σε τοπικά μηχανήματα.

Οι ροές εργασίας, που είναι δια-επιχειρησιακές (inter-organizational), περιλαμβάνουν σύνολα δεδομένων και προγράμματα, που βρίσκονται σε διαφορετικούς κόμβους (συστήματα) του **ARION**. Αυτού του είδους οι ροές εργασίας αποθηκεύονται μόνο στο **Κεντρικό Σύστημα Κ.Α.Ρ.Ε.Ε.Α**. Επομένως, η άντληση πληροφορίας ή οι επερωτήσεις, που αφορούν ροές εργασίας που είναι δια-επιχειρησιακές, θα γίνονται αυστηρώς από το κεντρικό σύστημα του **Κ.Α.Ρ.Ε.Ε.Α**. Τα **Τοπικά Συστήματα Κ.Α.Ρ.Ε.Ε.Α** μπορούν μόνο να αποθηκεύσουν ροές εργασίας, που αποτελούνται από προγράμματα (και σύνολα δεδομένων) που κατά

αποκλειστικότητα εκτελούνται τοπικά. Αυτά τα είδη ροών εργασίας αποθηκεύονται στο **Κεντρικό Σύστημα Κ.Α.Ρ.Ε.Ε.Α**.



Σχήμα 19 - Αρχιτεκτονική Ειδοποιητή και DNS Server

5.3.2.1. Χαρακτηριστικά της αρχιτεκτονικής του συστήματος Κ.Α.Ρ.Ε.Ε.Α

Η σωστή και διαρθρωτική σχεδίαση της αρχιτεκτονικής του συστήματος Κ.Α.Ρ.Ε.Ε.Α προσέδωσε σε αυτήν ενδιαφέρον και ουσιαστικά χαρακτηριστικά. Παραθέτουμε, ακολούθως, μια λίστα των χαρακτηριστικών μαζί με μια αιτιολόγηση του πώς προσδίδονται στην αρχιτεκτονική του συστήματος μας :

- *Openness*: Η αρχιτεκτονική του συστήματος μας είναι ανοικτή (open), διότι δεν διευκρινίζει: σε ποιο λειτουργικό σύστημα θα εκτελούνται τα διάφορα συστατικά της μέρη, σε ποια μορφής δίκτυο θα είναι συνδεδεμένα αναμεταξύ τους, ποια θα είναι η διάταξή τους στο δίκτυο, καθώς και σε ποιους υπολογιστές θα εκτελούνται. Επίσης, είναι ανοικτή ως προς την ύπαρξη νέων λειτουργιών, που θα αφορούν τη διαχείριση των αντικειμένων του συστήματος, και οι οποίες διαχωρίζονται στους προσδιορισμούς ροών εργασίας και περιπτώσεων ροών εργασίας και στη στατιστική πληροφορία για ροές εργασίας και για περιπτώσεις ροών εργασίας. Όπως είναι αντιληπτό, το σύστημα Κ.Α.Ρ.Ε.Ε.Α προσφέρει δύο βασικές λειτουργίες για κάθε αντικείμενο. Επιπλέον, με τη βοήθεια αυτών των λειτουργιών, χτίζει-δομεί νέες λειτουργίες, όπως γίνεται με την περίπτωση του **Κειμενογράφου Ροών Εργασίας**. Το σύστημά μας είναι ανοικτό αφενός, στη δημιουργία νέων βασικών λειτουργιών διαχείρισης των βασικών της αντικειμένων (αν υπάρχουν), και αφετέρου, στη δημιουργία νέων λειτουργιών,

που θα χτιστούν πάνω από τις βασικές. Οι νέες βασικές λειτουργίες-υπηρεσίες, προφανώς, θα παρέχονται από εκείνα τα συστατικά μέρη, που έχουμε ορίσει ως υπεύθυνα για τα αντικείμενα που διαχειρίζονται αυτές οι λειτουργίες.

- *Κατανομή (distribution)*: Η αρχιτεκτονική του συστήματός μας προσφέρει την λογική και φυσική κατανομή των διαδικασιών-λειτουργιών. Με την ύπαρξη ενός κεντρικού και πολλών τοπικών συστημάτων **Κ.Α.Ρ.Ε.Ε.Α** και με το διαχωρισμό των διαφορετικών λειτουργιών (ως προς το ποια αντικείμενα διαχειρίζονται), μεταξύ των συστατικών μερών, επιτυγχάνεται η λογική κατανομή. Επειδή το κάθε συστατικό μέρος ενός συστήματος **Κ.Α.Ρ.Ε.Ε.Α** μπορεί να βρίσκεται ταυτόχρονα σε διαφορετικούς υπολογιστές, θα επιτελείται και φυσική κατανομή των λειτουργιών.
- *Κλιμακωσιμότητα (scalability)*: Η κλιμακωσιμότητα αφορά αν ένα σύστημα μπορεί να ανταποκριθεί στην αύξηση του πλήθους των αντικειμένων, τα οποία χειρίζεται μέσα από τις λειτουργίες του. Τα αντικείμενα του συστήματος **Κ.Α.Ρ.Ε.Ε.Α** είναι τα εξής τέσσερα: προσδιορισμοί ροών εργασίας, προσδιορισμοί περιπτώσεων ροών εργασίας, στατιστική πληροφορία για ροές εργασίας και στατιστική πληροφορία για περιπτώσεις ροών εργασίας. Οι λειτουργίες για κάθε αντικείμενο είναι δύο: αποθήκευση και αναζήτηση. Οι ροές εργασίας, που γενικά θα ορίσει ένας επιστημονικός οργανισμός, δεν υπερβαίνουν, συνήθως, το πλήθος των 10. Αν θεωρήσουμε ότι ένας προσδιορισμός μιας τυπικής ροής εργασίας καταλαμβάνει χώρο γύρω στα 6 Kbytes, τότε συνολικά θα χρειαστεί χώρος περίπου 60 Mbytes για την αποθήκευση προσδιορισμών ροών εργασίας για 1000 οργανισμούς. Όμως, ένας τυπικός δίσκος ενός υπολογιστή κυμαίνεται σήμερα από 40 σε 100 Gigabytes. Συνεπώς, όλοι οι προσδιορισμοί δεν θα έχουν πρόβλημα να αποθηκευθούν στην κεντρική **Αποθήκη Ροών Εργασίας**. Κάθε προσδιορισμός περίπτωσης ροής εργασίας περιλαμβάνει, κατά μέσο όρο, χώρο 8 Kbytes συν ένα ακόμη ειδικό αρχείο του 1 Kbytes, δηλαδή συνολικά 9 Kbytes. Αν θεωρήσουμε ότι ένας επιστημονικός οργανισμός εκτελεί 500 περιπτώσεις μιας ροής εργασίας το χρόνο, τότε συνολικά ο χώρος, που θα χρειαστεί για την αποθήκευση όλων των προσδιορισμών περιπτώσεων ροών εργασίας, θα είναι γύρω στα 45 Gigabytes. Επομένως, και σε αυτήν την περίπτωση η κεντρική **Αποθήκη Ροών Εργασίας** μπορεί να 'αντέξει' για ένα χρόνο. Προφανώς, στο τέλος κάθε χρόνου θα πρέπει να εφαρμοστεί μια πολιτική καθαρισμού του δίσκου, διότι αφενός οι μισές περιπτώσεις ροών εργασίας που θα τρέξουν δεν θα έχουν επιθυμητά αποτελέσματα και άρα θα είναι άχρηστες και αφετέρου αυτό που ενδιαφέρει στο τέλος είναι το αποτέλεσμα και όχι ο ακριβής τρόπος που παράχθηκε αυτό. Οι τοπικές **Αποθήκες Ροών Εργασίας** θα χρειάζονται χώρο για 60 Kbytes προσδιορισμών ροών εργασίας (γενικά) και 45 Mbytes προσδιορισμών περιπτώσεων ροών εργασίας (κάθε χρόνο). Επομένως, θα γέμιζαν σε 1000 χρόνια από τώρα. Η στατιστική πληροφορία θα αποθηκεύεται σε μορφή εγγραφών σε μια σχεσιακή βάση δεδομένων. Οπότε κατά μέσο όρο θα χρειαστούν 10000 εγγραφές με στατιστική πληροφορία για ροές εργασίας (γενικά) και 5000000 εγγραφές με στατιστική πληροφορία για περιπτώσεις ροών εργασίας (το χρόνο) για την κεντρική **Στατιστική Βάση Δεδομένων**. Επομένως, και εδώ θα πρέπει να εφαρμοστεί μια πολιτική καθαρισμού πληροφορίας για περιπτώσεις ροών εργασίας, που θα εφαρμόζεται κάθε χρόνο. Οι τοπικές **Στατιστικές Βάσεις Δεδομένων**, αυτές θα χρειαστούν 10 εγγραφές για ροές εργασίας γενικά και 5000 εγγραφές για περιπτώσεις ροών εργασίας το χρόνο. Προφανώς, αν θέλουμε να έχουμε κατανομή της αποθήκευσης και αν θέλει κάθε οργανισμός να λειτουργεί τοπικά στους υπολογιστές του το σύστημα **ARION**,

τότε θα πρέπει να εκμεταλλευτούμε το χώρο, που έχουν στους υπολογιστές τους και να τους αναγκάσουμε να εγκαταστήσουν όλο το τοπικό σύστημα **Κ.Α.Ρ.Ε.Ε.Α**. Βέβαια, για να συμβεί αυτό πρέπει να είμαστε σίγουροι για την εκτίμησή μας, διότι είναι προφανείς οι δυσκολίες που θα προκύπτουν από/κατά τη συνεργασία ανταγωνιστικών οργανισμών, προκειμένου να μετάσχουν στο σύστημα **ARION**. Συνεπώς, ο αριθμός των 1000 οργανισμών μπορεί να μην είναι και τόσο ακριβής στην πραγματικότητα. Ωστόσο, εμείς οφείλαμε να εξετάσουμε τα όρια, που προσφέρει η αρχιτεκτονική του συστήματος μας. Τώρα, όσον αφορά τη λειτουργία της επερώτησης και για τα τέσσερα είδη αντικειμένων του συστήματος **ARION**, η εν λόγω λειτουργία πραγματοποιείται εν μέρει κατανομημένα. Με τις παραπάνω εκτιμήσεις και με αυτήν την κατανομή, η λειτουργία της επερώτησης δεν θα παρουσιάζει κανένα πρόβλημα. Ειδικότερα, αν όλοι οι οργανισμοί εγκαταστήσουν το τοπικό σύστημα **Κ.Α.Ρ.Ε.Ε.Α**, τότε όχι μόνο θα υπάρχει χώρος για περισσότερες ροές εργασίας και περιπτώσεις αυτών, αλλά και το πλήθος των εμπλεκόμενων οργανισμών δύναται να αυξηθεί. Επιπροσθέτως, η κεντρική βάση δεδομένων οι κεντρικές αποθήκες δεν έχουν πλέον νόημα ύπαρξης.

- **Ταυτοχρονισμός και Κατανομή Φόρτου (Concurrency and Load-balancing):** Αν ο αριθμός των χρηστών αυξηθεί σημαντικά, τότε θα πρέπει να διπλασιάσουμε ή να τριπλασιάσουμε τα προγράμματα, που υλοποιούν τα συστατικά μέρη των κεντρικών ή τοπικών **Υπηρετητών Αποθηκών Ροών Εργασίας** και **Υπηρετητών Στατιστικών Βάσεων Δεδομένων**. Θα πρέπει να κατανοήσουμε αυτά τα προγράμματα γεωγραφικά σε διάφορους υπολογιστές. Επιπροσθέτως, οι Υπηρετητές DNS, τόσο του κεντρικού συστήματος **ARION** όσο και των τοπικών, θα πρέπει να τροποποιηθούν, έτσι ώστε να διευκολύνουν την αντιστοίχιση ενός ονόματος υπολογιστή με πολλές IP διευθύνσεις - ή καλύτερα με πολλά σύνολα (IP και port), αφού αυτό το σύνολο αντιπροσωπεύει τη διεύθυνση ενός προγράμματος. Με αυτό τον τρόπο, θα προσφέρουν κατανομή φόρτου κάθε φορά που δέχονται μια αίτηση DNS για ένα όνομα υπολογιστή, αντιστοιχίζοντας αυτό το όνομα με διαφορετικό IP κάθε φορά. Κατά συνέπεια, αν το όνομα του **Υπηρετητή Αποθήκης Ροών Εργασίας** ή του **Υπηρετητή Στατιστικής Βάσης Δεδομένων** αντιστοιχίζεται με διαφορετική διεύθυνση IP κάθε φορά (και άρα σε διαφορετικό πρόγραμμα κάθε φορά), τότε αφενός επιτυγχάνεται κατανομή φόρτου και αφετέρου περισσότεροι χρήστες υπηρετούνται την ίδια χρονική στιγμή. Μάλιστα, η συγκεκριμένη τροποποίηση ενός **Υπηρετητή DNS** μπορεί να επιτευχθεί πιο σωστά και επιτυχημένα, αν γίνει σε συνεργασία με τον **Υπηρετητή Ειδοποίησης**, διότι μόνο αυτός γνωρίζει ποια προγράμματα είναι 'ζωντανά' και ποια είναι 'νεκρά'. Η συνεργασία **Υπηρετητή DNS - Υπηρετητή Ειδοποίησης** έχει αναφερθεί σε προηγούμενη παράγραφο. Εποπτικά, ο τρόπος με τον οποίο θα γίνεται η επικοινωνία μεταξύ των υπηρετητών του συστήματος **Κ.Α.Ρ.Ε.Ε.Α** φαίνεται στο σχήμα 19.
- **Αξιοπιστία (reliability):** Η αξιοπιστία του συστήματος σχετίζεται με το εάν επιτελούνται σωστά οι λειτουργίες του και χωρίς λάθη. Υπάρχουν δύο είδη αξιοπιστίας. Αξιοπιστία στην επικοινωνία μεταξύ των διάφορων συστατικών μερών και υποσυστημάτων, ώστε να ζητηθεί η εκτέλεση λειτουργιών. Και αξιοπιστία στην τέλεση των λειτουργιών. Η αξιοπιστία στην επικοινωνία αφορά τον τρόπο επιτυχίας της σύνδεσης και στη μεταφορά των δεδομένων. Η αξιοπιστία του τρόπου σύνδεσης επιτυγχάνεται με τα όσα γίνονται για την κατανομή φόρτου, που περιγράφηκαν προηγουμένως. Η αξιοπιστία στη μεταφορά δεδομένων αφορά την ποιότητα του καναλιού επικοινωνίας και το αν

θα προκύψει κάποιο πρόβλημα στο δίκτυο. Η ποιότητα του καναλιού επικοινωνίας εξασφαλίζεται με τη χρησιμοποίηση αξιόπιστων καναλιών επικοινωνίας, που ακολουθούν συγκεκριμένα πρωτόκολλα επικοινωνίας, όπως το TCP [88]. Η πλειοψηφία των προβλημάτων του δικτύου λύνονται εν μέρη από το πρωτόκολλο TCP. Τα πιο σοβαρά προβλήματα οδηγούν σε απώλεια της επικοινωνίας, πράγμα που θα οδηγήσει σε επανέναρξη της προσπάθειας επικοινωνίας και, άρα, επανέναρξη της συνολικής λειτουργίας. Προβλήματα αυτού του είδους, δύσκολα, μπορούν να αποφευχθούν. Συνεπώς, η αξιοπιστία της επικοινωνίας αντιμετωπίζεται εν μέρη. Σε περίπτωση μη επιτυχημένης αντιμετώπισης, η διαδικασία της επικοινωνίας επανεκινείται. Η αξιοπιστία στην τέλεση των λειτουργιών αφορά καθαρά τον τρόπο με τον οποίο εκτελούνται οι λειτουργίες. Υπάρχουν δύο ειδών λειτουργίες στο σύστημα **K.A.P.E.E.A** που σχετίζονται με την αποθήκευση αντικειμένων και την επερώτηση για αυτά. Πριν προχωρήσουμε στην ανάλυση του τρόπου εκτέλεσης αυτών των λειτουργιών θα πρέπει να τονίσουμε ότι τυχόν λάθη στην εκτέλεση θα πρέπει να οδηγούν σε επανεκτέλεση της λειτουργίας. Αυτό πρακτικά σημαίνει ότι ο πελάτης, που ζητάει την εκτέλεση της λειτουργίας, δεν θα πρέπει να διακόπτει την επικοινωνία του με τον υπηρετητή, παρά μόνο εάν λάβει μήνυμα επιβεβαίωσης από τον υπηρετητή ότι η εκτέλεση της λειτουργίας ήταν επιτυχής. Άρα, θα πρέπει να προβλεφθεί ένα πρωτόκολλο επικοινωνίας πελάτη-υπηρετητή, όπως αυτό του Remote Procedure Call – RPC [89] που θα χτίζεται πάνω από το TCP. Τώρα θα αναλύσουμε τον τρόπο εκτέλεσης των λειτουργιών: Η λειτουργία της αποθήκευσης αφορά την αποθήκευση δεδομένων σε ένα αρχείο ή σε μια βάση δεδομένων και γίνεται σε ένα ή σε δύο βήματα, ανάλογα με το ποιος την επιτελεί. Αν είναι ένας υπηρετητής, που βρίσκεται στο κεντρικό σύστημα, τότε εκτελείται σε ένα βήμα. Αλλιώς, αν είναι υπηρετητής του τοπικού συστήματος **K.A.P.E.E.A** θα την εκτελέσει σε δύο βήματα: θα αποθηκεύσει τα δεδομένα τοπικά και ύστερα θα προσπαθήσει να τα αποθηκεύσει και κεντρικά, επικοινωνώντας με ένα κεντρικό υπηρετητή. Για την πρώτη περίπτωση τα πράγματα είναι απλά. Η λειτουργία θα επαναεκτελείται όσο υπάρχουν λάθη, μέχρι κάποιο αριθμό προσπαθειών, και έπειτα θα ενημερώνεται ο χρήστης (ή ο κάθε ενδιαφερόμενος) ότι η λειτουργία δεν μπόρεσε να εκτελεστεί. Για τη δεύτερη περίπτωση, η λειτουργία χωρίζεται σε δύο κομμάτια. Το πρώτο κομμάτι είναι το ίδιο με την πρώτη περίπτωση. Το δεύτερο κομμάτι έχει δύο πτυχές. Αν δεν μπορεί να εκτελεστεί λόγω προβλήματος επικοινωνίας, τότε θα ανατίθεται στον **Υπηρετητή Ανανέωσης**. Αν όμως δεν μπορεί ο κεντρικός υπηρετητής να την εκτελέσει, τότε θα πρέπει αφενός να αναστραφεί το αποτέλεσμα του πρώτου κομματιού και αφετέρου να ενημερωθεί ο χρήστης ότι η λειτουργία εκτελέστηκε ανεπιτυχώς. Όσον αφορά τη λειτουργία της επερώτησης, αναφέρθηκε ήδη ότι είναι εν μέρη κατανεμημένη. Οπότε, η επιτυχία της εξαρτάται και από το κεντρικοποιημένο κομμάτι και από τα κατανεμημένα. Αν το κεντρικοποιημένο αποτύχει, τότε εκτελείται κάποιες φορές ακόμη και μετά ενημερώνεται ο χρήστης για την αποτυχία. Αν κάποιο κατανεμημένο κομμάτι αποτύχει, τότε γίνεται κεντρικοποιημένο και εκτελείται όπως πριν. Συνοπτικά, λοιπόν, η αξιοπιστία του συστήματός μας λογαριάζει διάφορες περιπτώσεις και τις αντιμετωπίζει επιτυχημένα, εκτός από μερικές όπου η εγκατάλειψη της εκτέλεσης μιας λειτουργίας δεν μπορεί να αποφευχθεί. Τέλος, θα πρέπει να τονιστεί ότι το σύστημα **K.A.P.E.E.A** δεν μπορεί να αντιμετωπίσει περιπτώσεις, όπου ή το λειτουργικό σύστημα ή η βάση δεδομένων ή ο σκληρός δίσκος του κεντρικού συστήματος δυσλειτουργούν ή έχουν υποστεί σοβαρή βλάβη, διότι αντιμετωπίζονται μόνο από τους διαχειριστές του συστήματος.

- Ασφάλεια (Security)*: Η ασφάλεια του συστήματος **K.A.P.E.E.A** εξασφαλίζεται σε δύο επίπεδα. Το πρώτο επίπεδο αφορά την εσωτερική επικοινωνία μεταξύ των συστατικών μερών του συστήματος, αλλά και την επικοινωνία των συστατικών μερών του συστήματος με τα άλλα υποσυστήματα του **ARION**. Το πρώτο επίπεδο μπορεί να επιτευχθεί με την κωδικοποίηση, μέσω κλειδιών, των μηνυμάτων που στέλνονται από το κανάλι επικοινωνίας δύο οποιαδήποτε συμμετεχόντων μερών. Αυτό επιτυγχάνεται με τη χρησιμοποίηση ενός πρωτοκόλλου, όπως το SSL [90] που βρίσκεται πάνω από το TCP αλλά κάτω από πρωτόκολλα της μορφής RPC. Το δεύτερο επίπεδο σχετίζεται με το εάν κάποιος χρήστης έχει ή όχι την εξουσιοδότηση να εκτελέσει κάποια λειτουργία. Αν ο πελάτης, που ζητάει την εκτέλεση μιας λειτουργίας, είναι κάποιο συστατικό μέρος του συστήματος, τότε η εκτέλεση γίνεται άμεσα. Αν ο πελάτης είναι κάποιο υποσύστημα του συστήματος **ARION** και ζητάει την εκτέλεση μιας λειτουργίας εκ μέρους ενός χρήστη, τότε ο υπηρετητής οφείλει να συνεργαστεί με το σύστημα **Ταυτοποίησης και Εξουσιοδότησης Χρήστη**, το οποίο θα εξετάσει αν θα πρέπει να εξουσιοδοτήσει αυτόν το χρήστη. Η τελευταία περίπτωση, αφορά ένα κομμάτι των **Γραφικών Διεπαφών Χρήσης** ή τον **Κειμενογράφο Ροών Εργασίας**, ο οποίος και έχει υλοποιηθεί με τη μορφή Java Applet. Οι Γραφικές Διεπαφές Χρήσης είναι δυναμικές σελίδες του διαδικτύου ή Java Applets και μεταφέρουν το αντικείμενο, που προκύπτει από την ταυτοποίηση του χρήστη. Οπότε, γίνεται γνωστό ποιος χρήστης προσπαθεί να ζητήσει την εκτέλεση μιας λειτουργίας. Από την άλλη μεριά, τόσο ο **Κειμενογράφος Ροών Εργασίας** όσο και ο **Κειμενογράφος Περιπτώσεων Ροών Εργασίας**, που λειτουργούν τοπικά, ταυτοποιούν το χρήστη κατά την έναρξη της λειτουργίας τους και αναλαμβάνουν να εξετάσουν εάν του έχει χορηγηθεί η εν λόγω εξουσιοδότηση. Έπειτα, ζητούν την εκτέλεση μιας λειτουργίας από ένα υπηρετητή-συστατικό μέρος του συστήματός μας. Συνεπώς, το σύστημά μας έχει σχεδιαστεί, ώστε να είναι αρκετά ασφαλές, μια και ο κάθε χρήστης δεν μπορεί να εκτελεί λειτουργίες ανεξέλεγκτα. Τώρα, πλέον, απομένει η Φάση της Υλοποίησης, η οποία θα εξασφαλίσει και θα προσδώσει τα προαναφερόμενα χαρακτηριστικά στο σύστημά μας.

5.4. Υλοποίηση του συστήματος

Η υλοποίηση συστήματος είναι το επιστέγασμα των προσπαθειών ανάπτυξης ενός συστήματος και είναι εξίσου σημαντική με τις υπόλοιπες διαδικασίες ανάπτυξης. Ο κύριος οδηγός της διαδικασίας της υλοποίησης είναι η αρχιτεκτονική του συστήματος **K.A.P.E.E.A**. Κατά τη διάρκεια της υλοποίησης λαμβάνονται αποφάσεις, που μπορεί να οδηγήσουν ακόμη και σε ανασχεδιασμό της αρχιτεκτονικής. Η υλοποίηση ενός συστήματος είναι μια αρκετά χρονοβόρα διαδικασία και, συνήθως, προτού εκτελεστεί πραγματοποιούνται προσπάθειες τμηματοποίησής της σε κομμάτια, πολλά από τα οποία δύναται να εκτελεστούν παράλληλα, προκειμένου να μειωθεί στο ελάχιστο ο χρόνος εκτέλεσής της.

Στην περίπτωση της υλοποίησης του συστήματος **K.A.P.E.E.A**, χρησιμοποιήθηκε η γλώσσα προγραμματισμού Java, μια αντικειμενοστραφής γλώσσα προγραμματισμού, ενώ η όλη διαδικασία επιτελέστηκε από ένα άτομο – αφού το σύστημα αυτό αναπτύχθηκε στα πλαίσια Μεταπτυχιακής Εργασίας. Η γλώσσα προγραμματισμού Java είναι από τις κορυφαίες γλώσσες προγραμματισμού και χρησιμοποιείται στην πλειοψηφία των προγραμμάτων-συστημάτων, που

αναπτύσσονται σήμερα. Η ευρεία ανταπόκριση της οφείλεται στα ορισμένα ακόλουθα χαρακτηριστικά :

- Υποστηρίζει τον αντικειμενοστραφή προγραμματισμό. Ο αντικειμενοστραφής προγραμματισμός βασίζεται στη δημιουργία αντικειμένων, που παρουσιάζουν συγκεκριμένες ιδιότητες και εκτελούν προκαθορισμένες λειτουργίες. Από τις λειτουργίες αυτές προκύπτουν άμεσα αποτέλεσμα, όπως η δημιουργία ενός νέου αντικειμένου (κάποιου άλλου τύπου). Κάθε αντικείμενο ενδέχεται να έχει και ιδιωτικές λειτουργίες, που ενώ είναι κρυφές υποστηρίζουν, ωστόσο, εκείνες που χαρακτηρίζονται ως δημόσιες. Ο εν λόγω προγραμματισμός στηρίζεται στον ορισμό τύπων αντικειμένων (κλάσεων), που εμφανίζουν τόσο ιδιωτικές και δημόσιες ιδιότητες όσο και ιδιωτικές και δημόσιες λειτουργίες. Οι δημόσιες ιδιότητες και λειτουργίες μιας κλάσης ονομάζονται διεπαφή χρήσης (interface) της εν λόγω κλάσης. Ένα πρόγραμμα δημιουργεί αντικείμενα ορισμένων κλάσεων και εκμεταλλεύεται στο έπακρο τις ιδιότητες και τις λειτουργίες τους, παρέχοντας μια συγκεκριμένη λειτουργικότητα. Ένα άλλο ενδιαφέρον χαρακτηριστικό του αντικειμενοστραφούς προγραμματισμού είναι η κληρονομικότητα. Μια κλάση μπορεί να κληρονομήσει τις ιδιότητες και τις λειτουργίες μιας άλλης κλάσης, εάν θεωρηθεί ότι την επεκτείνει, ή με άλλα λόγια ότι είναι μια ειδίκευση της, όπως ακριβώς συμβαίνει με τις κλάσεις 'λαγός' και 'ζώο'. Ο αντικειμενοστραφής προγραμματισμός είναι αρκετά δημοφιλής, διότι πλησιάζει κατά πολύ τον ανθρώπινο τρόπο σκέψης: θεωρεί ότι ο κόσμος, που τον περιβάλλει, αποτελείται από αντικείμενα που παρουσιάζουν ιδιότητες - αλληλεπιδρούν μεταξύ τους - ενεργούν κατά ένα συγκεκριμένο τρόπο. Επίσης, η ανθρώπινη σκέψη παρουσιάζει την τάση να ομαδοποιεί τα αντικείμενα σε κατηγορίες, ανάλογα με τις ιδιότητες τους.
- Η Java παρέχει ένα πλήθος από πακέτα κλάσεων. Το κάθε πακέτο εφαρμόζεται σε διάφορα πεδία του προγραμματισμού (προγραμματισμός στο δίκτυο, γραφικός προγραμματισμός) και αποτελείται από ορισμούς κλάσεων. Κάθε κλάση προσφέρει ορισμένες ιδιότητες και λειτουργίες, που έχουν άμεση σχέση με το πεδίο στο οποίο εφαρμόζεται η κλάση. Κατά συνέπεια, ο προγραμματιστής δύναται αφενός να χρησιμοποιήσει de facto λειτουργίες, όπως η επικοινωνία με το λειτουργικό σύστημα και η επικοινωνία με το δίκτυο και αφετέρου να οργανώσει την δική του εφαρμογή. Εν συνεχεία, η εφαρμογή του προγραμματιστή οργανώνεται σε μια ιεραρχία από πακέτα, που το καθένα υποδηλώνει κατηγορία από λειτουργίες. Έτσι λοιπόν, κάθε άλλος προγραμματιστής θα διευκολύνεται στη χρησιμοποίηση της εν λόγω εφαρμογής, σε περίπτωση που κρίνει σκόπιμο να εκμεταλλευτεί τη λειτουργικότητά της.
- Η Java διευκολύνει τον προγραμματισμό στο δίκτυο, διότι μέσω των πακέτων της net και rmi (Remote Method Invocation) παρέχει βασικούς τρόπους σύνδεσης στο διαδίκτυο. Βασική κλάση της είναι η 'Socket', που αναπαριστά το αμφίδρομο (από πλευράς προώθησης δεδομένων) κανάλι επικοινωνίας μεταξύ δύο κόμβων του διαδικτύου. Το κανάλι υπακούει στο πρωτόκολλο TCP. Ένα κανάλι επικοινωνίας μπορεί να γίνει ασφαλές, όταν αποτελέσει αντικείμενο της κλάσης SSLSocket. Δηλαδή, όταν η μεταφορά δεδομένων στο κανάλι ακολουθεί μια κωδικοποίηση, που βασίζεται στο πρωτόκολλο SSL, το οποίο και δομείται πάνω από το πρωτόκολλο TCP. Το πακέτο rmi προσφέρει ένα συγκεκριμένο μηχανισμό απομακρυσμένης εκτέλεσης λειτουργίας-μεθόδου, που μπορεί με ρυθμίσεις να παρέχει ασφάλεια και, συνάμα, να μειώνει την ταχύτητα της εκτέλεσης. Με τον όρο 'απομακρυσμένη εκτέλεση' εννοούμε την εκτέλεση μιας λειτουργίας-

μεθόδου ενός προγράμματος από ένα άλλο πρόγραμμα, που εκτελείται σε διαφορετικό υπολογιστή.

- Η Java διευκολύνει τη σύνδεση ενός προγράμματος με μια βάση δεδομένων και την εκτέλεση εντολών σε αυτή, μέσω του πακέτου με όνομα `sql`. Μάλιστα, αυτή η σύνδεση μπορεί να επιτευχθεί ακόμη και στην περίπτωση που το πρόγραμμα δεν βρίσκεται στον ίδιο υπολογιστή με τη βάση δεδομένων.
- Η Java διευκολύνει αφενός την επικοινωνία ενός προγράμματος με το λειτουργικό σύστημα στο οποίο εκτελείται και αφετέρου την εκτέλεση διαφόρων εντολών λειτουργικού συστήματος, όπως είναι η δημιουργία και η αντιγραφή αρχείου ή αποθήκευσή του στο υποσύστημα αρχείων του λειτουργικού συστήματος. Επιπλέον, μπορεί να χωρίσει την εκτέλεση ενός προγράμματος σε πολλά κομμάτια (threads), που θα εκτελούνται (ψευδό)παράλληλα, αφαιρώντας-κλέβοντας έτσι χρόνο εκτέλεσης από τον χρονοπρογραμματισμό διεργασιών του λειτουργικού συστήματος.
- Η Java προσφέρει ένα πλήρες πακέτο για τον γραφικό προγραμματισμό, δηλαδή για τη δημιουργία διεπαφών χρήσης και εφαρμογών, όπως είναι οι κειμενογράφοι.
- Εκτός από τα βασικά, προσφέρονται και άλλης κατηγορίας Java πακέτων στο διαδίκτυο. Ένα από αυτά είναι και το `dom4j` [91], που χρησιμοποιείται τόσο για την ‘φόρτωση’ XML αρχείων σε μορφή γραφικού δένδρου (JTree) όσο και για την αντίστροφη ‘εκφόρτωση’ γραφικών δένδρων σε XML αρχεία. Επίσης, προσφέρει λειτουργίες γραμματικής ανάλυσης (parsing) ενός XML αρχείου. Το πακέτο `dom4j` χρησιμοποιήθηκε στην ανάπτυξη του συστήματος **K.A.P.E.E.A**.

Η υλοποίηση του συστήματος **K.A.P.E.E.A** έγινε με τη δημιουργία μιας ιεραρχίας πακέτων. Η ιεραρχία είναι το πολύ τριών επιπέδων. Κάθε πακέτο παρέχει συγκεκριμένη λειτουργικότητα με τον ορισμό αντίστοιχων κλάσεων. Στη συνέχεια αυτής της ενότητας, θα περιγράψουμε εν συντομία τη λειτουργικότητα του κάθε πακέτου, θα αναλύσουμε τον τρόπο που εκτελούνται οι απομακρυσμένες λειτουργίες, θα διευκρινίσουμε από ποια αντικείμενα αποτελούνται τα συστατικά μέρη του συστήματος **K.A.P.E.E.A** και, τέλος, πώς επιτυγχάνονται οι λειτουργίες αυτών των μερών.

Τα πακέτα, που βρίσκονται στην κορυφή της Ιεραρχίας Πακέτων της Υλοποίησης του συστήματός μας είναι τα ακόλουθα :

- `Comm`: Σε αυτό το πακέτο ορίζονται οι κύριες οντότητες – υπηρετητές του συστήματός μας, καθώς επίσης και ο τρόπος εκτέλεσης των λειτουργιών τους.
- `Filetransfer`: Σε αυτό το πακέτο ορίζονται οι κλάσεις, που βοηθούν αφενός την αποθήκευση αρχείων σε έναν υπολογιστή και αφετέρου τη μεταφορά αρχείων από ένα κόμβο-υπολογιστή σε έναν άλλο. Επίσης, ορίζονται και βοηθητικές λειτουργίες, όπως εκείνη του ‘remote dir’, δηλαδή της απομακρυσμένης αναφοράς της λίστας αρχείων ενός συγκεκριμένου καταλόγου υπολογιστή.
- `Gui`: Σε αυτό το πακέτο ορίζονται οι κλάσεις του **Κειμενογράφου Ροών Εργασίας** και του **Κειμενογράφου Περιπτώσεων Ροών Εργασίας**. Παράλληλα, ορίζονται και βοηθητικές κλάσεις, που είτε συντελούν στην επίτευξη λειτουργιών των κειμενογράφων είτε αντιπροσωπεύουν γραφικά αντικείμενα, όπως γραφικούς διάλογους ή φόρμες, τα οποία με τη σειρά τους καθοδηγούν διαισθητικά τους χρήστες να επιτύχουν τον επιθυμητό σκοπό – λειτουργία.
- `Ontotree`: Το πακέτο αυτό παρέχει λειτουργίες σύνδεσης με τη **Μηχανή Αναζήτησης** και άντλησης πληροφορίας από αυτήν. Η πληροφορία μπορεί να αφορά είτε το σχήμα της επιστημονικής οντολογίας είτε μετα-δεδομένα για ένα συγκεκριμένο σύνολο δεδομένων. Επίσης, σε αυτό το πακέτο ορίζεται μια

γραφική κλάση, που και αναπαριστά το δένδρο-ιεραρχία της επιστημονικής οντολογίας και οδηγεί διαισθητικά το χρήστη να ορίσει τις κλάσεις της οντολογίας που θα ανήκει ένα σύνολο δεδομένων της εισόδου ή της εξόδου ενός προγράμματος καθορισμένου έργου.

- **Org.dom4j**: Αυτό είναι το έτοιμο πακέτο, που χρησιμοποιήθηκε για τη διαχείριση XML αρχείων και τη μετατροπή τους σε γραφικά δένδρα (JTrees). Παρέχει λειτουργίες γραμματικής ανάλυσης ενός XML αρχείου και επαλήθευσής του (validation), σύμφωνα με ένα συγκεκριμένο αρχείο τύπου DTD. Πρέπει να τονισθεί, ότι ορισμένες κλάσεις αυτού του πακέτου τροποποιήθηκαν, ενώ παράλληλα δημιουργήθηκαν και νέες. Και αυτό διότι το γραφικό δένδρο ήταν στατικό, και κατά συνέπεια δε μπορούσε να τροποποιηθεί ή να επεκταθεί.
- **Parser**: Αυτό το πακέτο παρέχει τον ορισμό μιας κλάσης αντικειμένων, που λαμβάνει ως είσοδο τον ορισμό μιας περίπτωσης ροής εργασίας (δηλαδή ένα αρχείο) και παράγει, κατόπιν γραμματικής ανάλυσεως, ένα νέο αντικείμενο. Το παραγόμενο αντικείμενο μεταφέρεται, μέσω του μηχανισμού serialization της Java, στο σύστημα **Εκτέλεσης Ροών Εργασίας** για να εκτελεστεί, και άρα να εκτελεστεί η συγκεκριμένη περίπτωση ροής εργασίας.
- **Xml**: Το πακέτο παρέχει δύο κλάσεις, που εκμεταλλεύονται τις λειτουργίες του πακέτου org.dom4j, για να διευκολύνουν τη 'φόρτωση' XML αρχείων σε γραφικά δένδρα και την 'εκφόρτωση' γραφικών δένδρων σε XML αρχεία.
- **Xml2java**: Αυτό το πακέτο ορίζει τις κλάσεις, που αντιπροσωπεύουν τα στοιχείο-δομικά μέρη της γλώσσας ορισμού ροών εργασίας. Οι κλάσεις χρησιμεύουν για την αποθήκευση της πληροφορίας, η οποία και δίνεται από τους χρήστες στους γραφικούς διάλογους του πακέτου 'gui', και το πέρασμά της στο γραφικό δένδρο αναπαράστασης ροής εργασίας.
- **Aa**: Το πακέτο χρησιμεύει για την επικοινωνία με το σύστημα **Ταυτοποίησης και Εξουσιοδότησης Χρήστη**, ώστε αυτό να αποφανθεί αν ένας συγκεκριμένος χρήστης έχει εξουσιοδότηση στη διαχείριση ενός συγκεκριμένου αντικειμένου.
- **ARION.workflow**: Το πακέτο χρησιμοποιείται για το πέρασμα του ορισμού περίπτωσης ροής εργασίας στο σύστημα **Εκτέλεσης Ροών Εργασίας**.

Οι απομακρυσμένες λειτουργίες του συστήματός μας εκτελούνται με ένα ειδικό τρόπο. Έστω ότι υπάρχει ένα υπηρετητής με όνομα MyServer και θέλουμε να εκτελέσουμε σε αυτόν μια λειτουργία. Για κάθε τέτοιο υπηρετητή, θα υπάρξει-ορίζεται ένας αντίστοιχος πελάτης MyClient, που θα συνδέεται με τον υπηρετητή. Για κάθε λειτουργία του υπηρετητή, θα υπάρχει αντίστοιχη λειτουργία του πελάτη, που θα προκαλεί το ίδιο αποτέλεσμα. Οι λειτουργίες του πελάτη θα εκτελούν ένα πρωτόκολλο επικοινωνίας με τον υπηρετητή, το οποίο και θα βασίζεται σε ένα νέο αντικείμενο MyOpts και θα είναι το εξής :

Στείλε κωδικό λειτουργίας (π.χ MyOpts.MyFunction).

Στείλε δεδομένα, που αφορούν την αρχικοποίηση της λειτουργίας.

Λάβε πίσω το αποτέλεσμα.

Αν για κάποιο λόγο χαθεί η σύνδεση καθώς εκτελείται κάποιο βήμα του πρωτοκόλλου, τότε ο παραπάνω κώδικας θα εκτελεστεί μέχρι δύο φορές. Αν και πάλι υπάρξει πρόβλημα με την επικοινωνία, τότε η απομακρυσμένη εκτέλεση της λειτουργίας αποτυγχάνει. Το ίδιο συμβαίνει και στην περίπτωση που το αποτέλεσμα είναι null ή MyOpts.FAILED. Από την άλλη πλευρά, ο κώδικας, που εκτελείται στον υπηρετητή, για κάθε αίτηση ενός πελάτη είναι :

Λάβε το είδος της λειτουργίας (π.χ. MyOpts.MyFunction).
Λάβε τα δεδομένα εκκίνησης της λειτουργίας.
Εκτέλεσε τη λειτουργία.
Στείλε το αποτέλεσμα (π.χ. MyOpts.SUCCEDED ή MyOpts.FAILED).

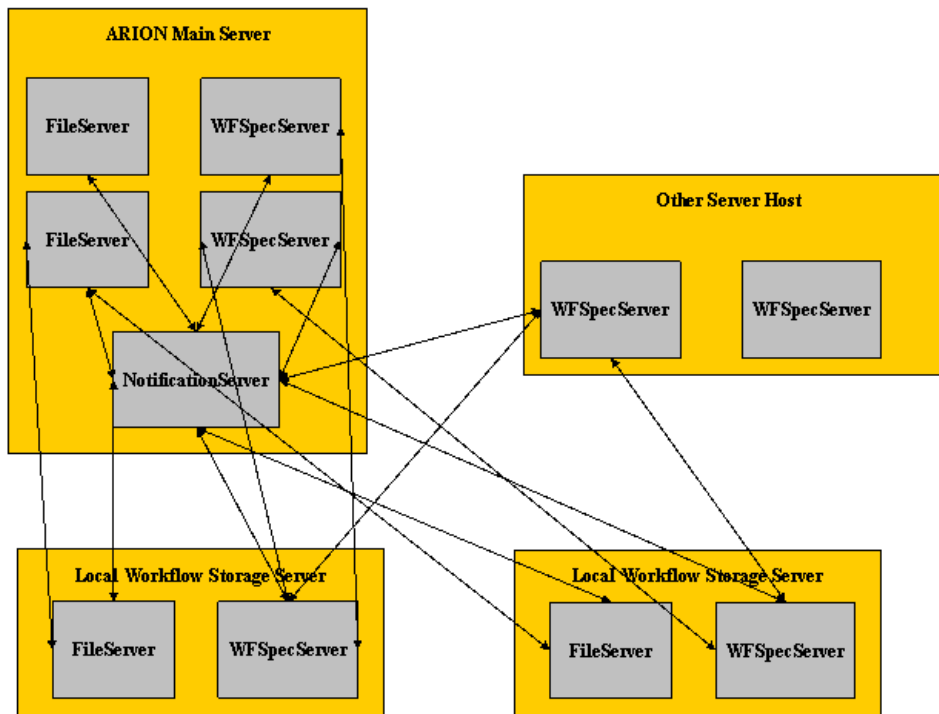
Αν για κάποιο λόγο χαθεί η σύνδεση με τον πελάτη, πριν από την εκτέλεση της λειτουργίας, τότε η εκτέλεση του κώδικά σταματά. Αν η σύνδεση χαθεί έπειτα, τότε απλώς δεν εκτελείται το επόμενο βήμα, δηλαδή το βήμα της αποστολής του αποτελέσματος. Επομένως, στην τελευταία περίπτωση, ο πελάτης δεν θα γνωρίζει το αποτέλεσμα της εκτέλεσης, αλλά η λειτουργία θα έχει εκτελεστεί. Αν ο πελάτης, εν συνεχεία, ζητήσει να επανεκτελέσει την ίδια λειτουργία με τα ίδια δεδομένα, τότε η λειτουργία θα εκτελεστεί ξανά και το αποτέλεσμα θα είναι ακριβώς το ίδιο. Και αυτό, διότι τα δύο είδη λειτουργιών του συστήματός μας, η αποθήκευση αντικειμένων και η επερώτηση για αντικείμενα, όσες φορές και αν εκτελεστούν θα επιφέρουν το ίδιο αποτέλεσμα. Επιπλέον, η αποθήκευση ορισμών (περιπτώσεων) ροών εργασίας, όσες φορές και αν εκτελεστεί, πάλι θα προκαλέσει το ίδιο αποτέλεσμα. Η αποθήκευση στατιστικών δεδομένων για μια περίπτωση ροής εργασίας, την πρώτη φορά θα έχει επιτυχές αποτέλεσμα, ενώ τις επόμενες φορές δε θα εκτελείται, λόγω παραβίασης του περιορισμού ακεραιότητας της βάσης δεδομένων. Όμως, εάν ο υπηρετητής ρυθμιστεί να απαντάει με MyOpts.SUCCEDED, κάθε φορά που γίνεται παράβαση τέτοιου είδους, τότε δε θα προκύπτει απολύτως κανένα πρόβλημα. Επιπλέον, επειδή οι επερωτήσεις δεν επιφέρουν αλλαγές, αλλά επιστρέφουν αποτελέσματα-αντικείμενα, έπεται ότι όσες φορές και αν εκτελεστούν πάλι δεν θα προκύπτει κανένα πρόβλημα. Απεναντίας, όσο αργεί η απάντηση, που θα πάρει ο χρήστης λόγω του δικτύου, τόσο καλύτερο θα είναι το αποτέλεσμα, αφού υπάρχει περίπτωση κάτι να αλλάξει στη βάση δεδομένων ή στην αποθήκη και το αποτέλεσμα να είναι ποιοτικά καλύτερο. Πρέπει να σημειωθεί, ότι για την επικοινωνία πελάτη με υπηρετητή χρησιμοποιήθηκε η κλάση SSLSocket της Java, που προσφέρει ασφαλή και αξιόπιστη επικοινωνία μεταξύ των συμμετεχόντων μερών.

Πριν προχωρήσουμε στην ανάλυση του πώς υλοποιήθηκαν τα συστατικά μέρη του συστήματος **K.A.P.E.E.A**, θα πρέπει να αναφέρουμε ότι ο χρόνος υλοποίησης ήταν περιορισμένος και επικεντρώθηκε, κυρίως, στη δημιουργία των Κειμενογράφων και των Διεπαφών Χρήσης. Η Υλοποίηση των συστατικών μερών έγινε σύμφωνα με τις προδιαγραφές της αρχιτεκτονικής του συστήματος **K.A.P.E.E.A**.

Η **Αποθήκη Ροής Εργασίας** σχεδιάστηκε τελικώς να είναι ένας κατάλογος στο λειτουργικό σύστημα ενός υπολογιστή του συστήματος (μπορεί να είναι ο **Κύριος Υπηρετητής** του ARION ή ένας υπολογιστής του τοπικού συστήματος ARION) με μια ιδιαίτερη δομή. Το όνομα του καταλόγου είναι 'workflows'. Μέσα στον κατάλογο αυτό υπάρχουν τόσοι υποκατάλογοι όσοι είναι και οι επιστημονικοί οργανισμοί, που συμμετέχουν στο σύστημα **ARION**, και με ονομασία τον τίτλο του οργανισμού στον οποίο ανήκουν-αντιστοιχούν. Επιπλέον, υπάρχει ένας υποκατάλογος με όνομα 'inter', ο οποίος και περιέχει τους ορισμούς των ροών εργασίας, οι οποίες περιλαμβάνουν προγράμματα και δεδομένα που ανήκουν σε διάφορους επιστημονικούς οργανισμούς. Κάθε άλλος υποκατάλογος περιέχει τους ορισμούς των ροών εργασίας, που αφορούν ένα συγκεκριμένο επιστημονικό οργανισμό. Για παράδειγμα, οι ορισμοί ροών εργασίας για τον οργανισμό με όνομα 'MyCompany' θα βρίσκονται στο μονοπάτι του λειτουργικού συστήματος '/workflows/MyCompany/'. Οι ορισμοί περιπτώσεων ροών εργασίας αποθηκεύονται

όλοι μαζί στον κατάλογο 'instances', έχοντας ως όνομα το ID της περίπτωσης της ροής εργασίας συν την κατάληξη '.xml'. Αν τώρα ο χρήστης θέλει να ορίσει μια περίπτωση ροής εργασίας, που θα τρέχει στο background, τότε θα πρέπει να δημιουργηθεί ένα επιπλέον αρχείο με το ίδιο ακριβώς όνομα, αλλά με την κατάληξη '.data'. Την περίπτωση αυτή θα την περιγράψουμε παρακάτω. Επομένως, ο ορισμός μιας περίπτωσης ροής εργασίας με ID: WF_32632788, θα αποθηκεύεται με το εξής όνομα: 'instances/WF_32632788.xml', ενώ αν εκτελείται στο background θα συνοδεύεται από ένα αρχείο με όνομα: 'instances/WF_32632788.data'.

Ο **Υπηρετητής Αποθήκης Ροών Εργασίας** υλοποιείται από δύο οντότητες-κλάσεις: την FileServer και την WFSpecServer, οι οποίες και είναι τύπου Thread και ανήκουν στα πακέτα 'filetransfer' και 'comm.server' αντίστοιχα. Ο FileServer είναι υπεύθυνος να αποθηκεύει τα αρχεία (τους ορισμούς –περιπτώσεων- ροών εργασίας), που του στέλνει ένας FileClient ('filetransfer'). Τα αντικείμενα, που ανήκουν στην κλάση FileServer, θα τρέχουν μόνο στον **Κύριο Υπηρετητή του ARION**, εκεί δηλαδή που βρίσκεται και η **Κεντρική Αποθήκη Ροών Εργασίας**. Κάθε αντικείμενο FileClient θα βρίσκεται στον υπολογιστή ενός τοπικού συστήματος **Κ.Α.Ρ.Ε.Ε.Α**. Ο WFSpecServer είναι υπεύθυνος να απαντάει στις ερωτήσεις, που σχετίζονται με ορισμούς ροών εργασίας που του στέλνει ένα αντικείμενο τύπου WFSpecClient ('comm.client'). Για κάθε ερώτηση, ο WFSpecServer βλέπει, σε συνεργασία με ένα NotificationServer, σε ποια τοπικά συστήματα μπορεί να κατανέμει την ερώτηση, δημιουργεί τόσα αντικείμενα WFSpecClient όσα είναι και τα τοπικά συστήματα και στέλνει τις ερωτήσεις με τη βοήθεια των WFSpecClient στα τοπικά αντικείμενα WFSpecServer. Οι υπόλοιποι υποκατάλογοι (των οργανισμών που είτε δεν έχουν εγκαταστήσει το τοπικό σύστημα **Κ.Α.Ρ.Ε.Ε.Α** είτε το τοπικό τους σύστημα δεν μπορεί να προσεγγιστεί δικτυακά από τον WFSpecServer) μαζί με τον υποκατάλογο 'workflows/inter' ελέγχονται από τον ίδιο τον WFSpecServer. Έπειτα, ο WFSpecServer συλλέγει όλες τις απαντήσεις και στέλνει πίσω μια ενιαία απάντηση στον WFSpecClient, ο οποίος και διατύπωσε την ερώτηση. Επιπλέον, κάθε αντικείμενο τύπου WFSpecServer ενημερώνει, ανά τακτά χρονικά διαστήματα, τον NotificationServer αν είναι τοπικό ή κεντρικό και σε ποια διεύθυνση βρίσκεται (IP και port). Ένας κεντρικός WFSpecServer μπορεί να τρέχει στον **Κύριο Υπηρετητή του ARION** ή σε υπολογιστές, που είναι στο ίδιο δίκτυο LAN με αυτόν του κύριου υπηρετητή. Αν τρέχει στον κύριο υπηρετητή, τότε η διαδικασία επιτέλεσης ερωτήσεων είναι η παραπάνω. Αν τρέχει σε κοντινούς υπολογιστές, τότε το αντικείμενο του (κεντρικού) WFSpecServer θα αναλάβει την κατανομή ερωτήσεων και θα στείλει μια ειδική ερώτηση σε ένα αντικείμενο WFSpecServer, που τρέχει στον **Κύριο Υπηρετητή του ARION**, για εκείνους τους υποκαταλόγους που έχουν μείνει χωρίς έλεγχο. Εν συνεχεία, ο WFSpecServer συλλέγει όλες τις απαντήσεις και επιστρέφει μια ενιαία απάντηση σε αυτόν που διατύπωσε την ερώτηση. Η τελευταία διαδικασία οφείλει να πραγματοποιηθεί απαραίτητως, για να μην επιβαρύνεται ο **Κύριος Υπηρετητής του ARION**, μια και η **Κεντρική Αποθήκη Ροών Εργασίας** βρίσκεται κατά αποκλειστικότητα στον **Κύριο Υπηρετητή του ARION**. Τέλος, το αντικείμενο τύπου WFSpecServer, που τρέχει σε ένα τοπικό σύστημα **Κ.Α.Ρ.Ε.Ε.Α**, εκτελεί μια ερώτηση, τοπικά, και αν δε βρει αποτέλεσμα απευθύνεται σε ένα κεντρικό αντικείμενο τύπου WFSpecServer. Στο τέλος, συλλέγει την απάντηση που του στέλνεται. Η εικόνα των δύο συστατικών μερών για το πώς υλοποιήθηκαν, πως συνδέονται και σε ποιους υπολογιστές βρίσκονται φαίνεται στο σχήμα 20.



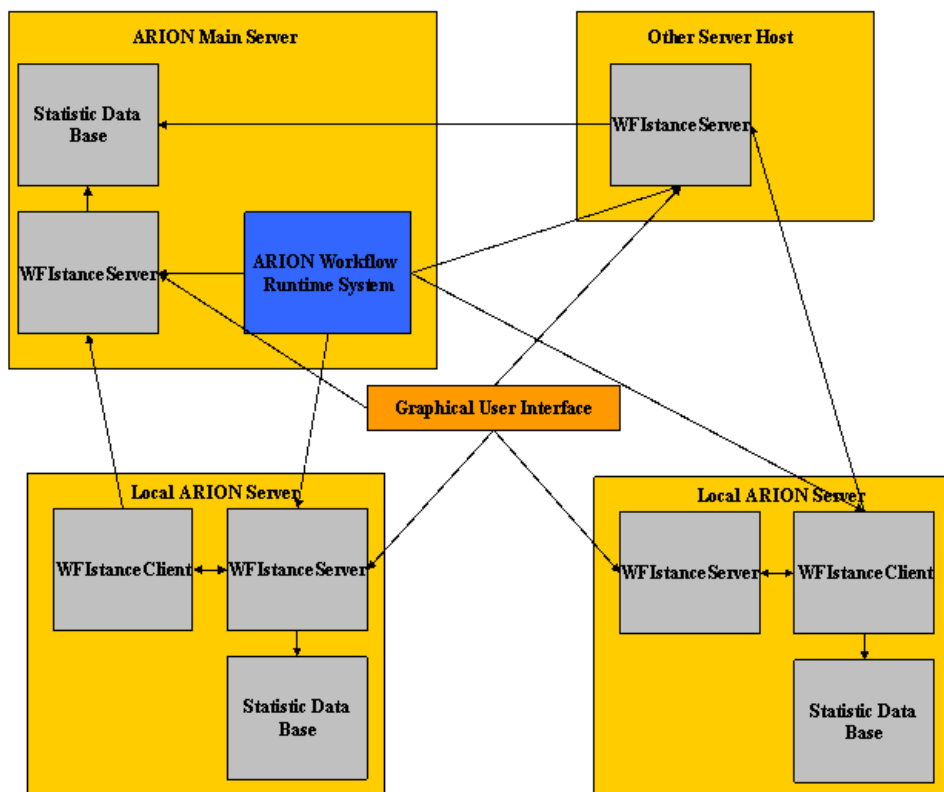
Σχήμα 20 - Αρχιτεκτονική του Workflow Storage Server

Η **Στατιστική Βάση Δεδομένων** υλοποιήθηκε με τη χρησιμοποίηση μιας οντοσχεσιακής βάσης δεδομένων, την PostGres. Αυτή η βάση δεδομένων υπάρχει ήδη στο σύστημα του **ARION** και χρησιμεύει για την αποθήκευση των μετα-δεδομένων και του σχήματος της επιστημονικής οντολογίας από το εργαλείο RDFSuite. Οπότε και κρίθηκε σκόπιμο να μη χρησιμοποιηθεί κάποια νέα βάση δεδομένων. Στη βάση αποθηκεύθηκαν τρία σχήματα, που σχετίζονται με τη στατιστική πληροφορία. Η στατιστική πληροφορία σχετίζεται αφενός με το μέσο κόστος και το μέσο χρόνο εκτέλεσης μιας ροή εργασίας και αφετέρου με το κόστος και το χρόνο εκτέλεσης μιας περίπτωση ροής εργασίας. Προφανώς, το περιεχόμενο της στατιστικής πληροφορίας μπορεί να επεκταθεί και σε άλλες παραμέτρους εκτέλεσης. Το πρώτο σχήμα `wf_class(String ID, String name, String descr, int avg_cost, int avg_run_time)` αφορά την καταχώρηση στατιστικής πληροφορίας για μια ροή εργασίας, που έχει συγκεκριμένο ID, όνομα και περιγραφή. Η αρχική εγγραφή για μια ροή εργασίας στο σχήμα αποθηκεύεται μόλις ολοκληρωθεί ο ορισμός της και τοποθετηθούν μηδενικές τιμές στα πεδία του μέσου κόστους και του μέσου χρόνου εκτέλεσης. Το δεύτερο σχήμα `workflow(String ID, int cost, int run_time)` αφορά την καταχώρηση στατιστικής πληροφορίας για μια περίπτωση ροής εργασίας. Το τρίτο σχήμα `wf_map(String WF_ID, String WFC_ID)` αφορά την αντιστοίχιση μιας περίπτωσης ροής εργασίας με την ίδια τη ροή εργασίας, μέσω των ID τους. Μόλις οριστεί μια περίπτωση ροής εργασίας, δημιουργούνται δύο εγγραφές: μια για το δεύτερο και μια για το τρίτο σχήμα. Επιπροσθέτως, μόλις εκτελεστεί η περίπτωση ροής εργασίας, ενημερώνεται η δεύτερη εγγραφή στα πεδία του κόστους και του χρόνου εκτέλεσης και ενημερώνεται η αντίστοιχη εγγραφή της ροής εργασίας μέσω ενός sql update, το

οποίο και γίνεται trigger από την προηγούμενη ενημέρωση, ούτως ώστε να αλλαχθούν οι μέσες τιμές του κόστους και του χρόνου εκτέλεσης.

Ο **Υπηρετητής Στατιστικής Βάσης Δεδομένων** υλοποιείται από μία μόνο κλάση, την `WFInstanceServer`, η οποία και βρίσκεται στο πακέτο `'comm.server'`. Το κάθε αντικείμενο τύπου `WFInstanceServer` είναι υπεύθυνο για την καταχώρηση και επερώτηση πληροφορίας στη **Στατιστική Βάση Δεδομένων**. Συγκεκριμένα, καταχωρεί την πληροφορία που αφορά μια ροή εργασίας ύστερα από τον ορισμό της. Δεύτερον, καταχωρεί στατιστική πληροφορία για μια περίπτωση ροής εργασίας. Τρίτον, απαντά ερωτήσεις, σχετικές με τη στατιστική πληροφορία για ροές εργασίας. Οι παραπάνω λειτουργίες ανακινούνται από ένα αντικείμενο τύπου `WFInstanceClient` (`'comm.client'`), που εκτελείται σε ένα πρόγραμμα του συστήματος **Εκτέλεσης Ροών Εργασίας**. Οι επερωτήσεις αποφασίστηκε, τελικώς, να απευθύνονται μόνο στην **Κεντρική Στατιστική Βάση Δεδομένων**, η οποία και βρίσκεται στον υπολογιστή του **Κυρίου Υπηρετητή** του **ARION**. Λόγω του τρόπου που γίνεται η σύνδεση ενός Java προγράμματος με ένα σύστημα διαχείρισης βάσεων δεδομένων, όπως αυτό της `PostGres`, το κάθε κεντρικό αντικείμενο τύπου `WFInstanceServer` δεν είναι ανάγκη να εκτελείται στον **Κύριο Υπηρετητή** του **ARION**, αλλά θα μπορεί να εκτελείται και σε κοντινούς υπολογιστές του ίδιου δικτύου LAN. Από την άλλη πλευρά, το κάθε τοπικό αντικείμενο `WFInstanceServer` είναι υπεύθυνο για να αποθηκεύει, τοπικά, στατιστικές πληροφορίες. Επιπλέον, είναι υπεύθυνο να διεξάγει τοπικές επερωτήσεις, οι οποίες, ακόμη και αν δεν επιφέρουν αποτέλεσμα, αποστέλλονται σε ένα κεντρικό `WFInstanceServer`. Το σχήμα 21 δείχνει την τοπολογία και τη συνδεσμολογία των αντικειμένων, που υλοποιούν τον Υπηρετητή Στατιστικής Βάσης Δεδομένων.

Ο **Υπηρετητής Ενημέρωσης** υλοποιείται από την κλάση `NotificationServer`, που βρίσκεται στο πακέτο `'comm.server'`. Ανά τακτά χρονικά διαστήματα, τα κεντρικά και τοπικά αντικείμενα `WFSpecServer` και `WFInstanceServer`, μέσω του αντικειμένου `NotificationClient` (`'comm.client'`) που περιέχουν, ενημερώνουν το ένα και μοναδικό αντικείμενο `NotificationServer` (βρίσκεται στον υπολογιστή του **Κυρίου Υπηρετητή του ARION**) που υπάρχει στο σύστημα **K.A.P.E.E.A** είτε ότι υπάρχουν (την πρώτη φορά) είτε ότι είναι `'alive'`. Μόλις το `NotificationServer` καταλάβει ότι ένα αντικείμενο εγγράφηκε ή διαγράφηκε ή δεν απάντησε ότι είναι `'alive'`, μέσα σε ένα μικρό χρονικό περιθώριο, τότε ενημερώνει όλα τα άλλα για την αλλαγή, στέλνοντας μήνυμα στο `NotificationClient` αντικείμενο που περιέχουν. Επίσης, το `NotificationServer` ενημερώνει τον κεντρικό DNS και τους τοπικούς για οποιοσδήποτε αλλαγές στα τοπικά και στα κεντρικά αντικείμενα `WFSpecServer` και `WFInstanceServer` αντίστοιχα. Για οτιδήποτε συμβαίνει, θα είναι ενήμερο και το αντικείμενο `BackUpNotificationServer` (`'comm.server'`) από το αντικείμενο του `NotificationServer`. Το αντικείμενο `BackUpNotificationServer` είναι υπεύθυνο να ανιχνεύει αν το αντικείμενο `NotificationServer` παρουσιάζει δυσλειτουργίες. Σε περίπτωση που ισχύει το τελευταίο, αναλαμβάνει το ρόλο του `NotificationServer`, ενημερώνοντας παράλληλα όλα τα αντικείμενα που επικοινωνούσαν-ενημερώνονταν από τον προηγούμενο `NotificationServer`.



Σχήμα 21 - Αρχιτεκτονική του Statistic Database Server

Ο **Κειμενογράφος Ροών Εργασίας** υλοποιείται από την κλάση ArionEditor, που βρίσκεται στο γραφικό πακέτο gui. Το κύριο στίγμα αυτού του αντικειμένου είναι ότι βοηθάει στον ορισμό και την αποθήκευση ροών εργασίας. Οι ορισμοί ροών εργασίας αποθηκεύονται σε ένα XML αρχείο, που υπακούει στο DTD της γλώσσας ορισμού ροών εργασίας AWL. Το ArionEditor, μέσα από τις γραφικές του λειτουργίες-μέρη, 'φορτώνει' ένα ορισμό ροής εργασίας με χρήση του αντικειμένου 'FileLoader' του πακέτου 'xml'. Το τελευταίο αντικείμενο διαβάζει ένα XML αρχείο ορισμού ροής εργασίας και δημιουργεί ένα αντικείμενο τύπου Document. Μέσω του αντικειμένου Document, δημιουργείται ένα αντικείμενο της κλάσης DocumentTreeModel, η οποία και αποτελεί ειδικευση της κλάσης TreeModel. Η κλάση TreeModel περιγράφει το μοντέλο ενός γραφικού δένδρου και μπαίνει ως είσοδος στην συνάρτηση κατασκευής του αντικειμένου του ίδιου του δένδρου JTree. Η διεπαφή χρήσης του Κειμενογράφου οδηγεί το χρήστη, διαισθητικά, να τροποποιήσει ή να επεκτείνει το γραφικό δένδρο μέσω γραφικών διαλόγων. Οι γραφικοί διάλογοι αφενός προτρέπουν το χρήστη να συμπληρώσει κάποια στοιχεία για τον ορισμό της υπό εξέταση ροής εργασίας και αφετέρου επιστρέφουν τα αντικείμενα, που θα μεταφραστούν ακολούθως σε κόμβους του δένδρου. Κάθε αλλαγή στο δένδρο έχει ως αντίκτυπο και αλλαγή στο μοντέλο του δένδρου αλλά και στο αντικείμενο Document. Συνεπώς, το αντικείμενο ArionEditor, με την υποστήριξη του αντικειμένου DocumentWriter ('xml'), σώζει τον ορισμό εργασίας. Άρα, το αντικείμενο Document σώζεται σε ένα XML αρχείο στον υποκατάλογο 'workflows/MyCompany/', του οργανισμού στον οποίο ανήκει η ροή εργασίας, ακολουθώντας την αντίστροφη πορεία από αυτήν του διαβάσματος ενός XML αρχείου. Κατά την αποθήκευση ενός ορισμού ροής εργασίας, το αντικείμενο ArionEditor ενημερώνει το αντικείμενο FileTransferClient να

αποθηκεύσει τον ορισμό ροής εργασίας και στην **Κεντρική Αποθήκη Ροών Εργασίας**. Αν υπάρξει αποτυχία της κεντρικής αποθήκευσης , τότε το αντικείμενο ArionEditor ενημερώνει το UpdateTool να προωθήσει αυτήν την αλλαγή αργότερα. Ο ορισμός μιας νέας ροής εργασίας αρχίζει από την αρχικοποίηση του αντικειμένου Document, καθώς επίσης και του αντικειμένου JTree. Επιπλέον, για κάθε νέο ορισμό ροής εργασίας, που αποθηκεύεται δημιουργείται ένα μοναδικό ID, μέσω του UIDClient, το οποίο και εγγράφεται τελικά στο XML αρχείο. Παράλληλα, ενημερώνεται ένας τοπικός και ένας κεντρικός WFInstanceServer για να τοποθετήσουν μια νέα εγγραφή στον πίνακα wf_class της Στατιστικής Βάσης Δεδομένων, που έχουν υπό την προστασία τους. Μια σημαντική λειτουργία του ArionEditor είναι ότι επιτρέπει τη συγχώνευση ορισμών ροών εργασίας. Αυτό γίνεται ως εξής: Ο χρήστης σημαδεύει ένα κόμβο του τρέχοντος δένδρου, όπου κάτω από αυτόν θέλει να κρεμάσει ένα νέο δένδρο. Ενημερώνει τον ArionEditor για το ποιον ορισμό ροής εργασίας θα χρησιμοποιήσει και αφού κατασκευαστεί το νέο δένδρο, κόβεται ο αρχικός κόμβος του νέου δένδρου - που αντιστοιχεί στο στοιχείο 'workflow' της γλώσσας ορισμού ροών εργασίας AWL - και το υπόλοιπο υπο-δένδρο τοποθετείται κάτω από τον επιλεγμένο κόμβο του τρέχοντος δένδρου. Περισσότερα στοιχεία για αυτόν τον κειμενογράφο θα περιγραφθούν στο Παράρτημα Γ.

Ο **Κειμενογράφος Περιπτώσεων Ροών Εργασίας** υλοποιείται από την κλάση InstanceEditor του γραφικού πακέτου 'gui'. Το αντικείμενο τύπου InstanceEditor είναι υπεύθυνο για τη μετατροπή ενός ορισμού ροής εργασίας σε ένα ορισμό περίπτωσης ροής εργασίας. Αυτή η μετατροπή γίνεται γραφικά με την προσθήκη πληροφορίας διεύθυνσης υπολογιστή στον οποίο και βρίσκεται κάθε σύνολο δεδομένων, που μπαίνει ως είσοδος σε κάποιο πρόγραμμα ενός έργου. Επιπλέον, δημιουργείται και ένα ID για αυτήν την περίπτωση της ροής εργασίας, μέσω του αντικειμένου UIDClient. Έπειτα, ο προσδιορισμός περίπτωσης ροής εργασίας μεταφέρεται προς εκτέλεση στο σύστημα **Εκτέλεσης Ροών Εργασίας**. Αυτό το σύστημα αλληλεπιδρά με τον χρήστη, ώστε αυτός να δώσει τιμές στις υποχρεωτικές και προαιρετικές παραμέτρους ενός προγράμματος. Αν , όμως, η περίπτωση ροής εργασίας πρέπει να τρέξει στο background, τότε ο προσδιορισμός περίπτωσης ροής εργασίας δεν είναι αρκετός. Θα πρέπει να υπάρχει ένα επιπρόσθετο αρχείο στο οποίο θα περιγράφονται ποιες παράμετροι θα χρησιμοποιηθούν και με ποια τιμή για το πρόγραμμα ενός έργου. Το τελευταίο θα ισχύει αν ο χρήστης επιθυμήσει η συγκεκριμένη περίπτωση ροής εργασίας να εκτελεστεί μόνο μια φορά. Αν ο χρήστης επιθυμήσει η περίπτωση ροής εργασίας να τρέξει πέντε φορές για τις πέντε διαφορετικές τιμές μιας παραμέτρου, τότε το προαναφερόμενο αρχείο θα πρέπει να αναφέρει για κάθε εκτέλεση τις τιμές όλων των παραμέτρων, που θα χρησιμοποιηθούν. Συνεπώς, η δομή αυτού του αρχείου αποφασίστηκε να είναι η εξής:

```
< -----
task ID1 param1='lala1' param2='lala2' ....
task ID2 par1='lal1' par2='lal2' ....
.
.
.
----- >
< -----
task ID1 param1='Lala1' param2='lala2' ...
```

```
task ID2 par1='lal1' par2='lal2' ...
```

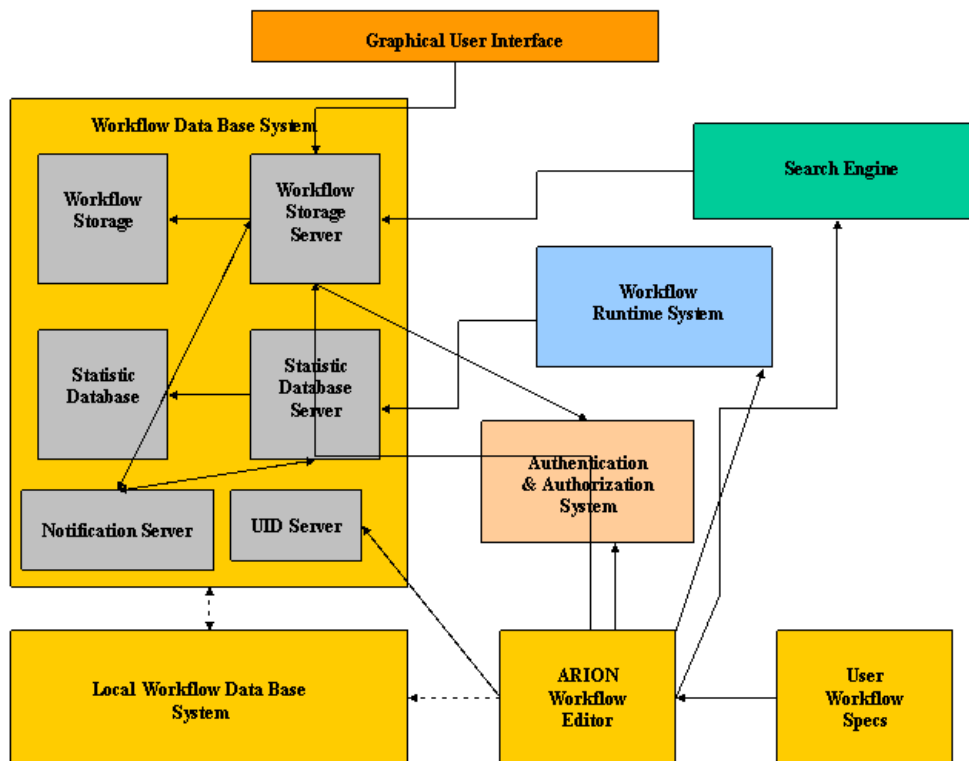
```
.  
. .  
----- >
```

Επομένως, σε αυτό το αρχείο προσδιορίζεται αφενός, ότι κάθε αρχή και τέλος εκτέλεσης θα αναπαρίστανται με ένα ειδικό αλφαριθμητικό, και αφετέρου, ότι σε κάθε γραμμή μιας εκτέλεσης θα αναγράφονται με συγκεκριμένη σειρά: το ID του έργου που χρειάζεται παραμέτρους, η τιμή της κάθε παραμέτρου. Αναλυτικότερα, αν μια παράμετρος έχει το όνομα param και την τιμή lala, τότε θα δίνεται με τη μορφή param='lala'. Μόλις ο χρήστης ζητήσει την εκτέλεση μιας περίπτωσης ροής εργασίας, τόσο το αρχείο προσδιορισμού της περίπτωσης όσο και το αρχείο προσδιορισμού παραμέτρων εκτέλεσης θα αποθηκεύονται και στο τοπικό σύστημα αλλά και κεντρικά. Επιπλέον, θα ενημερώνεται ένα κεντρικό αντικείμενο WFInstanceServer για την αντιστοίχιση του ID της ροής εργασίας με αυτό της περίπτωσης ροής εργασίας. Περισσότερα στοιχεία για αυτόν τον κειμενογράφο θα περιγραφθούν στο Παράρτημα Γ.

Το **Εργαλείο Ανανέωσης** υλοποιείται από την αντίστοιχη κλάση UpdateTool, που βρίσκεται στο πακέτο 'comm.server'. Το κάθε υποσύστημα του **ARION**, όπως και κάθε συστατικό μέρος του συστήματός μας ενημερώνει το UpdateTool. Το αντικείμενο UpdateTool εκτελείται-τρέχει στον ίδιο υπολογιστή, που εκτελείται και το συστατικό μέρος του συστήματός μας ή του υποσυστήματος του **ARION**. Στην περίπτωση αποτυχίας μιας απομακρυσμένης λειτουργίας ενός υπηρετητή, αυτό επιτυγχάνεται με τη χρήση του αντικειμένου UpdateClient ('comm.client'). Πιο συγκεκριμένα, υλοποιείται με την προσθήκη μιας γραμμής σε ένα ειδικό αρχείο από το αντικείμενο UpdateClient. Μόλις το UpdateTool αντιληφθεί αυτήν την αλλαγή στο αρχείο, αναλαμβάνει να εκτελέσει την εντολή που του δίνεται στην επικείμενη γραμμή. Όταν επιτύχει η εκτέλεση της εν λόγω εντολής, το UpdateTool διαγράφει τη γραμμή. Ανάλογη διαδικασία ακολουθείται και στην περίπτωση, όπου το UpdateTool εντοπίζει και άλλες, ανάλογες γραμμές στο αρχείο. Βέβαια, απαραίτητη προϋπόθεση είναι να μπορεί να συνδεθεί με τον αντίστοιχο υπηρετητή. Η δυνατότητα σύνδεσης διαπιστώνεται από το αντικείμενο του τύπου NotificationServer.

Εκτός από τις κλάσεις, που υλοποιούν την λειτουργικότητα των συστατικών μερών του συστήματός μας, υπάρχει μια κλάση που χρησιμοποιείται για να προσδίδει μοναδικά ID σε ροές εργασίας ή σε περιπτώσεις αυτών. Επιπροσθέτως, η κλάση μπορεί να επεκταθεί, ώστε να δίνει ID και σε άλλα αντικείμενα του συστήματος (αν υπάρχουν). Ονομάζεται κλάση UIDServer και βρίσκεται στο πακέτο 'comm.server'. Οι υπηρεσίες της μπορεί να προσφερθούν, μέσω απομακρυσμένης πρόσβασης, με τη βοήθεια της κλάσης UIDClient ('comm.client'). Επειδή αυτή η οντότητα οφείλει να είναι μοναδική, δε θα πρέπει να υπάρχουν άλλα προγράμματα που θα την εκτελούν στο σύστημα **ARION**. Επίσης, θα πρέπει να υπάρχει και μια πολιτική υποστήριξης μέσω ενός back up υπηρετητή, όπως γίνεται και με την περίπτωση του Υπηρετητή Ειδοποίησης. Προφανώς, η παραπάνω οντότητα θα αποτελεί ένα συστατικό μέρος του κεντρικού συστήματος **K.A.P.E.E.A**, οπότε και η αρχιτεκτονική του συστήματός μας θα πρέπει να τροποποιηθεί ελάχιστα, όπως χαρακτηριστικά φαίνεται στο σχήμα 22. Το όνομα του νέου συστατικού μέρους θα είναι: **Υπηρετητής Παροχής Μοναδικών Προσδιοριστικών (Unique Identifier Server)**. Συνεπώς, ερχόμαστε

αντιμέτωποι με την περίπτωση, όπου η διαδικασία της υλοποίησης αναθεωρεί την αρχιτεκτονική του συστήματος, η οποία και είχε ήδη καθοριστεί κατά τη διαδικασία της σχεδίασης.



Σχήμα 22 - Η νέα αρχιτεκτονική του K.A.P.E.E.A

Εν κατακλείδι, η διαδικασία της υλοποίησης του συστήματος **K.A.P.E.E.A** στέφθηκε με επιτυχία ακολουθούμενη τις προδιαγραφές της αρχιτεκτονικής του συστήματος. Συνεπώς, κληρονομεί όλα εκείνα τα χαρακτηριστικά που είχαν αποδοθεί στην προσεγμένη σχεδίαση της αρχιτεκτονικής του. Το τελικό σύστημα εκπληρώνει τους αρχικούς στόχους. Τα χαρακτηριστικά, που τονίζουν τη διαφορετικότητα του τελικού συστήματος, είναι :

- **Openness:** Το σύστημα είναι ανοικτό για την προσθήκη νέων λειτουργιών σε αυτό. Επειδή η υλοποίησή του στηρίχθηκε στη γλώσσα προγραμματισμού Java, το σύστημα **K.A.P.E.E.A** δύναται: να εγκατασταθεί σε οποιοδήποτε λειτουργικό σύστημα, να συνδεθεί με οποιαδήποτε βάση, να αποθηκεύσει ένα αρχείο σε ένα συγκεκριμένο μονοπάτι του λειτουργικού συστήματος, να λειτουργήσει πάνω από οποιοδήποτε δίκτυο.
- **Workflow interoperability:** Το σύστημα **K.A.P.E.E.A** παρουσιάζει διαλειτουργικότητα ροών εργασίας. Όλες οι ροές εργασίας ορίζονται σύμφωνα με μια ενιαία γλώσσα προσδιορισμού ροών εργασίας και ερμηνεύονται με κοινό τρόπο από το σύστημα **Εκτέλεσης Ροών Εργασίας**. Αν τώρα, ένας οργανισμός εκφράσει την επιθυμία να συμμετάσχει στο σύστημα **ARION**, χωρίς αλλαγή του συστήματος διαχείρισης ροών εργασίας που διαθέτει, το οποίο και χρησιμοποιεί διαφορετική γλώσσα προσδιορισμού ροών εργασίας, τότε η λύση δίνεται με το να δημιουργηθεί ένα εργαλείο αντιστοίχισης της AWL με την υπάρχουσα γλώσσα.

- Standards compliance: Με τη χρησιμοποίηση των: Java, XML, XSL, XRL και της συμβατικής τεχνολογίας βάσεων δεδομένων, το σύστημά μας συμμορφώνεται με τα διεθνή πρότυπα.

5.5. Επεκτάσεις-Ανοικτά Θέματα

Το σύστημα Κατανεμημένης Αποθήκης Ροών Εργασίας Επιστημονικών Αντικειμένων είναι ένα πρωτότυπο, εξελίξιμο σύστημα που δεν αποτελεί τελικό προϊόν. Επιπροσθέτως, είναι ένα σύστημα ανοικτό σε οποιεσδήποτε τροποποιήσεις ή νεωτερισμούς είτε σε επίπεδο σχεδίασης είτε σε επίπεδο υλοποίησης. Συνεπώς, κρίθηκε σκόπιμο να αναλυθούν αφενός κάποια ανοικτά και κρίσιμα θέματα που αφορούν αυτό το σύστημα και αφετέρου κάποιες τροποποιήσεις-επεκτάσεις που θα το μετατρέψουν σε ένα ανταγωνιστικό προϊόν. Παρακάτω παραθέτουμε αυτά τα ανοικτά θέματα και τις δυνατές τροποποιήσεις του συστήματός μας.

5.5.1. Προσθήκη Οικονομικού Μοντέλου

Είναι σημαντικό να γνωρίζει κάποιος χρήστης πόσο θα του κοστίσει η εκτέλεση μιας περίπτωσης μιας ροής εργασίας, έστω και κατά προσέγγιση. Για αυτό τον λόγο θα πρέπει να υπάρξει ένα οικονομικό μοντέλο χρέωσης αφενός για την εκτέλεση ροών εργασίας και αφετέρου για τη πρόσβαση σε σημαντικά σύνολα δεδομένων στο σύστημα του ARION. Μάλιστα, αυτό το οικονομικό μοντέλο θα μπορούσε να είναι το εξής:

- Η πρόσβαση σε συγκεκριμένο σύνολο δεδομένων θα κοστίζει τόσο όσο είναι το κόστος αυτού του συνόλου δεδομένων.
- Η πρόσβαση σε σύνολο δεδομένων που θα παραχθεί από την εκτέλεση μιας περίπτωσης ροής εργασίας θα κοστίζει τόσο όσο προδίδει η ακόλουθη συνάρτηση: $cost = a * \text{sum_of_costs_of_data_sets_used} + b * \text{sum_of_costs_of_programs_used} + \text{average_cost_of_nodes_used} * \text{num_of_nodes_passed} + c * \text{total_execution_time}$. Συνεπώς, το προαναφερόμενο κόστος θα ισούται με το γινόμενο ενός ποσοστού a επί του συνολικού αθροίσματος του κόστους των συνόλων δεδομένων που χρησιμοποιήθηκαν συν το γινόμενο ενός ποσοστού b επί το συνολικό κόστος εκτέλεσης όλων των προγραμμάτων που χρησιμοποιήθηκαν συν το γινόμενο του μέσου κόστους χρησιμοποίησης κόμβων του διαδικτύου επί το πλήθος των κόμβων δικτύου που χρησιμοποιήθηκαν συν το γινόμενο ενός ποσοστού c επί το συνολικό χρόνο εκτέλεσης της περίπτωσης της ροής εργασίας. Άρα, το κόστος ισούται με το άθροισμα τεσσάρων όρων. Τα ποσοστά a και b θα δίνονται κατά τον ορισμό της ροής εργασίας από τον προμηθευτή ενώ το μέσο κόστος χρησιμοποίησης των κόμβων του διαδικτύου και το ποσοστό c θα είναι μεταβλητές του συστήματος ARION.

Για την εφαρμογή του παραπάνω οικονομικού μοντέλου θα πρέπει να γίνουν αλλαγές στα υποσυστήματα του ARION:

- Στην επιστημονική οντολογία θα πρέπει να προστεθεί η ιδιότητα του κόστους τόσο στην οντότητα του συνόλου δεδομένων όσο και στην οντότητα του επιστημονικού μοντέλου ώστε τα μετα-δεδομένα της Μηχανής Αναζήτησης να περιέχουν πληροφορίες κόστους.

- Θα πρέπει να προστεθούν δύο ιδιότητες στο στοιχείο ‘workflow’ της γλώσσας ορισμού ροών εργασίας. Η μια θα αφορά το προαναφερόμενο ποσοστό a και η άλλη το αντίστοιχο ποσοστό b.
- Ο Κειμενογράφος Ροών Εργασίας θα πρέπει να αναγκάζει τον σχεδιαστή μιας ροής εργασίας να δώσει τιμές στις προηγούμενες ιδιότητες.
- Τα σχήματα wf_class και workflow της Στατιστικής Βάσης Δεδομένων θα πρέπει να αλλάξουν. Στο μεν σχήμα wf_class (για μια ροή εργασίας) θα πρέπει να προστεθούν οι ιδιότητες: του μέσου αθροιστικού κόστους των συνόλων δεδομένων, του μέσου αθροιστικού κόστους των προγραμμάτων, του μέσου αριθμού κόμβων δικτύου που χρησιμοποιήθηκαν κατά την εκτέλεση των περιπτώσεων της ροής εργασίας, του ποσοστού a και του ποσοστού b. Στο δε σχήμα workflow (για μια περίπτωση ροής εργασίας) θα πρέπει να προστεθούν οι αντίστοιχες ιδιότητες: του αθροίσματος του κόστους των συνόλων δεδομένων, του αθροίσματος του κόστους των προγραμμάτων και του πλήθους των κόμβων του διαδικτύου που χρησιμοποιήθηκαν κατά την εκτέλεση αυτής την περίπτωσης ροής εργασίας. Προφανώς, τα δύο σχήματα θα πρέπει να ενημερώνονται με ένα πιο επεκταμένο τρόπο (λόγω των επιπρόσθετων ιδιοτήτων) αυτή τη φορά από τους: Κειμενογράφο Ροών Εργασίας και σύστημα Εκτέλεσης Ροών Εργασίας.
- Όταν κάποιος χρήστης ρωτήσει το σύστημα αν υπάρχει τρόπος να παραχθεί ένα σύνολο δεδομένων μιας κλάσης της επιστημονικής οντολογίας, θα πρέπει να του παρουσιάζεται από τις Γραφικές Διεπαφές Χρήσης μια λίστα από ροές εργασίας που παράγουν αυτό το σύνολο δεδομένων, η οποία και θα περιέχει πληροφορία για το μέσο χρόνο εκτέλεσης της κάθε ροής εργασίας.
- Θα πρέπει να δημιουργηθεί ένα υποσύστημα **Χρέωσης Πελατών** που θα συνεργάζεται με τα υπόλοιπα υποσυστήματα του συστήματος ARION για να χρεώσει ένα πελάτη. Η χρέωση θα γίνεται σε δύο βήματα. Στο πρώτο βήμα, θα παρουσιάζεται στο χρήστη αφενός το κόστος της εκτέλεσης της επιθυμητής περίπτωσης ροής εργασίας όσο αφορά τους δύο τελευταίους όρους (δηλαδή το κόστος της εκτέλεσης που αφορά τη χρήση του συστήματος ARION) και αφετέρου το συνολικό κόστος εκτέλεσης κατά προσέγγιση. Αν ο χρήστης επιθυμεί να συνεχίσει, τότε θα πρέπει να πληρώσει το ποσό του πρώτου κόστους που του παρουσιάστηκε, δίνοντας τα στοιχεία της πιστωτικής του κάρτας μέσω μιας ασφαλούς σύνδεσης στο υποσύστημα **Χρέωσης Πελατών**. Μόλις γίνει αυτό, η εκτέλεση της περίπτωσης ροής εργασίας θα ξεκινήσει. Όταν εν τέλει τελεστεί η εκτέλεση, τότε ο χρήστης θα κληθεί να πληρώσει τη διαφορά του συνολικού πραγματικού κόστους εκτέλεσης από το ποσό που έδωσε αρχικά ώστε να έχει πρόσβαση στο τελικό σύνολο δεδομένων που παράχθηκε από αυτή την εκτέλεση.

5.5.2. Αναθεώρηση της ARION Workflow Language-AWL

Ένα κρίσιμο ανοικτό θέμα είναι αυτό της γλώσσας ορισμού ροών εργασίας. Τόσο η XRL όσο και η AWL είναι γλώσσες που δεν είναι αρκετά διαδεδομένες. Αν και η XRL είναι αρκετά γνωστή, χρησιμεύει περισσότερο στη διαχείριση αρχείων του διαδικτύου. Για αυτό και δεν αποτελεί πρότυπο μιας γλώσσας ορισμού ροών εργασίας. Παρόμοια, η γλώσσα AWL είναι μια αρκετά καινούρια γλώσσα που ακόμη δεν έχει καν δημοσιευθεί σε πολλά επιστημονικά άρθρα και έγγραφα. Συνεπώς, θα ήταν πιο σοφό να χρησιμοποιηθεί μια πιο ευρέως διαδεδομένη γλώσσα όπως είναι η FDL ή μια μετα-γλώσσα όπως είναι η XPSL, αρκεί να ταιριάζει το μοντέλο της με

αυτό της επιστημονικής ροής εργασίας. Ειδικά η χρησιμοποίηση μετα-γλωσσών ενδεικνύεται διότι οι μετα-γλώσσες αποτελούν παγκόσμια πρότυπα και συνεπώς συνεισφέρουν στη διαλειτουργικότητα ροών εργασίας.

Μια διαφορετική προσέγγιση στο θέμα της αλλαγής της γλώσσας ορισμού ροών εργασίας θα ήταν η έκφραση των ροών εργασίας μέσω των οντολογιών. Το μοντέλο της επιστημονικής ροής εργασίας και άρα η ίδια η γλώσσα ορισμού επιστημονικών ροών εργασίας μπορεί να εκφραστεί με βάση το σχήμα μιας οντολογίας. Τόσο οι ορισμοί των ροών εργασίας όσο και οι ορισμοί των περιπτώσεων ροών εργασίας μπορούν να εκφραστούν στο επίπεδο των περιπτώσεων (instance level) – δηλαδή στο επίπεδο των μετα-δεδομένων – των κλάσεων της επιστημονικής ροής εργασίας. Αυτή η προσέγγιση πρέπει να εξεταστεί εξονυχιστικά πριν εφαρμοστεί διότι θα προκαλέσει σημαντικές αλλαγές στο σύστημά μας, που είναι οι ακόλουθες:

- Αλλαγή του Κειμενογράφου Ροών Εργασίας με ένα εξειδικευμένο Κειμενογράφο Οντολογίας (Ontology Editor) για την περιγραφή ροών εργασίας.
- Αποθήκευση των ροών εργασίας και σε μορφή αρχείου και σε βάση δεδομένων (όπως γίνεται με την RDF).
- Ο τρόπος αναζήτησης γίνεται πια μέσω της βάσης δεδομένων και μπορεί να είναι πολύπλοκος.

Η τελευταία αλλαγή είναι και η πιο σημαντική. Αν κατορθώσουμε και περιορίσουμε ακόμη περισσότερο το χρόνο αναζήτησης για ροές εργασίας που παράγουν σύνολα δεδομένων συγκεκριμένου τύπου, τότε ίσως χρειαστεί να ανασχεδιάσουμε το σύστημά μας και να το ξανα-υλοποιήσουμε. Αυτό, όμως, θα πάρει αρκετό χρόνο και μπορεί να μην είναι επιθυμητό από τους χορηγούς και τα συμμετέχοντα μέλη της επιστημονικής μελέτης-εργασίας του ARION.

5.5.3. Διαλειτουργικότητα ροών εργασίας

Όλοι οι ορισμοί ροών εργασίας και περιπτώσεων ροών εργασίας στο σύστημα ARION γίνονται σύμφωνα με τη γλώσσα AWL. Οι ορισμοί της γλώσσας AWL είναι αναγνωρίσιμοι από το σύστημα Εκτέλεσης Ροών Εργασίας, το οποίο και αναλαμβάνει να τους εκτελέσει. Συνεπώς, τόσο η γλώσσα AWL όσο και το σύστημα Εκτέλεσης Ροών Εργασίας συνεισφέρουν στην διαλειτουργικότητα ροών εργασίας. Τι θα γίνει, όμως, όταν ένα σύστημα διαχείρισης ροών εργασίας X με διαφορετική γλώσσα ορισμού ροών εργασίας Y θέλει να συνεργαστεί με το σύστημα ARION; Τότε υπάρχουν δύο τρόποι λύσης του προβλήματος. Ο πρώτος τρόπος είναι με τη μετατροπή του κομματιού της ροής εργασίας που θα εκτελεστεί από το σύστημα X και είναι εκφρασμένο στη γλώσσα AWL μέσω ενός mapper στη γλώσσα ορισμού ροών εργασίας Y. Αυτός ο mapper αντιστοιχεί δομικά μέρη της μιας γλώσσας σε αντίστοιχα της άλλης. Ο δεύτερος τρόπος, που είναι πιο γενικός, προϋποθέτει την έκφραση του επίμαχου κομματιού της ροής εργασίας μέσω ενός mapper σε ένα κομμάτι εκφρασμένο σε μια πρότυπη και ευρέως γνωστή μετα-γλώσσα ορισμού ροών εργασίας. Το νέο κομμάτι θα αποστέλλεται στο σύστημα X, όπου θα υπάρχει ένας άλλος mapper, ο οποίος και θα μετατρέπει το νέο κομμάτι σε ένα τελικό κομμάτι εκφρασμένο στη γλώσσα Y. Επειδή το κομμάτι αυτό του ορισμού μιας (περίπτωσης) ροής εργασίας είναι συγκεκριμένο, θα πρέπει η γλώσσα AWL να επεκταθεί μέσω ενός νέου δομικού στοιχείου που θα περικλείει το εν λόγω κομμάτι. Επιπλέον, πριν την εκτέλεση του συγκεκριμένου ορισμού ροής εργασίας, το εργαλείο μετατροπής

(mapper) θα μετατρέπει το επίμαχο κομμάτι στην γλώσσα Y ή στη μετα-γλώσσα, θα το αποθηκεύει σε ένα αρχείο και θα το αντικαθιστά μέσα στον ορισμό με ένα δομικό μέρος-στοιχείο τύπου 'task'. Αυτό το στοιχείο 'task' θα παίρνει ως είσοδο το αποθηκευμένο αρχείο και θα έχει ως έξοδο όλα τα τελικά σύνολα δεδομένων που θα έπρεπε να είχαν παραχθεί από το επίμαχο κομμάτι της ροής εργασίας. Επομένως, το σύστημα Εκτέλεσης Ροών Εργασίας, στο οποίο θα αποστέλλεται ο τροποποιημένος ορισμός της ροής εργασίας προς εκτέλεση, θα βλέπει το σύστημα X σαν ένα πρόγραμμα που θα παίρνει ως είσοδο το αρχείο του μεταφρασμένου επίμαχου κομματιού και θα έχει ως έξοδο μια σειρά από σύνολα δεδομένων.

5.5.4. Χρησιμοποίηση μιας XML Βάσης Δεδομένων

Η Αποθήκη Ροών Εργασίας είναι μια έννοια που αντιπροσωπεύεται από έναν κατάλογο με όνομα "workflows" σε ένα λειτουργικό σύστημα. Κάθε ροή εργασίας αποθηκεύεται σε ένα υποκατάλογο αυτού του καταλόγου. Από την άλλη πλευρά, οι επερωτήσεις στην Αποθήκη Ροών Εργασίας εκτελούνται με τον εξής τρόπο: Εξετάζεται κάθε υποκατάλογος, για ορισμούς ροών εργασίας που παράγουν σύνολα δεδομένων συγκεκριμένου τύπου, με το άνοιγμα όλων των αρχείων του υποκαταλόγου και με την εκτέλεση Xpath επερωτήσεων.

Η τεχνολογία της XML έχει προχωρήσει αρκετά ώστε να έχει προσεγγίσει το στάδιο της δημιουργίας σταθερών XML Βάσεων Δεδομένων, όπως είναι αυτή της Xindice [92]. Οπότε είναι απορίας άξιο αν θα έπρεπε να χρησιμοποιήσουμε μια XML Βάση Δεδομένων για να υλοποιήσουμε την έννοια της Αποθήκης Ροών Εργασίας. Η απάντηση στην προηγούμενη απορία δεν δύναται να είμαι άμεση. Θα πρέπει να εξεταστεί αφενός πόσο μεγάλο αποθηκευτικό χώρο χρειάζεται μια XML Βάση Δεδομένων για να λειτουργήσει και αφετέρου το ποσοστό του χρόνου της CPU που σπαταλιέται για την διαχείρισή της. Επίσης, θα πρέπει να εξεταστεί πόσο γρήγορα αποθηκεύονται τα XML αρχεία στη βάση και πως αντιμετωπίζεται ο χειρισμός των ταυτόχρονων αιτήσεων αποθήκευσης ή επερωτήσεων. Επιπλέον, θα πρέπει να εξεταστεί η κλιμάκωση της XML Βάσης Δεδομένων αφενός ως προς το πλήθος των αρχείων που μπορούν να αποθηκευθούν σε αυτή και αφετέρου ως προς το πλήθος των αιτήσεων που μπορούν να διατυπωθούν σε αυτή. Όμως, το πιο σημαντικό θέμα προς εξέταση είναι ο χρόνος απόκρισης της XML Βάσης Δεδομένων ως προς την κάθε επερωτήση για ροές εργασίας που παράγουν σύνολα δεδομένων συγκεκριμένου τύπου. Η γνώμη μας είναι ότι εφόσον οι απαντήσεις στα παραπάνω ζητήματα είναι θετικές, τότε η Αποθήκη Ροών Εργασίας πρέπει να υλοποιηθεί από μια XML Βάση Δεδομένων. Αν συμβεί το τελευταίο, τότε η νέα υλοποίηση θα παρουσιάζει τρία σημαντικά πλεονεκτήματα:

1. Θα μειωθεί ο χρόνος απόκρισης των επερωτήσεων.
2. Οι αλλαγές που γίνονται σε ήδη αποθηκευμένους ορισμούς συνήθως είναι πολύ μικρές. Οπότε σε κάθε αλλαγή δεν θα χρειάζεται να στέλνεται όλο το αρχείο προς αποθήκευση. Μόνο οι αλλαγές σε μορφή XML-DB:XUupdate [93] χρειάζονται να σταλούν προς αποθήκευση. Συνεπώς, αφού στέλνονται λιγότερα σε μήκος μηνύματα, θα είναι μειωμένη η κίνηση προς την XML Βάση Δεδομένων. Επιπλέον, η λειτουργία της τροποποίησης ορισμού ροής εργασίας θα επιτελείται πιο σύντομα. Σημειωτέον ότι κάθε ανανέωση-αλλαγή τύπου XML-DB:XUupdate μπορεί να αφορά πολλά XML έγγραφα.
3. Τόσο η Αποθήκη Ροών Εργασίας όσο και η Στατιστική Βάση Δεδομένων μπορούν να ενοποιηθούν σε μια μόνο XML Βάση Δεδομένων. Αρκεί να

επιτελεστούν δύο ενέργειες. Η πρώτη ενέργεια είναι τα πεδία των σχημάτων 'wf_class' και 'workflow' να προστεθούν στο δομικό στοιχείο 'workflow' της γλώσσας ορισμού ροών εργασίας με τη μορφή ιδιοτήτων. Η δεύτερη ενέργεια είναι οι εισαγωγές, οι ανανεώσεις και οι επερωτήσεις που διεκπεραιώνονταν στη Στατιστική Βάση Δεδομένων να μετατραπούν σε XML-DB:XUpdate εντολές και σε Xpath [94] επερωτήσεις. Εν συνεχεία, η ενοποίηση των δύο προαναφερόμενων συστατικών μερών απλοποιεί την αρχιτεκτονική του συστήματός μας.

5.5.5. Επεκτάσεις στις Επερωτήσεις

Στο paper 'Querying and Splicing of XML Workflows' [95] περιγράφονται τρόποι πρόσβασης στην πληροφορία που ενθυλακώνεται στους ορισμούς (περιπτώσεων) ροών εργασίας, οι οποίοι και εκφράζονται μέσω της γλώσσας XML. Ορίζονται τρία είδη επερωτήσεων για την πρόσβαση σε αυτού του είδους την πληροφορία:

1. *Απλές*: Αυτό το είδος επερωτήσεων ελέγχει την κατάσταση μιας περίπτωσης ροής εργασίας. Παραδείγματα απλών επερωτήσεων είναι:
 - a. Ποιά είναι η κατάσταση ενός έργου.
 - b. Ποιά είναι η λίστα που περιέχει όλα τα έργα με συγκεκριμένο είδος κατάστασης.
 - c. Ποιά έργα επεξεργάζονται αυτή τη δεδομένη στιγμή ένα αντικείμενο δεδομένου.
 - d. Πόσες φορές εκτελέστηκε ένα δομικό στοιχείο ανακύκλωσης για μια περίπτωση ροής εργασίας.
2. *Επερωτήσεις Διάσχισης (Traversal queries)*: Αυτό το είδος επερωτήσεων ελέγχει ιδιότητες που είναι πιο γενικές (global) για μια ροή εργασίας ή περίπτωση ροής εργασίας. Περιλαμβάνει επερωτήσεις που αφορούν τη σχέση μεταξύ έργων (parallel, sequential, exclusive), τον προσδιορισμό των έργων που θα εκτελεστούν σίγουρα ή μπορεί να εκτελεστούν για μια περίπτωση ροής εργασίας (ή ένα μέρος αυτής της περίπτωσης) και τέλος, τον αναμενόμενο χρόνο εκτέλεσης μιας (περίπτωσης) ροής εργασίας.
3. *Επερωτήσεις Κατασκευής Σχήματος*: Αυτό το είδος επερωτήσεων μπορεί να χρησιμοποιηθεί για την κατασκευή σχήματος ροής εργασίας, με την αξιοποίηση συστατικών μερών από μια βιβλιοθήκη από 'templates' και 'base templates'.

Οι άνωθεν επερωτήσεις παρέχουν μια βάση για τη βελτίωση της σχεδίασης και της αποδοτικότητας των ροών εργασίας, τόσο κατά το χρόνο compile όσο και κατά το χρόνο εκτέλεσης. Για παράδειγμα, βοηθάνε αφενός στην εύρεση ενδεχόμενων bottlenecks και συνθηκών ανταγωνισμού, κατά τη χρήση δεδομένων, και αφετέρου στη ανίχνευση και αποφυγή ενδεχόμενων καθυστερήσεων, κατά το χρόνο εκτέλεσης. Η παραπάνω λειτουργικότητα είναι απαραίτητη για τον προσδιορισμό, την εκτέλεση και την επίβλεψη ηλεκτρονικών υπηρεσιών στο περιβάλλον των διάφορων εφαρμογών ηλεκτρονικού εμπορίου.

Σε αυτό το paper εισάγετε ένα απλό, αλλά ικανοποιητικά εκφραστικό, μοντέλο ροών εργασίας με όνομα Workflow Query Model – WQM, το οποίο χρησιμοποιεί δομικά στοιχεία τύπου flowchart που είναι 'σωστά φωλιασμένα' ('properly nested'). Επομένως, το μοντέλο WQM είναι ιδιαίτερα βολικό για τη διατύπωση επερωτήσεων μέσω των γλωσσών επερωτήσεως XML (XML query languages). Επιλέχθηκε η XML Query Algebra, διότι προσφέρει τόσο δομική

επανάληψη (structural recursion) όσο και συναρτήσεις οριζόμενες από χρήστες για τη διάσχιση σχημάτων ροών εργασίας. Τα παραπάνω χαρακτηριστικά είναι απαραίτητα για την έκφραση των προαναφερθέντων τριών ειδών επερώτησης.

Δυστυχώς, οι τεχνικές που αναφέρονται στο εν λόγω paper δεν μπορούν να εφαρμοστούν στο πλαίσιο των μοντέλων ροών εργασίας που βασίζονται είτε στα flowcharts και δεν είναι 'σωστά φωλιασμένα' είτε στα Petri Nets είτε στα διαγράμματα καταστάσεων. Συνεπώς, μια μελλοντική επέκταση-εργασία θα αποτελούσε η εφαρμογή των παραπάνω τεχνικών στο μοντέλο ροών εργασίας της AWL, δηλαδή σε ένα μοντέλο ροής εργασίας που βασίζεται στα Petri Nets. Μάλιστα, επειδή αφενός η RDF υποστηρίζει τη δημιουργία απλών και πολύπλοκων τύπων (αυτό το χαρακτηριστικό απαιτείται για να έκφραση των τύπων τιμής των παραμέτρων ενός προγράμματος στην AWL) και αφετέρου η RQL υποστηρίζει τα χαρακτηριστικά της XML Query Algebra, θα ήταν ενδιαφέρον αν η AWL εκφραζόταν μέσω ενός RDF Schema. Η τελευταία επέκταση θα προσέφερε τα εξής πλεονεκτήματα:

- Έκφραση των σχημάτων (περιπτώσεων) ροών εργασίας σε RDF (XML-based).
- Αποθήκευση των σχημάτων ροών εργασίας και του μοντέλου της AWL στην RSSDB.
- Τη διατύπωση των παλαιών και νέων ειδών επερώτησης μέσω της RQL. Μειώνεται ο χρόνος απόκρισης των επερωτήσεων.
- Την απλοποίηση της αρχιτεκτονικής του συστήματος **K.A.P.E.E.A**, η οποία τώρα θα μοιάζει περίπου με εκείνη της Μηχανής Αναζήτησης, μια και θα χρησιμοποιεί πολλά από τα συστατικά της μέρη.

Η τελευταία επέκταση δύναται να αποτελέσει την τελική και θεμελιώδη λύση στο σύστημά μας.

5.5.6. Επεκτάσεις στους Κειμενογράφους

Ο Κειμενογράφος Ροών Εργασίας μπορεί να επεκταθεί με τη χρησιμοποίηση μιας διαφορετικής-εναλλακτικής όψης (view) για μια ροή εργασίας, σε σχέση με τη δενδρική, που να βασίζεται σε γράφους. Ο χρήστης θα έχει την ευχέρεια να αλλάζει όψεις κατά τον ορισμό της ροής εργασίας ανάλογα με τις διαθέσεις του. Σε κάθε όψη θα είναι διαφορετικός ο τρόπος που θα τροποποιείται η δομή της ροής εργασίας. Όμως, κάθε τροποποίηση στη δομή της ροής εργασίας θα έχει αντίκτυπο και στις δύο όψεις. Μια επιπλέον επέκταση, για τον εν λόγω κειμενογράφο, είναι η υποστήριξη όλων των δομικών στοιχείων της AWL, μια και υποστηρίζεται ένα μόνο υποσύνολο αυτών.

Η προσθήκη μιας νέας όψης πρέπει να πραγματοποιηθεί και στον Κειμενογράφο Περιπτώσεων Ροών Εργασίας. Επίσης, ο εν λόγω κειμενογράφος πρέπει να ενισχυθεί με νέες λειτουργίες όπως αυτή της τροποποίησης του ορισμού περίπτωσης ροής εργασίας. Ένας χρήστης μπορεί να θέλει να ξανατρέξει μια περίπτωση ροής εργασίας ή να τροποποιήσει ένα μικρό σημείο του ορισμού της ή να την τρέξει τώρα στο background. Οπότε θα πρέπει να του δίνεται η δυνατότητα να 'φορτώνει' ορισμούς περιπτώσεων ροών εργασίας και να τους τροποποιεί.

5.6. Συμπεράσματα

Το σύστημα **Κ.Α.Ρ.Ε.Ε.Α** αποτελεί κατανεμημένο σύστημα αποθήκευσης ροών εργασίας. Οι παρεχόμενες λειτουργίες του είναι: ο ορισμός, η αποθήκευση και η επερώτηση σε (περιπτώσεις) ροές εργασίας. Επιπλέον, παρέχει τις λειτουργίες της αποθήκευσης και επερώτησης στατιστικής πληροφορίας, η οποία και προκύπτει από την εκτέλεση περίπτωσης ροής εργασίας. Είναι ένα σύστημα αξιόπιστο και ασφαλές, το οποίο υποστηρίζει τόσο τον ταυτοχρονισμό χρηστών όσο και την κατανομή του φόρτου παροχής λειτουργιών. Επιπροσθέτως, το σύστημα **Κ.Α.Ρ.Ε.Ε.Α** κρίνεται ανοικτό, όσον αφορά την προσθήκη νέων αντικειμένων και λειτουργιών. Συνεπώς, αποτελεί πρωτότυπο και πρότυπο σύστημα, του οποίου η χρήση συντελείται σε δύο ανεξάρτητα επίπεδα. Στο πρώτο επίπεδο, δύναται να χρησιμοποιηθεί ως ένα αυτοτελές σύστημα, το οποίο και θα προσφέρει τις λειτουργίες του σε άλλες υπολογιστικές εφαρμογές ή συστήματα. Στο δεύτερο επίπεδο, καθίσταται εφικτή η χρήση του ως ένα ήδη υλοποιημένο τμήμα ενός υπό κατασκευή, αυτοτελούς συστήματος διαχείρισης ροών εργασίας.

Παραρτήματα

A. To Document Type Definition της γλώσσας XRL

The data type definition (DTD) file for the workflow specification

```
<!ENTITY % routing_element
"task|sequence|any_sequence|choice|condition|parallel_sync|
parallel_no_sync|parallel_part_sync|wait_all|wait_any|while_do|
stop|terminate">

<!ELEMENT route (%routing_element);>
<!ATTLIST route name ID #REQUIRED
created_by CDATA #IMPLIED
date CDATA #IMPLIED>
<!ELEMENT task (event*)>
<!ATTLIST task name ID #REQUIRED
address CDATA #REQUIRED
doc_read NMTOKENS #IMPLIED
doc_update NMTOKENS #IMPLIED
doc_create NMTOKENS #IMPLIED
result CDATA #IMPLIED
status (ready|running|enabled|disabled|aborted) #IMPLIED
start_time NMTOKEN #IMPLIED
end_time NMTOKEN #IMPLIED
notify CDATA #IMPLIED>
<!ELEMENT event EMPTY>
<!ATTLIST event name ID #REQUIRED>
<!ELEMENT sequence ((%routing_element;|state)+)>
<!ELEMENT any_sequence ((%routing_element;)+)>
<!ELEMENT choice ((%routing_element;)+)>
<!ELEMENT condition (true|false)*>
<!ATTLIST condition condition CDATA #REQUIRED>
<!ELEMENT true (%routing_element;)>
<!ELEMENT false (%routing_element;)>
<!ELEMENT parallel_sync ((%routing_element;)+)>
<!ELEMENT parallel_no_sync ((%routing_element;)+)>
<!ELEMENT parallel_part_sync ((%routing_element;)+)>
<!ATTLIST parallel_part_sync number NMTOKEN #REQUIRED>
<!ELEMENT wait_all (event_ref|timeout)+>
<!ELEMENT wait_any (event_ref|timeout)+>
<!ELEMENT event_ref EMPTY>
<!ATTLIST event_ref name IDREF #REQUIRED>
<!ELEMENT timeout (%routing_element;)*>
<!ATTLIST timeout time CDATA #REQUIRED
type (relative|s_relative|absolute) "absolute">
<!ELEMENT while_do (%routing_element;)>
```

```
<!ATTLIST while_do condition CDATA #REQUIRED>  
<!ELEMENT stop EMPTY>  
<!ELEMENT terminate EMPTY>  
<!ELEMENT state (event+)>
```


B. To Document Type Definition της γλώσσας AWL

```
<!ENTITY % workflow_element
"task|sequence|any_sequence|choice|condition|back_track|parallel_sync|
parallel_no_sync|parallel_part_sync|wait_all|wait_any|while_do|
stop|terminate">
<!ELEMENT workflow (%workflow_element;)>
<!ATTLIST workflow
    ID ID #REQUIRED
    name CDATA #REQUIRED
    created_by CDATA #IMPLIED
    date CDATA #IMPLIED
    last_modified CDATA #IMPLIED
    description CDATA #IMPLIED
    roles CDATA #IMPLIED
    rights CDATA #IMPLIED
>
<!ELEMENT task (event*, input, program, output)>
<!ATTLIST task
    ID ID #REQUIRED
    AgencyPort CDATA #IMPLIED
    result CDATA #IMPLIED
    status (ready | running | enabled | disabled | aborted) #IMPLIED
    start_time NMTOKEN #IMPLIED
    end_time NMTOKEN #IMPLIED
    max_run_time NMTOKEN #IMPLIED
    notify CDATA #IMPLIED
>
<!ELEMENT event EMPTY>
<!ATTLIST event
    name ID #REQUIRED
>
<!ELEMENT input (data-set*, parameter*)>
<!ELEMENT program (#PCDATA)>
<!ATTLIST program
    name CDATA #REQUIRED
    ID ID #REQUIRED
    ref IDREF #IMPLIED
    hostname CDATA #REQUIRED
    URI CDATA #REQUIRED
    sync (Y | N) #REQUIRED
    heavy (Y | N) #REQUIRED
    mobile (Y | N) #REQUIRED
    simulation (Y | N) #IMPLIED
    visualization (Y | N) #IMPLIED
    statistical (Y | N) #IMPLIED
    validation (Y | N) #IMPLIED
    database_retrieval (Y | N) #IMPLIED
    output_filter (Y | N) #IMPLIED
```

```

        transformation (Y | N) #IMPLIED
    >
    <!ELEMENT output (data-set+, parameter*)>
    <!ELEMENT data-set (#PCDATA)>
    <!ATTLIST data-set
        ID ID #REQUIRED
        ref IDREF #IMPLIED
        name CDATA #REQUIRED
        URI CDATA #IMPLIED
        hostname CDATA #IMPLIED
        type (general | specific | produced | produced-storable) #REQUIRED
        ontology_name CDATA #REQUIRED
        ontology_uri CDATA #REQUIRED
        classname CDATA #REQUIRED
    >
    <!ELEMENT parameter (#PCDATA)>
    <!ATTLIST parameter
        name CDATA #REQUIRED
        type (integer | double | string | boolean | no-value | special) #REQUIRED
        syntax CDATA #REQUIRED
        default_value CDATA #IMPLIED
        description CDATA #IMPLIED
        optional (true | false) #REQUIRED
        type_of_value_constraint (free | range | discrete) #REQUIRED
        value_constraint CDATA #IMPLIED
    >
    <!ELEMENT sequence ((%workflow_element; | state)+)>
    <!ELEMENT any_sequence ((%workflow_element;)+)>
    <!ELEMENT back_track ((%workflow_element;)+)>
    <!ELEMENT choice ((%workflow_element;)+)>
    <!ELEMENT condition (true | false)*>
    <!ATTLIST condition
        condition CDATA #REQUIRED
    >
    <!ELEMENT true (%workflow_element;)>
    <!ELEMENT false (%workflow_element;)>
    <!ELEMENT parallel_sync ((%workflow_element;)+)>
    <!ELEMENT parallel_no_sync ((%workflow_element;)+)>
    <!ELEMENT parallel_part_sync ((%workflow_element;)+)>
    <!ATTLIST parallel_part_sync
        number NMTOKEN #REQUIRED
    >
    <!ELEMENT wait_all (event_ref | timeout)+>
    <!ELEMENT wait_any (event_ref | timeout)+>
    <!ELEMENT event_ref EMPTY>
    <!ATTLIST event_ref
        name IDREF #REQUIRED
    >
    <!ELEMENT timeout (%workflow_element;)*>
    <!ATTLIST timeout

```

```
    time CDATA #REQUIRED
    type (relative | s_relative | absolute) "absolute"
>
<!ELEMENT while_do (%workflow_element;)>
<!ATTLIST while_do
    condition CDATA #REQUIRED
>
<!ELEMENT terminate EMPTY>
<!ELEMENT state (event+)>
<!ELEMENT stop EMPTY>
```

Γ. Περιγραφή Κειμενογράφου (Περιπτώσεων) Ροών Εργασίας

Γ.Ι. Εισαγωγή

Σε αυτήν την ενότητα του παραρτήματος θα ασχοληθούμε τόσο με τη περιγραφή του Κειμενογράφου Ροών Εργασίας όσο και με την περιγραφή του Κειμενογράφου Περιπτώσεων Ροών Εργασίας. Η περιγραφή θα αφορά αφενός την λειτουργικότητα των κειμενογράφων, η οποία και παρέχεται στους χρήστες του συστήματος ARION, και αφετέρου τη γραφική διεπαφή χρήσης (δηλαδή τους γραφικούς διαλόγους/φόρμες) αυτών των κειμενογράφων. Επιπλέον, θα αναφερθεί παράδειγμα ορισμού μιας ροής εργασίας και περίπτωσης αυτής, το οποίο και θα χαρακτηρίζεται από συγκεκριμένα και ακριβή βήματα αλληλεπίδρασης του χρήστη με τους κειμενογράφους. Στόχος μας είναι να εξοικειώσουμε το χρήστη του ARION στη χρήση των εν λόγω κειμενογράφων.

Γ.ΙΙ. Ο Κειμενογράφος Ροών Εργασίας

Ο Κειμενογράφος Ροών Εργασίας είναι ένα γραφικό εργαλείο ορισμού είτε ροών εργασίας είτε περιπτώσεων ροών εργασίας, και χρησιμοποιείται από εμπειρογνώμονες χρήστες του συστήματος ARION. Βασική προϋπόθεση για την εκμετάλλευση του εν λόγω κειμενογράφου είναι τόσο η γνώση του μοντέλου επιστημονικής ροής εργασίας όσο και η εξοικείωση με τη γλώσσα AWL. Ο εν λόγω κειμενογράφος μπορεί να χρησιμοποιηθεί είτε ως off-line εργαλείο είτε ως ένα γραφικό εργαλείο βασισμένο στο διαδίκτυο (web-based graphical tool). Η υλοποίηση έγινε μέσω της οντο-κεντρικής γλώσσας προγραμματισμού Java και με τη βοήθεια του πακέτου 'dom4j', ενός πακέτου βασικού χειρισμού XML εγγράφων μέσω της Java.

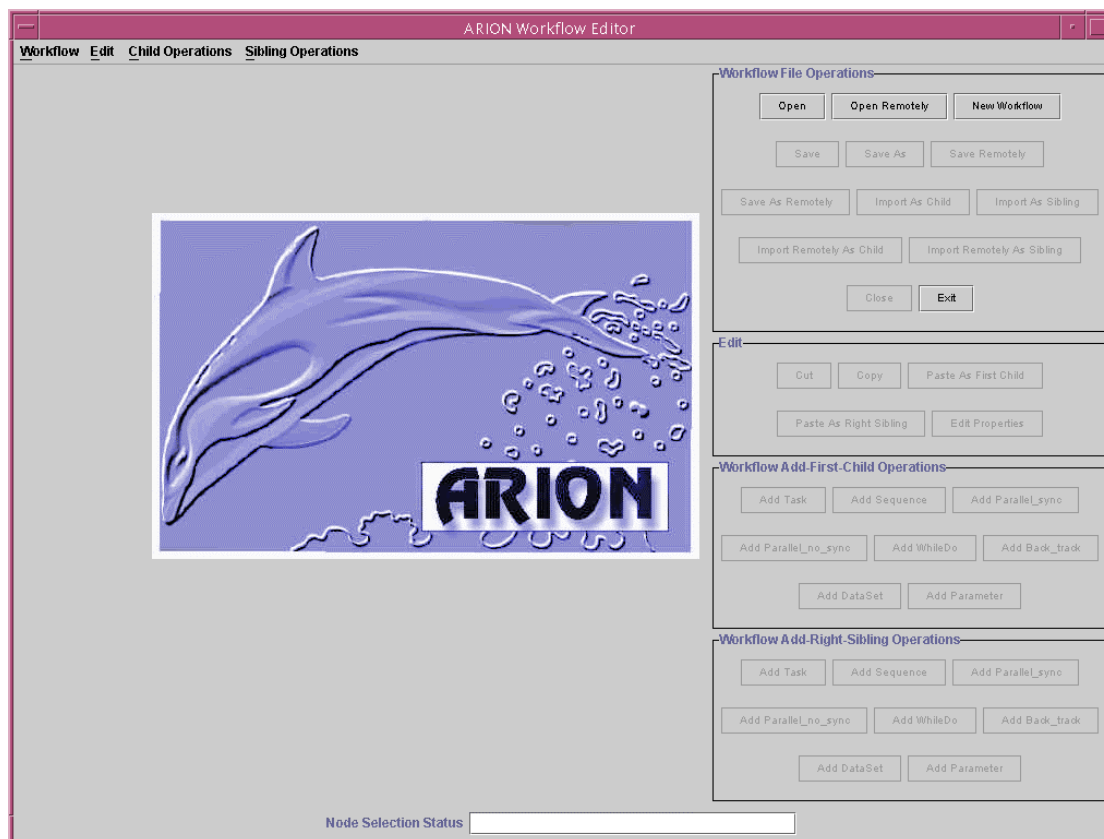
Βασικές Λειτουργίες

Ο Κειμενογράφος Ροών Εργασίας συνδέεται με σχεδόν όλα τα συστατικά μέρη του συστήματος K.A.P.E.E.A, ώστε να διεκπεραιώσει το σύνολο των λειτουργιών που του ανατίθενται, είτε άμεσα είτε έμμεσα. Οι βασικότερες λειτουργίες είναι οι ακόλουθες :

- Ταυτοποίηση εμπειρογνώμονος χρήστη. Μια διαδικασία-λειτουργία που πραγματοποιείται πριν τη 'φόρτωση' του κειμενογράφου, ώστε αυτός να χρησιμοποιηθεί από έγκυρους χρήστες του συστήματος ARION.
- 'Άνοιγμα' προσδιορισμού ροής εργασίας. Πραγματοποιείται όταν ο χρήστης θέλει να αλλάξει τον ορισμό μιας ροής εργασίας.
- 'Απομακρυσμένο άνοιγμα' προσδιορισμού ροής εργασίας. Αυτό συμβαίνει σε περιπτώσεις όπου ο χρήστης θέλει να αλλάξει ένα προσδιορισμό ροής εργασίας, που δεν βρίσκεται αποθηκευμένος στον υπολογιστή του. Απαραίτητη προϋπόθεση για να πραγματοποιηθεί η εν λόγω λειτουργία είναι ο χρήστης να έχει εγκαταστήσει και εκκινήσει ένα FileServer στον απομακρυσμένο υπολογιστή, στον οποίο έχει αποθηκευθεί ο ζητούμενος προσδιορισμός.

- Αποθήκευση προσδιορισμού ροής εργασίας. Ο χρήστης με αυτήν τη λειτουργία αποθηκεύει τον εν λόγω προσδιορισμό στο σύστημα **K.A.P.E.E.A.**
- 'Απομακρυσμένη αποθήκευση' προσδιορισμού ροής εργασίας. Με αυτή τη λειτουργία ο χρήστης αποθηκεύει ένα προσδιορισμό όχι μόνο στο σύστημα K.A.P.E.E.A, αλλά και σε έναν απομακρυσμένο υπολογιστή. Στον απομακρυσμένο υπολογιστή θα πρέπει να εκτελείται ένας FileServer.
- Δημιουργία νέου προσδιορισμού ροής εργασίας. Με αυτή τη λειτουργία δημιουργείται ένας προσωρινός προσδιορισμός ροής εργασίας. Όταν αυτός αποθηκευθεί, θα πάψει να είναι προσωρινός.
- 'Εισαγωγή και ένωση' προσδιορισμού με κομμάτι του τρέχοντος προσδιορισμού ροής εργασίας. Αυτή η λειτουργία είναι από τις πιο σημαντικές. Πρώτον, διότι βοηθάει στη δημιουργία δια-επιχειρησιακών (inter-organizational) προσδιορισμών ροών εργασίας μέσω της ενοποίησης ροών εργασίας που ανήκουν σε προμηθευτές του συστήματος ARION. Δεύτερον, διότι παρακινεί τους εμπειρογνώμων χρήστες να ορίσουν κομμάτια προσδιορισμών ροών εργασίας (workflow specification templates), τα οποία επαναλαμβάνονται σε αρκετούς προσδιορισμούς μεγάλων ροών εργασίας. Με αυτόν το τρόπο, όταν οι χρήστες επεξεργάζονται ένα προσδιορισμό μεγάλης ροής εργασίας, απλά θα εισάγουν το κομμάτι προσδιορισμού στο σωστό σημείο και θα αποφύγουν την επιπλέον επεξεργασία του ορισμού του εν λόγω κομματιού.

Υπάρχουν και άλλες βασικές λειτουργίες του Κειμενογράφου Ροών Εργασίας, οι οποίες, όμως, αποδίδονται ως λειτουργίες του Κειμενογράφου Περιπτώσεων Ροών Εργασίας.

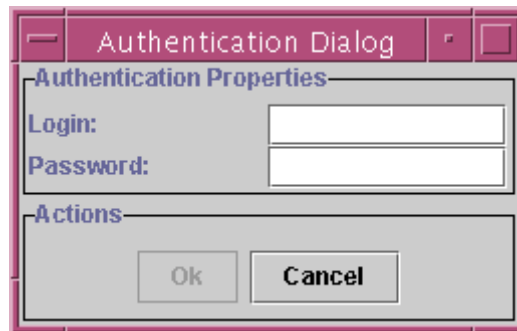


Εικόνα 23 - Ο Κειμενογράφος Ροών Εργασίας

Η Γραφική Διεπαφή Χρήσης του Κειμενογράφου

Ο Κειμενογράφος ροών εργασίας φαίνεται στην εικόνα 23.

Η πρώτη διεπαφή που παρουσιάζεται στον χρήστη όταν ‘ανοίγει’ τον Κειμενογράφο Ροών Εργασίας είναι η Φόρμα Ταυτοποίησης, η οποία παρουσιάζεται στην εικόνα 24. Αυτή αποτελείται από δύο πεδία που ονομάζονται “login” και “password” και από δύο κουμπιά με ονόματα: “Ok” και “Cancel”. Για να ταυτοποιηθεί, ο χρήστης πρέπει να συμπληρώσει τα στοιχεία του γεμίζοντας τα προαναφερόμενα πεδία και έπειτα να πατήσει το κουμπί “Ok”. Αν τα στοιχεία του χρήστη είναι λανθασμένα, τότε τα πεδία καθαρίζονται και πρέπει να ξαναεπιναληφθεί η προηγούμενη διαδικασία.



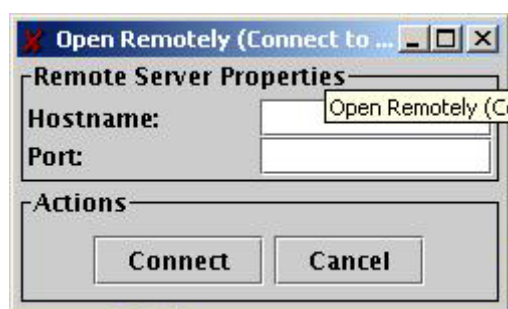
Εικόνα 24 - Η φόρμα Ταυτοποίησης Χρήστη

Μόλις η διαδικασία ταυτοποίησης επιτύχει, το κύριο πλαίσιο-παράθυρο του κειμενογράφου εμφανίζεται. Αυτό το πλαίσιο αποτελείται από: μια μπάρα από μενού, ένα γραφικό δένδρο, ένα πάνελ από ομαδοποιημένα κουμπιά και μια μπάρα καταστάσεων. Η μπάρα από μενού προσφέρει τόσο την ίδια ομαδοποίηση στα μενού όσο και την ίδια λειτουργικότητα (ως προς τα στοιχεία των μενού – menu items) με αυτή του πάνελ των ομαδοποιημένων κουμπιών. Οι ομάδες των κουμπιών στο πάνελ είναι οι εξής: ‘Λειτουργίες Αρχείων Ροών Εργασίας (Workflow File Operations)’, ‘Λειτουργίες Τροποποίησης (Edit)’, ‘Λειτουργίες Προσθήκης Πρώτου Παιδιού Σε Στοιχείο Ροής Εργασίας (Workflow Add First-Child Operations)’ και ‘Λειτουργίες Προσθήκης Επόμενου Αδελφικού Κόμβου Σε Στοιχείο Ροής Εργασίας (Workflow Add Right-Sibling Operations)’. Το γραφικό δένδρο αναπαριστά τη ροή του ελέγχου της υπό επεξεργασία ροής εργασίας αποδίδοντας μια ιεραρχική δενδρική δομή των στοιχείων της ροής εργασίας, τα οποία και απαρτίζουν τη εν λόγω ροή ελέγχου. Η μπάρα κατάστασης ενημερώνει το χρήστη για το πιο στοιχείο της ροής εργασίας-κόμβος του δένδρου έχει διαλεχτεί/σημαδευτεί από το ποντίκι του υπολογιστή.

Η μπάρα των μενού αποτελείται από τέσσερα μενού, τα οποία ονομάζονται: ‘Ροή Εργασίας (Workflow)’, ‘Τροποποίηση (Edit)’, ‘Λειτουργίες Προσθήκης Παιδιού Κόμβου (Add Child Operations)’ και ‘Λειτουργίες Προσθήκης Αδελφικού Κόμβου (Add Sibling Operations)’. Το μενού ‘Workflow’ αποτελείται από τα επόμενα δώδεκα στοιχεία μενού: ‘Open’, ‘Open Remotely’, ‘New’, ‘Save’, ‘Save As’, ‘Save Remotely’, ‘Save As Remotely’, ‘Execute’, ‘Import As First-Child’, ‘Import As Right-Sibling’, ‘Close’, ‘Exit’. Το μενού ‘Edit’ αποτελείται από τα ακόλουθα πέντε στοιχεία μενού: ‘Cut’, ‘Copy’, ‘Paste as First Child’, ‘Paste As Right Sibling’, ‘Edit Properties’. Τόσο το μενού ‘Add First-Child Operations’ όσο και το μενού ‘Add Next-Sibling Operations’ αποτελούνται από τα εξής οκτώ στοιχεία

μενού: ‘Add a Task Element’, ‘Add a Sequence Element’, ‘Add a Parallel_Sync Element’, ‘Add a While-Do Element’, ‘Add a BackTrack Element’, ‘Add a Data Set Element’ and ‘Add a Parameter Element’. Στις επόμενες παραγράφους, θα περιγράψουμε την λειτουργικότητα που εσωκλείει κάθε ένα από αυτά τα στοιχεία μενού.

Το στοιχείο μενού ‘Open’ χρησιμοποιείται για το άνοιγμα ενός προσδιορισμού ροής εργασίας. Όταν το στοιχείο αυτό επιλεγθεί, τότε ‘εκτοξεύεται’ μια φόρμα με όνομα ‘Open’, όπως φαίνεται και στην εικόνα . Έπακολούθως, ο χρήστης μπορεί να επιλέξει ένα συγκεκριμένο αρχείο από μια λίστα αρχείων της εν λόγω φόρμας είτε με το να πατήσει το δεξί κουμπί του ποντικιού είτε πάνω στο όνομα του αρχείου είτε πάνω στο εικονίδιό του. Εναλλακτικά, μπορεί να γράψει το όνομα του επιθυμητού αρχείου σε ένα πεδίο με όνομα ‘File Name’. Αν το όνομα του συγκεκριμένου αρχείου δεν έχει κατάληξη ‘.xml’, τότε ο χρήστης πρέπει να αλλάξει το γραφικό στοιχείο ‘Files of Type’ από την τιμή ‘Just XML Files’ στην τιμή ‘All Files’. Τελικώς, ο χρήστης πρέπει να πατήσει το κουμπί ‘Open’. Μόλις το κάνει αυτό, το γραφικό δένδρο του προσδιορισμού της ροής εργασίας εμφανίζεται στα αριστερά του κύριου πλαισίου του κειμενογράφου. Το στοιχείο μενού ‘Open’ και το αντίστοιχο κουμπί ‘Open’ απενεργοποιούνται αν ένα προσδιορισμός ροής εργασίας έχει ήδη ‘εκτοξευθεί’ στον κειμενογράφο.

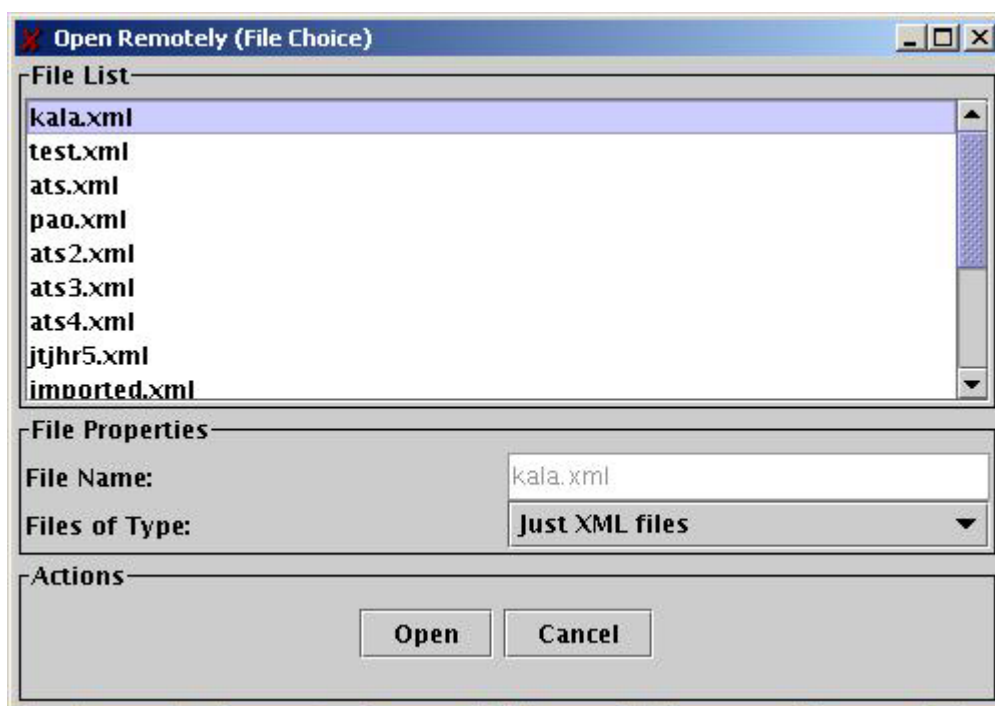


Εικόνα 25 - Φόρμα Ιδιοτήτων Απομακρυσμένου Υπολογιστή

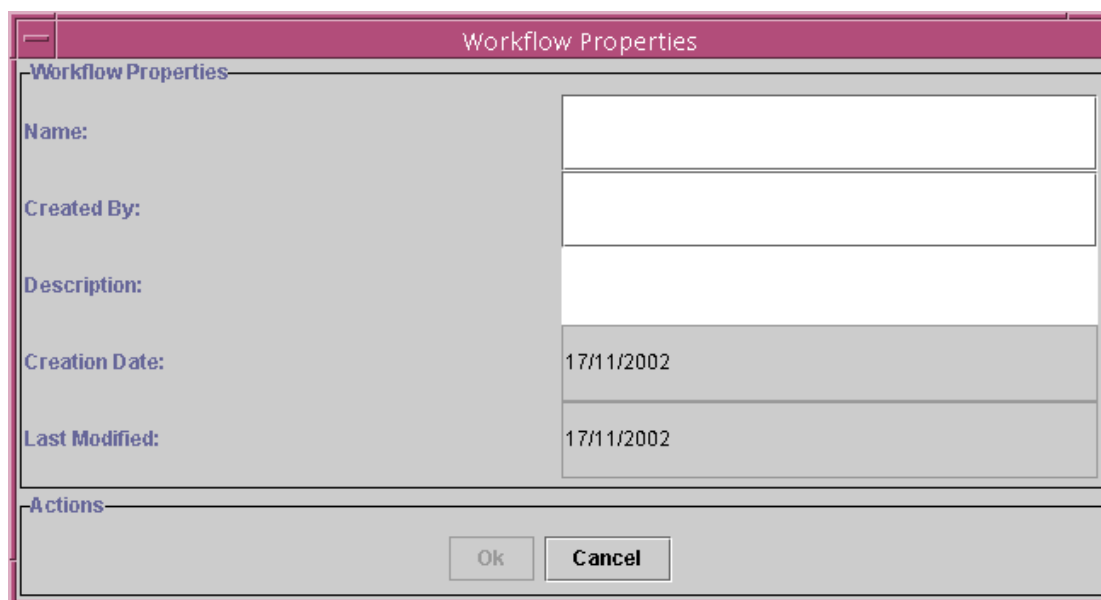
Το στοιχείο μενού ‘Open Remotely’ χρησιμοποιείται για το άνοιγμα ενός προσδιορισμού ροής εργασίας, ο οποίος είναι αποθηκευμένος σε ένα απομακρυσμένο υπολογιστή. Μόλις αυτό το στοιχείο μενού επιλεγθεί, μια φόρμα εμφανίζεται, η οποία προτρέπει τον χρήστη να συμπληρώσει τα στοιχεία του απομακρυσμένου υπολογιστή, δηλαδή το IP και το port του, μέσω ανάλογων πεδίων, όπως φαίνεται και στην εικόνα 25. Έπειτα, όταν ο χρήστης πατήσει το κουμπί ‘Connect’, εγκαθιδρύεται η σύνδεση με τον απομακρυσμένο υπολογιστή. Συνεπακολούθως, μια νέα φόρμα εμφανίζεται – όπως φαίνεται και στην εικόνα 26, η οποία έχει περίπου την ίδια μορφή με αυτή που ‘εκτοξεύεται’ όταν επιλεγεί το στοιχείο μενού ‘Open’. Όταν ο χρήστης επιλέξει το αρχείο που τον ενδιαφέρει, ο επιθυμητός προσδιορισμός ροής εργασίας μεταφέρεται σε ένα προσωρινό τοπικό αρχείο και το γραφικό δένδρο της ροής εργασίας φορτώνεται. Όπως συμβαίνει και με το στοιχείο μενού ‘Open’, αυτό το στοιχείο απενεργοποιείται όταν φορτωθεί ένας προσδιορισμός ροής εργασίας.

Το στοιχείο μενού ‘New’ χρησιμοποιείται για την δημιουργία ενός νέου προσδιορισμού ροής εργασίας. Μόλις αυτό το στοιχείο επιλεγθεί, μια φόρμα με όνομα ‘Workflow Properties’, όπως φαίνεται στο εικόνα 27, φορτώνεται που προτρέπει το χρήστη να δώσει τα στοιχεία που αφορούν τη ροή εργασίας, όπως το όνομά του και τον όνομα του συγγραφέα του, μέσω της συμπλήρωσης των αντίστοιχων πεδίων. Έπειτα, ένα γραφικό δένδρο φορτώνεται, το οποίο έχει μόνο ένα στοιχείο, το ριζικό, με όνομα ‘workflow’. Ο χρήστης πρέπει να εμπλουτίσει το

δένδρο με νέα στοιχεία για να ολοκληρώσει τον ορισμό της ροής εργασίας που τον ενδιαφέρει. Το εν λόγω στοιχείο μενού απενεργοποιείται στις ίδιες περιπτώσεις με τα προαναφερόμενα στοιχεία μενού.



Εικόνα 26 - Επιλογή του Απομακρυσμένου Αρχείου



Εικόνα 27 - Η Φόρμα Επιλογής Ιδιοτήτων Ροής Εργασίας

Το στοιχείο μενού 'Save' χρησιμοποιείται για την αποθήκευση του τρέχοντος προσδιορισμού ροής εργασίας. Μόλις επιλεγθεί, μια φόρμα με όνομα 'Save' φορτώνεται. Ο χρήστης πρέπει να συμπληρώσει το όνομα του αρχείου (αν είναι νέα ροή εργασίας) ή να διαλέξει ένα ήδη υπάρχων (αν πρόκειται να τροποποιήσει μια 'παλιά' ροή εργασίας) και έπειτα να πατήσει το κουμπί 'Save'. Τόσο αυτό το στοιχείο μενού όσο και όλα τα άλλα που χρησιμοποιούνται για την αποθήκευση ροών

εργασίας απενεργοποιούνται στην περίπτωση που δεν υπάρχει ένα προσδιορισμός ροής εργασίας που να έχει 'φορτωθεί'.

Το στοιχείο μενού 'Save As' χρησιμοποιείται για την αποθήκευση (ενός ήδη αποθηκευμένου) προσδιορισμού ροής εργασίας με νέο όνομα. Ότι αλλαγές γίνουν από εκεί και πέρα σε αυτόν τον προσδιορισμό θα αντικατοπτριστούν στο αρχείο με το νέο όνομα.

Το στοιχείο μενού 'Save Remotely' χρησιμοποιείται για την αποθήκευση ενός προσδιορισμού ροής εργασίας σε ένα απομακρυσμένο υπολογιστή. Μόλις αυτό το στοιχείο επιλεγεί, μια φόρμα εμφανίζεται. Ο χρήστης πρέπει να συμπληρώσει τα πεδία της φόρμας που φανερώνουν την τοποθεσία του υπολογιστή στο διαδίκτυο και να πατήσει το κουμπί 'Connect'. Έπειτα, μια νέα φόρμα εμφανίζεται παρόμοια με αυτή του στοιχείου 'Save' οπότε και η διαδικασία συμπλήρωσής της είναι παρόμοια.

Το στοιχείο 'Save As Remotely' παρουσιάζει την ίδια λειτουργικότητα με αυτή του στοιχείου 'Save As', μόνο που τα στοιχεία της τοποθεσίας του απομακρυσμένου υπολογιστή πρέπει να συμπληρωθούν αρχικά.

Το στοιχείο μενού 'Execute' χρησιμοποιείται για την δημιουργία ενός προσδιορισμού περίπτωσης ροής εργασίας από ένα προσδιορισμό ροής εργασίας και την εκτέλεση αυτού του νέου προσδιορισμού. Μόλις αυτό το στοιχείο επιλεγεί, ο Κειμενογράφος Περιπτώσεων Ροών Εργασίας εμφανίζεται, η λειτουργικότητα του οποίου θα αναλυθεί παρακάτω.

Τα στοιχεία μενού 'Import As First Child' και 'Import As Right Sibling' χρησιμοποιείται για την ένωση ενός προσδιορισμού ροής εργασίας με ένα επιλεγμένο στοιχείο-κόμβο του υπό επεξεργασία προσδιορισμού. Το μεν πρώτο στοιχείο μενού 'κολλάει' το εισαγόμενο προσδιορισμό ως πρώτο παιδί του στοιχείου-κόμβου του υπό επεξεργασία προσδιορισμού. Το δε δεύτερο στοιχείο μενού 'κολλάει' τον εισαγόμενο προσδιορισμό ως επόμενο αδερφό του επιλεγμένου στοιχείου. Η διαδικασία εισαγωγής είναι η ίδια με την διαδικασία που ακολουθείται για το άνοιγμα ενός προσδιορισμού. Η μόνη διαφορά είναι ότι αν ο προσδιορισμός ροής εργασίας δεν ταιριάζει ως παιδί ή ως αδερφός του επιλεγμένου στοιχείου-κόμβου, τότε εμφανίζεται ένα μήνυμα λάθους στο χρήστη και η διαδικασία εισαγωγής ακυρώνεται. Το εν λόγω στοιχείο μενού απενεργοποιείται ή ενεργοποιείται ανάλογα με το εάν το επιλεγμένο στοιχείο-κόμβος του υπό επεξεργασία προσδιορισμού ροής εργασίας μπορεί να έχει παιδί ή επόμενο αδερφό.

Το στοιχείο μενού 'Close' χρησιμοποιείται για το κλείσιμο ενός προσδιορισμού ροής εργασίας. Μόλις επιλεγεί το εν λόγω στοιχείο και ο προσδιορισμός δεν έχει σωθεί – ενώ έχει τροποποιηθεί από την τελευταία φορά που είχε 'σωθεί' – τότε ο χρήστης επερωτάτε αν θέλει να 'σώσει' τις τελευταίες αλλαγές ή όχι. Αν απαντήσει όχι, τότε οι αλλαγές χάνονται. Αν απαντήσει ναι, τότε ακολουθείται η διαδικασία αποθήκευσης του προσδιορισμού. Πάντως, σε κάθε περίπτωση, ο προσδιορισμός ροής εργασίας 'κλείνει', και το γραφικό του δένδρο εξαφανίζεται.

Το στοιχείο μενού 'Exit' χρησιμοποιείται για την έξοδο από τον κειμενογράφο. Αν το στοιχείο αυτό επιλεγεί, τότε ελέγχεται αν υπάρχει προσδιορισμός που δεν έχει σωθεί. Αν υπάρχει τέτοιος προσδιορισμός, τότε καλείται το στοιχείο μενού 'Close'. Αν η έπειτα διαδικασία ακυρωθεί, τότε ακυρώνεται και η έξοδος από τον κειμενογράφο. Σε όλες τις άλλες περιπτώσεις έχουμε έξοδο από τον κειμενογράφο.

Το στοιχείο μενού 'Cut' χρησιμοποιείται για την αποκοπή του τρέχοντος (δηλαδή του επιλεγμένου) στοιχείου-κόμβου του υπό επεξεργασία προσδιορισμού ροής εργασίας. Ο κόμβος αυτός αντιγράφεται σε περίπτωση που επικολληθεί σε ένα

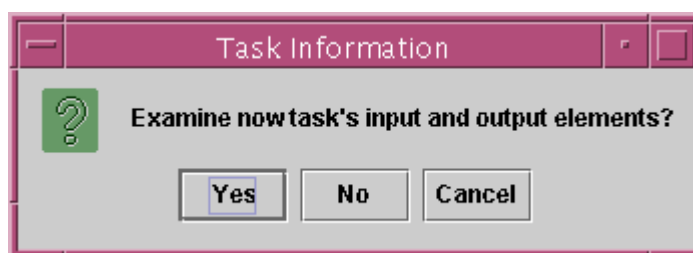
άλλο κόμβο του υπό επεξεργασία προσδιορισμού. Το ριζικό στοιχείο ενός προσδιορισμού δεν μπορεί να αποκολληθεί.

Το στοιχείο μενού 'Copy' χρησιμοποιείται για την αντιγραφή ενός επιλεγμένου κόμβου του υπό επεξεργασία προσδιορισμού. Ο ριζικός κόμβος δεν μπορεί να αντιγραφθεί.

Τα στοιχεία μενού 'Paste As First Child' και 'Paste As Right Sibling' χρησιμοποιείται για την επικόλληση ενός αντιγραμμένου ή αποκομμένου κόμβου σε ένα επιλεγμένο κόμβο του υπό επεξεργασία προσδιορισμού είτε ως πρώτο παιδί είτε ως επόμενος αδερφός. Σε περίπτωση που η γλώσσα AWL δεν επιτρέπει την επικόλληση, το εν λόγω στοιχείο μενού απενεργοποιείται.

Το στοιχείο μενού 'Edit Properties' ενεργοποιείται όταν επιλέγονται στοιχεία του προσδιορισμού που έχουν ιδιότητες που μπορούν να τροποποιηθούν. Ανάλογα με το ποιος είναι ο τρέχων κόμβος-στοιχείο όταν το εν λόγω στοιχείο μενού επιλεγθεί, μια διαφορετική φόρμα παρουσιάζεται. Αυτές οι φόρμες θα εξεταστούν παρακάτω, διότι χρησιμοποιούνται και για τον εμπλουτισμό του δένδρου ενός προσδιορισμού.

Τα μενού 'Child Operations' και 'Sibling Operations' περιέχουν τα ίδια στοιχεία αλλά διαφορετική λειτουργικότητα ανά στοιχείο. Το μενού 'Child Operations' περιέχει λειτουργίες που χρησιμοποιούνται για τη προσθήκη στοιχείου ροής εργασίας ως κόμβου 'πρώτο παιδί' κάτω από τον τρέχων κόμβο του υπό επεξεργασία προσδιορισμού. Από την άλλη πλευρά, οι λειτουργίες του μενού 'Sibling Operations' χρησιμοποιούνται για την προσθήκη ενός στοιχείου ροής εργασίας ως κόμβου 'επόμενος αδερφός' δίπλα στον τρέχων κόμβο του υπό επεξεργασία προσδιορισμού. Και τα δύο μενού περιέχουν τα ακόλουθα στοιχεία: 'Add A Task Element', 'Add a Sequence Element', 'Add a Parallel_Sync Element', 'Add a Parallel_no_sync Element', 'Add a While-do Element', 'Add a Backtrack Element', 'Add a Data Set Element' και 'Add a Parameter Element'. Αυτά τα στοιχεία μενού ενεργοποιούνται ανάλογα με τον αν πρόκειται να ικανοποιήσουν όλους τους περιορισμούς της γλώσσας AWL όταν θα προσθέσουν ένα στοιχείο ροής εργασίας κάποιου είδους δίπλα ή κάτω από ένα τρέχων στοιχείο.



Εικόνα 28 - Φόρμα Επερώτησης Πληροφορίας για Έργο

Μόλις το στοιχείο μενού 'Add a Task Element' επιλεγθεί, ο χρήστης επερωτάτε αν θέλει να παραδώσει όλη τη πληροφορία σχετικά με το έργο που θα προστεθεί στον προσδιορισμό ή αν θα φορτωθεί μια προεπιλεγμένη πληροφορία για αυτό το έργο, όπως φαίνεται στην εικόνα 28. Αν επιλεγθεί η δεύτερη επιλογή, τότε ένα στοιχείο 'task' θα δημιουργηθεί, το οποίο θα περιέχει ένα στοιχείο 'data-set' για είσοδο και ένα για έξοδο καθώς και ένα στοιχείο προγράμματος. Σε κάθε ένα από τα εσωτερικά στοιχεία θα δοθούν πλασματικές τιμές στις υποχρεωτικές ιδιότητές τους. Αν, όμως, επιλεγθεί η πρώτη επιλογή, τότε θα εμφανιστεί μια φόρμα με όνομα 'Input DataSet Properties', όπως φαίνεται στην εικόνα 29. Αρχικά, σε αυτή τη φόρμα, ο χρήστης πρέπει να δώσει τιμή στο όνομα του στοιχείου "data-set", το οποίο θα είναι είσοδος στο προστιθέμενο στοιχείο 'task'. Έπειτα, ο χρήστης θα πρέπει να αποφασίσει αν το εν λόγω στοιχείο "data-set" θα αναφέρεται σε ένα στοιχείο 'data-

set' της εξόδου ενός προηγούμενου έργου της ροής εκτέλεσης της υπό επεξεργασία ροής εργασίας. Μετά θα πρέπει να αποφασίσει ποιος θα είναι ο τύπος του 'input data-set'. Αν είναι τύπου 'general', τότε δεν πρέπει να δώσει την πληροφορία για το σε ποιον υπολογιστή και διεύθυνση βρίσκεται το στοιχείο 'data-set'. Αν, όμως, είναι τύπου 'specific', τότε θα πρέπει να δώσει την προαναφερόμενη πληροφορία. Η πληροφορία αυτή θα αφορά το URL της διεύθυνσης του συνόλου δεδομένων, αν αυτό βρίσκεται σε διεύθυνση προσβάσιμη από το διαδίκτυο. Ενώ θα αφορά το όνομα του υπολογιστή και το μονοπάτι του λειτουργικού συστήματος (URI), σε κάθε άλλη περίπτωση. Όταν τελειώσουν όλα αυτά, ο χρήστης πρέπει να δώσει το όνομα και το URI της επιστημονικής οντολογίας (ή ενός κομματιού της) και το όνομα της οντότητας, περίπτωση της οποίας θα είναι το εν λόγω σύνολο δεδομένων. Στο τέλος, ο χρήστης πρέπει να πατήσει το κουμπί 'Save'. Αν τώρα θέλει να δημιουργήσει ένα νέο 'input data-set', τότε θα πρέπει να πατήσει το κουμπί 'New' ενώ αν θέλει να συνεχίσει το ορισμού του στοιχείου 'task' θα πρέπει να πατήσει το κουμπί 'Continue'.

Η επόμενη φόρμα που θα εμφανιστεί στον χρήστη είναι η φόρμα με όνομα 'Input Parameter Properties', όπως φαίνεται στην εικόνα 30. Εδώ ο χρήστης πρέπει να παραδώσει το όνομα, τον τύπο και μια προκαθορισμένη τιμή για αυτή τη παράμετρο. Έπειτα, θα πρέπει να αναφέρει σε τι είδους περιορισμό υπόκειται αυτή η παράμετρος και ποιος είναι ακριβώς ο περιορισμός. Μετά, θα πρέπει να πατήσει το κουμπί 'Save'. Τέλος, θα πρέπει να αποφασίσει αν θα συνεχίσει με ένα νέο ορισμό στοιχείου 'input parameter' (κουμπί 'New') ή θα συνεχίσει (κουμπί 'Continue').

Input DataSet Properties			
General Info			
ID:	data-set0		
Name:			
Referencing:	NONE	With ID:	NONE
URI:			
Hostname:			
Type:	NONE		
Ontology Related Info			
Ontology Name:	OCEAN_FEATURE		
Ontology URI:	http://www.csi.forth.gr/~kritikos/MyOnt...		
Selected Classes			
OCEAN_FEATURE			
Change Selection			
Actions			
New Save Continue Cancel			

Εικόνα 29 - Η Φόρμα Ιδιοτήτων Συνόλου Δεδομένων

Επακολούθως, εμφανίζεται η φόρμα με το όνομα 'Program Properties', όπως φαίνεται στην εικόνα 31. Εδώ ο χρήστης πρέπει να δώσει το όνομα του

προγράμματος του προστιθέμενου έργου. Έπειτα, θα πρέπει να τονίσει αν το πρόγραμμά του αναφέρεται σε ένα ορισμό προγράμματος ενός προηγούμενου έργου της ροής εκτέλεσης. Ακόμη, θα πρέπει να δώσει το όνομα του υπολογιστή και το μονοπάτι του λειτουργικού συστήματος στο οποίο τρέχει το εν λόγω πρόγραμμα. Τέλος, θα πρέπει να αναφέρει το είδος του προγράμματος (αν είναι προσομοίωσης ή οπτικοποίησης) και αν μπορεί να εκτελεστεί ταυτόχρονα από πολλούς χρήστες (ιδιότητα 'sync'). Με το πάτημα του κουμπιού 'Save' θα τελειώσει ο ορισμός του προγράμματος.

Μετάπειτα, εμφανίζεται η φόρμα με όνομα 'Output DataSet Properties'. Αυτή η φόρμα χρησιμοποιείται για τον ορισμό ενός 'data-set' στοιχείου που θα παράγεται κατά την εκτέλεση του προστιθέμενου στοιχείου 'task'. Η διαδικασία ορισμού είναι η ίδια με αυτή που συντελείται στη φόρμα 'Input DataSet Properties'.

Εν τέλει, εμφανίζεται η φόρμα 'Output Parameter Properties'. Αυτή χρησιμοποιείται για τον ορισμό μιας παραμέτρου εξόδου. Η διαδικασία ορισμού της είναι η ίδια με αυτήν του ορισμού μιας παραμέτρου εισόδου.

Προτού προχωρήσουμε στον γραφικό τρόπο ορισμού/προσθήκης των δομικών στοιχείων της ροής εκτέλεσης/ελέγχου μιας ροής εργασίας, θα πρέπει να επεξηγήσουμε περισσότερο πως γίνεται ο καθορισμός του ονόματος ή των ονομάτων οντοτήτων, περίπτωση των οποίων θα είναι ένα σύνολο δεδομένων. Αρχικά, όταν ορίζεται ένα σύνολο δεδομένων, αυτό έχει μια προκαθορισμένη τιμή για την παραπάνω ιδιότητα, η οποία τιμή αφορά το ριζικό στοιχείο της επιστημονικής οντολογίας. Ο χρήστης μπορεί να προσθέσει ή να αλλάξει την τιμή αυτή πατώντας το κουμπί 'Change'. Τότε θα εμφανιστεί μια φόρμα με όνομα 'Class Selection', όπως φαίνεται στην εικόνα 32. Αυτή η φόρμα αποτελείται στο πρώτο της μισό από δύο υπο-παράθυρα και από τρία κουμπιά. Το πρώτο υπο-παράθυρο φορτώνει την επιστημονική οντολογία σε ένα δένδρο. Το δένδρο αυτό απεικονίζει όλες τις οντότητες της οντολογίας και την σχέση 'subclassof' που έχουν αυτές μεταξύ τους. Το δεύτερο υπο-παράθυρο απεικονίζει είτε τις ιδιότητες μιας οντότητας (αν πατηθεί το πρώτο κουμπί 'View Properties') είτε τις περιπτώσεις της (αν πατηθεί το κουμπί 'View instances'). Το τρίτο κουμπί προσθέτει το όνομα μιας επιλεγμένης οντότητας στη λίστα με τις επιλεγμένες τιμές, η οποία βρίσκεται στο άλλο μισό. Στο δεύτερο μισό, λοιπόν, της επικείμενης φόρμας, υπάρχουν δύο λίστες και κάποια κουμπιά ανάμεσά τους. Η πρώτη λίστα φανερώνει τα ονόματα των οντοτήτων που δεν έχουν επιλεγθεί ακόμη. Η δεύτερη λίστα φανερώνει τα ονόματα που έχουν επιλεγθεί. Τα κουμπιά ανάμεσα μεταφέρουν ονόματα από τη μια λίστα στην άλλη. Όταν ο χρήστης πατήσει το κουμπί 'Change' θα μεταφερθούν οι αλλαγές του στον ορισμό του εν λόγω συνόλου δεδομένων. Αν πατήσει το κουμπί 'Cancel', τότε οι αλλαγές θα ακυρωθούν.

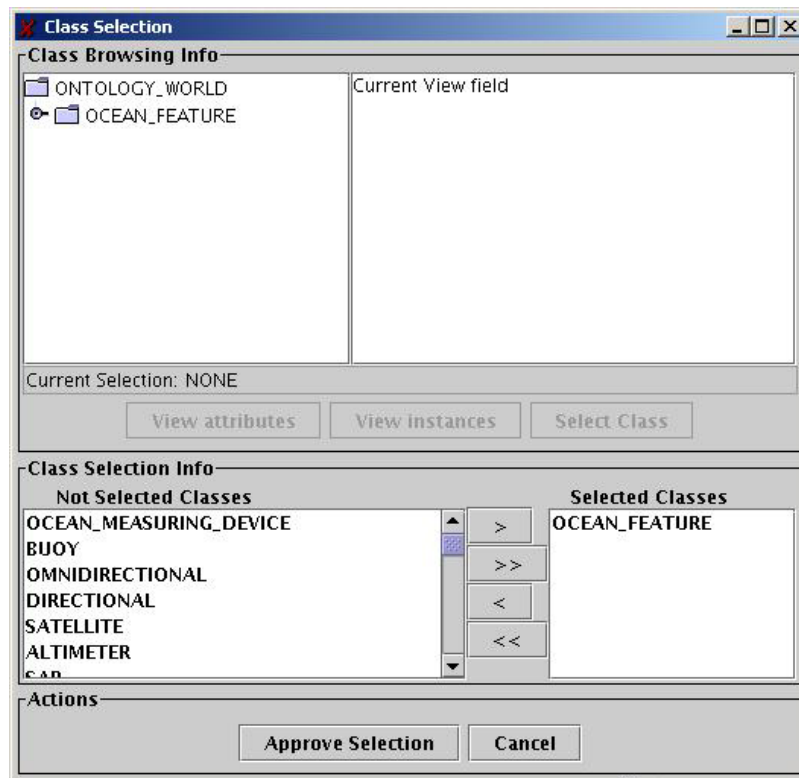
Το στοιχείο μενού 'Add a Sequence Element' χρησιμοποιείται για την προσθήκη ενός στοιχείου 'sequence' στον υπό επεξεργασία προσδιορισμό. Αυτό το στοιχείο, όπως και τα 'parallel_sync' και 'parallel_no_sync' που προστίθενται από τα στοιχεία μενού 'Add a Parallel_sync Element' και 'Add a Parallel_no_sync Element' αντίστοιχα, μπορούν να έχουν πολλά παιδιά τύπου 'workflow element'.

Το στοιχείο μενού 'Add a While-do Element' χρησιμοποιείται για την προσθήκη ενός επαναληπτικού στοιχείου 'while-do' στον υπό επεξεργασία προσδιορισμό. Μόλις αυτό το στοιχείο επιλεγθεί, μια φόρμα εμφανίζεται που προτρέπει στον χρήστη να διατυπώσει μια συνθήκη, όπως φαίνεται στην εικόνα 33. Όσο αυτή η συνθήκη είναι αληθής, θα επαναλαμβάνεται η εκτέλεση του ενός παιδιού του εν λόγω στοιχείου.

Εικόνα 30 - Η Φόρμα Ιδιοτήτων Παραμέτρου

Εικόνα 31 - Η Φόρμα Ιδιοτήτων Προγράμματος

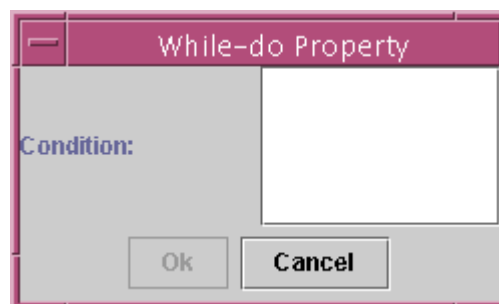
Το στοιχείο μενού ‘Add a Backtrack Element’ χρησιμοποιείται για τον ορισμό ενός στοιχείου ‘backtrack’ στον υπό επεξεργασία προσδιορισμό. Το πρώτο και το τελευταίο (σε εκτέλεση) στοιχείο ‘task’ του μοναδικού παιδιού του εν λόγω στοιχείου θα συνδέονται με τη σχέση “backtrack”.



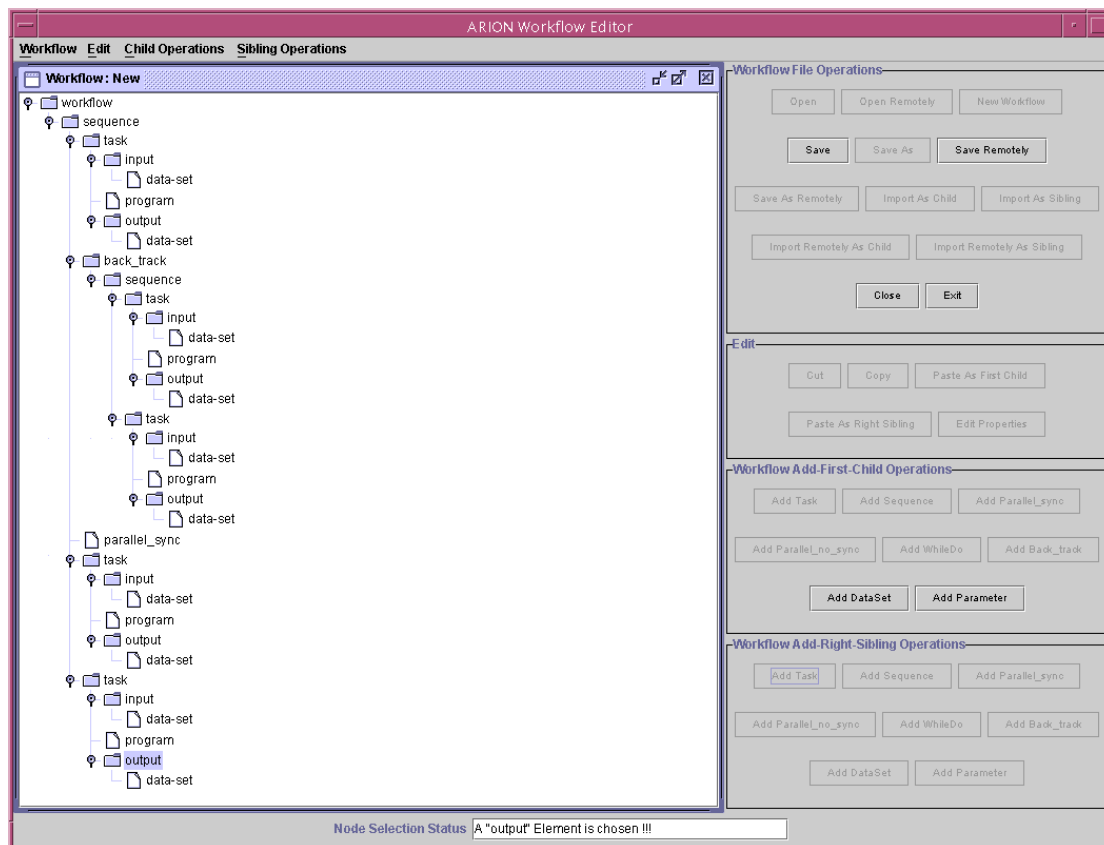
Εικόνα 32 - Η Φόρμα Επιλογής Ονόματος Κλάσης της Επιστημονικής Οντολογίας

Το στοιχείο μενού 'Add a DataSet Element' χρησιμοποιείται για τη προσθήκη ενός στοιχείου 'data-set' είτε εισόδου είτε εξόδου σε ένα στοιχείο 'task' του υπό επεξεργασία προσδιορισμού. Η φόρμα που εμφανίζεται έπειτα από την επιλογή του συγκεκριμένου στοιχείου μενού είναι παρόμοια οπτικά με τη φόρμα 'Input (Output) DataSet Properties'. Η διαδικασία συμπλήρωσής της είναι και αυτή παρόμοια με αυτή που συντελείται κατά τον πλήρη ορισμό ενός στοιχείου 'task'.

Το στοιχείο μενού 'Add a Parameter Element' χρησιμοποιείται για τη προσθήκη ενός στοιχείου 'parameter' είτε εισόδου είτε εξόδου σε ένα στοιχείο 'task' του υπό επεξεργασία προσδιορισμού. Η φόρμα που εμφανίζεται έπειτα από την επιλογή του συγκεκριμένου στοιχείου μενού είναι παρόμοια οπτικά με τη φόρμα 'Input (Output) Parameter Properties'. Η διαδικασία συμπλήρωσής της είναι και αυτή παρόμοια με αυτή που συντελείται κατά τον πλήρη ορισμό ενός στοιχείου 'task'.



Εικόνα 33 - Η Φόρμα Ιδιοτήτων του Στοιχείου 'While-do'



Εικόνα 34 - Ένας πλήρης προσδιορισμός ροής εργασίας

Γ.ΙΙΙ. Ο Κειμενογράφος Περιπτώσεων Ροών Εργασίας

Ο Κειμενογράφος Περιπτώσεων Ροών Εργασίας είναι ένα γραφικό εργαλείο ορισμού περιπτώσεων ροών εργασίας. Αποτελεί κομμάτι του Κειμενογράφου Ροών Εργασίας ενώ μπορεί να λειτουργήσει και ως μια ανεξάρτητη εφαρμογή. Λειτουργεί είτε off-line είτε στο διαδίκτυο με τη μορφή ενός Java Applet. Χρησιμοποιείται μόνο από εξειδικευμένους χρήστες του συστήματος ARION, οι οποίοι και πρέπει να είναι εξοικειωμένοι τόσο με το μοντέλο επιστημονικής ροής εργασίας όσο και με τη γλώσσα AWL. Η υλοποίησή του έγινε και πάλι με τη χρήση της γλώσσας Java.

Βασικές Λειτουργίες

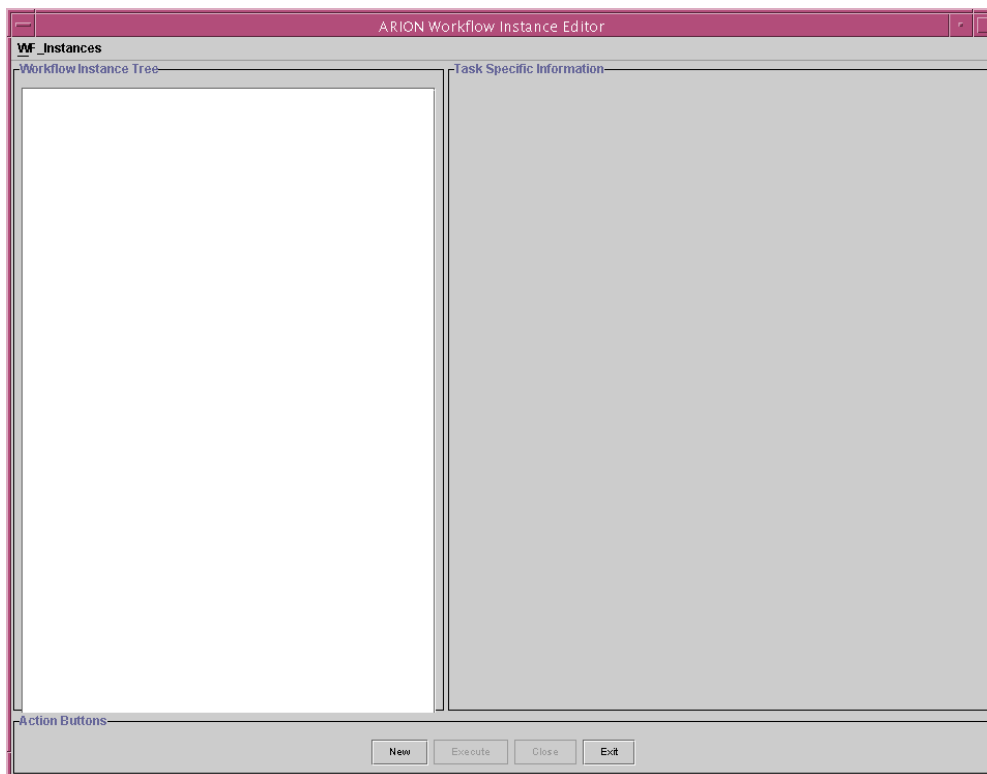
Ο Κειμενογράφος Περιπτώσεων Ροών Εργασίας συνδέεται με σχεδόν όλα τα συστατικά μέρη του συστήματος Κ.Α.Ρ.Ε.Ε.Α ώστε να επιτελέσει τις ακόλουθες λειτουργίες:

- Ταυτοποίησης εξειδικευμένου χρήστη.
- Εύρεση ροών εργασίας που παράγουν σύνολα δεδομένων συγκεκριμένου τύπου.
- Ορισμός περίπτωσης ροής εργασίας που εκτελείται αλληλεπιδραστικά με το χρήστη, ο οποίος πρέπει να δίνει τιμές στις παραμέτρους των προγραμμάτων.
- Ορισμός περίπτωσης ροής εργασίας που εκτελείται στο background χωρίς την αλληλεπίδραση με κάποιον χρήστη.

- Μεταφορά ορισμού περίπτωσης ροής εργασίας στο σύστημα Εκτέλεσης Ροών Εργασίας προς εκτέλεσή τους.
- Για κάθε ορισμό περίπτωσης ροής εργασίας που θα εκτελεστεί, αποθήκευση τόσο του XML αρχείου προσδιορισμού όσο και του ειδικού αρχείου των παραμέτρων εκτέλεσης (αν χρειάζεται).

Η Γραφική Διεπαφή Χρήσης του Κειμενογράφου

Ο Κειμενογράφος Περιπτώσεων Ροών Εργασίας εμφανίζεται στην εικόνα 35.



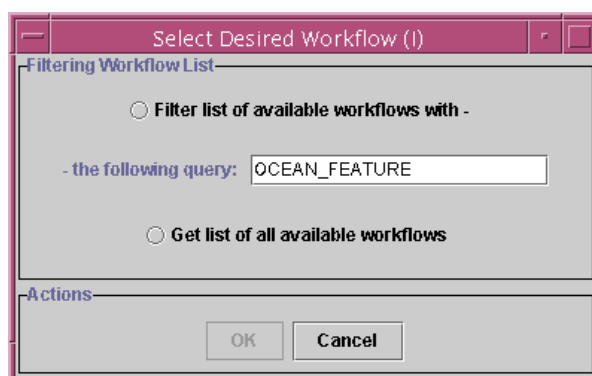
Εικόνα 35 - Ο Κειμενογράφος Περιπτώσεων Ροών Εργασίας

Η πρώτη διεπαφή που παρουσιάζεται στον χρήστη όταν ‘ανοίγει’ τον Κειμενογράφο Περιπτώσεων Ροών Εργασίας είναι η Φόρμα Ταυτοποίησης. Αυτή αποτελείται από δύο πεδία που ονομάζονται “login” και “password” και από δύο κουμπιά με ονόματα: “Ok” και “Cancel”. Για να ταυτοποιηθεί, ο χρήστης πρέπει να συμπληρώσει τα στοιχεία του γεμίζοντας τα προαναφερόμενα πεδία και έπειτα να πατήσει το κουμπί “Ok”. Αν τα στοιχεία του χρήστη είναι λανθασμένα, τότε τα πεδία καθαρίζονται και πρέπει να ξανα-επαναληφθεί η προηγούμενη διαδικασία.

Μόλις η διαδικασία ταυτοποίησης επιτύχει, το κύριο πλαίσιο-παράθυρο του κειμενογράφου εμφανίζεται. Αυτό το πλαίσιο αποτελείται από: μια μπάρα από μενού, ένα γραφικό δένδρο, ένα άδειο υπό-παράθυρο και ένα πάνελ με μια ομάδα από κουμπιά. Η μπάρα από μενού προσφέρει τόσο την ίδια ομαδοποίηση στα μενού όσο και την ίδια λειτουργικότητα (ως προς τα στοιχεία των μενού – menu items) με αυτή του πάνελ της ομάδας των κουμπιών. Τόσο τα στοιχεία του μενού όσο και τα κουμπιά του πάνελ έχουν τα εξής ονόματα: ‘New’, ‘Execute’, ‘Close’ και ‘Exit’. Το γραφικό δένδρο αναπαριστά τη ροή του ελέγχου της υπό επεξεργασία ροής εργασίας

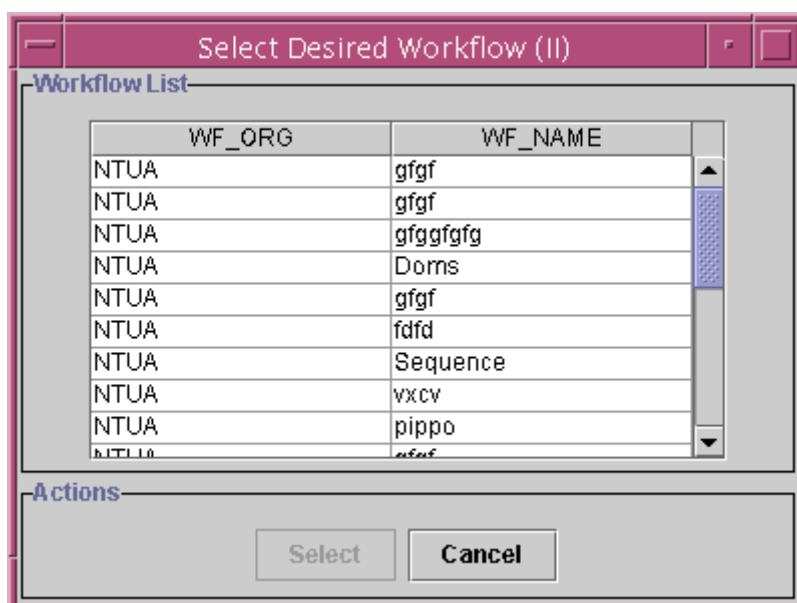
αποδίδοντας μια ιεραρχική δενδρική δομή των στοιχείων της ροής εργασίας, τα οποία και απαρτίζουν τη εν λόγω ροή ελέγχου. Στο δένδρο αυτό με ξεχωριστό χρώμα (πράσινο) απεικονίζονται τα στοιχεία 'task' που είναι πλήρως ορισμένα από αυτά που δεν είναι (χρώμα κόκκινο). Στο υπο-παράθυρο φορτώνεται η φόρμα ορισμού ενός στοιχείου 'task', όταν το τελευταίο επιλέγεται από τον χρήστη.

Το στοιχείο μενού 'New' χρησιμοποιείται για την έναρξη του ορισμού μιας περίπτωσης ροής εργασίας. Μόλις επιλεγθεί αυτό το στοιχείο από το χρήστη, 'φορτώνεται' μια φόρμα με όνομα 'Select Desired Workflow I', όπως φαίνεται στην εικόνα 36. Σε αυτή τη φόρμα ο χρήστης επιλέγει αν θα ζητήσει να του εμφανιστεί μια πλήρης λίστα με όλες τις ροές εργασίας ή αν θα διατυπώσει μια επερώτηση. Αν επιλέξει το δεύτερο, θα πρέπει να δώσει και το όνομα της οντότητας, περιπτώσεις της οποίας θα παράγουν οι ζητούμενες ροές εργασίας. Ανεξάρτητα από ποια είναι η επιλογή του χρήστη, θα 'φορτωθεί' μια νέα φόρμα με όνομα 'Select Desired Workflow II', όπως φαίνεται στην εικόνα 37. Σε αυτή τη νέα φόρμα θα εμφανίζεται μια λίστα από ροές εργασίας που θα έχουν ένα συγκεκριμένο όνομα και θα παράγουν σύνολα δεδομένων κάποιου τύπου. Ο χρήστης καλείται να διαλέξει μια από αυτές τις ροές εργασίας και να πατήσει το κουμπί 'Select'. Κατόπιν τούτου, εμφανίζεται μια ακόμη φόρμα με όνομα 'Workflow Instance Processing Mode', όπως φαίνεται στην εικόνα 38. Σε αυτή τη φόρμα καλείται ο χρήστης να διαλέξει αν θέλει να δημιουργήσει μια απλή περίπτωση ροής εργασίας ή μια περίπτωση που θα τρέχει στο background. Αν διαλέξει το δεύτερο, τότε θα πρέπει επίσης να αναφέρει αν η περίπτωση ροής εργασίας θα τρέξει συγκεκριμένες φορές ή αν οι φορές εκτέλεσής της θα παράγονται από τους συνδυασμούς των φορών εκτέλεσης όλων των έργων. Εδώ πρέπει να εξηγήσουμε ότι αν μια παράμετρος ενός έργου πάρει πολλές τιμές, τότε η περίπτωση ροής εργασίας θα σπάσει σε τόσες περιπτώσεις όσες είναι και οι τιμές που παίρνει αυτή η παράμετρος. Αν και μια άλλη παράμετρος ενός έργου παίρνει πολλές τιμές, μπορεί οι τιμές των δύο παραμέτρων του έργου να συνδυαστούν (αν το θέλει ο χρήστης) και η περίπτωση ροής εργασίας να σπάσει σε περισσότερες περιπτώσεις. Το πλήθος των συνδυασμών των τιμών των παραμέτρων ενός έργου το ονομάζουμε φορές εκτέλεσης ενός έργου. Τώρα, αν ο χρήστης το επιθυμεί, ακόμη και οι φορές εκτέλεσης των έργων μπορούν να συνδυαστούν. Επομένως, έχουμε δύο επίπεδα συνδυασμών φορών εκτέλεσης μιας περίπτωσης ροής εργασίας. Όταν και η συμπλήρωση της τρίτης φόρμας επιτύχει, ο ορισμός της περίπτωσης ροής εργασίας φορτώνεται στο γραφικό δένδρο. Θα πρέπει εδώ να σημειωθεί ότι όταν ο Κειμενογράφος Περιπτώσεων Ροών Εργασίας φορτωθεί μέσω του Κειμενογράφου Ροών Εργασίας, τότε αρχίζουμε με την Τρίτη φόρμα διότι ο προσδιορισμός της επιθυμητής ροής εργασίας είναι ήδη γνωστός.



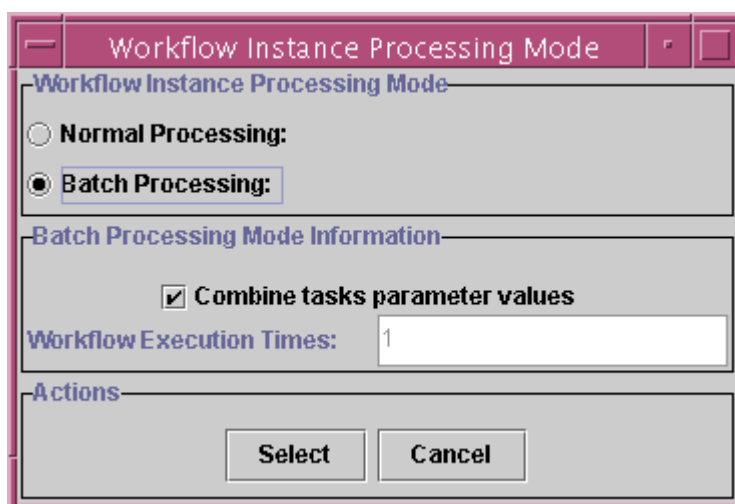
Εικόνα 36 - Η Φόρμα Επιλογής της Ροής Εργασίας 1

Το στοιχείο μενού 'Execute' χρησιμοποιείται για την αποστολή ενός ορισμού περίπτωσης ροής εργασίας στο σύστημα Εκτέλεσης Ροών Εργασίας. Αυτό το στοιχείο ενεργοποιείται μόνο όταν όλα τα έργα του προσδιορισμού περίπτωσης ροής εργασίας έχουν πλήρως οριστεί. Όταν αναφέρουμε 'πλήρως οριστεί', εννοούμε ότι για όλα τα σύνολα δεδομένων εισόδου του έργου θα πρέπει να έχει οριστεί η τοποθεσία τους στο διαδίκτυο. Επιπλέον, αν η περίπτωση ροής εργασίας θα τρέχει στο background, τότε κάθε έργο θα είναι πλήρως ορισμένο αν και όλες οι υποχρεωτικοί παράμετροι εισόδου του πάρουν μια ή περισσότερες τιμές.



Εικόνα 37 - Φόρμα Επιλογής Ροής Εργασίας 2

Το στοιχείο μενού 'Close' χρησιμοποιείται για το κλείσιμο του τρέχοντος προσδιορισμού περίπτωσης ροής εργασίας.



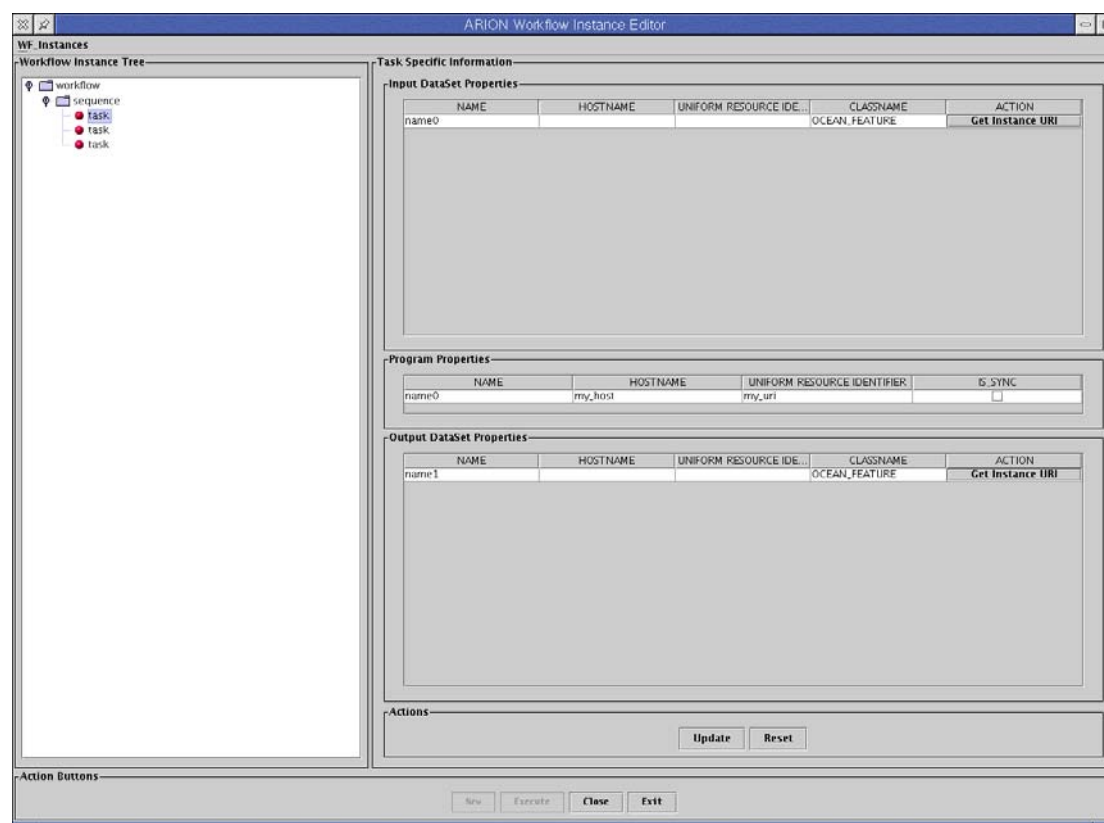
Εικόνα 38 - Φόρμα Εισαγωγής Ιδιοτήτων Περίπτωσης Ροής Εργασίας

Το στοιχείο μενού 'Exit' χρησιμοποιείται για την έξοδο από τον κειμενογράφο και είναι πάντα ενεργοποιημένο.

Το υπο-παράθυρο του κειμενογράφου 'γεμίζει' με μια φόρμα με όνομα 'Task Specification' κάθε φορά που επιλέγεται ένα (κόκκινο ή πράσινο) στοιχείο 'task'.

Ανάλογα με το είδος της περίπτωσης ροής εργασίας, η φόρμα αυτή έχει και διαφορετικό περιεχόμενο. Αν η περίπτωση ροής εργασίας είναι αλληλεπιδραστική με τον χρήστη, τότε η φόρμα φαίνεται στην εικόνα 39 και σε αυτή εμφανίζονται τα εξής:

- Μια λίστα από σύνολα δεδομένων εισόδου. Αυτή η λίστα παρουσιάζει σε κάθε γραμμή της: το όνομα του συνόλου δεδομένων, την τοποθεσία του (αν είναι ορισμένη) δηλαδή το Hostname και το URI και το όνομα της οντότητας της επιστημονικής οντολογίας, για την οποία οντότητα το σύνολο δεδομένων αποτελεί περίπτωση. Επίσης, σε κάθε γραμμή (row) της λίστας εμφανίζεται και ένα κουμπί με όνομα 'Change location'.
- Πληροφορίες σχετικά με το πρόγραμμα του συγκεκριμένου έργου όπως είναι το όνομά του, η τοποθεσία του και αν μπορεί να εκτελεστεί ταυτόχρονα από πολλούς χρήστες.
- Μια αντίστοιχη λίστα με την πρώτη που παρουσιάζει πληροφορία για τα σύνολα δεδομένων της εξόδου



Εικόνα 39 - Απλή Περίπτωση Ροής Εργασίας

Ο χρήστης θα πρέπει να ορίσει την τοποθεσία όλων των συνόλων δεδομένων και να πατήσει το κουμπί 'Save' ώστε το χρώμα του στοιχείου 'task' να γίνει από κόκκινο σε πράσινο και άρα το εν λόγω στοιχείο να είναι πλήρως ορισμένο. Ο ορισμός της τοποθεσίας μπορεί να γίνει με δύο τρόπους. Ο πρώτος τρόπος είναι άμεσος με τη συμπλήρωση του πεδίου URI ή/και του πεδίου HostName. Ο δεύτερος τρόπος είναι έμμεσος. Ο χρήστης πατάει το κουμπί 'Change location' και του εμφανίζεται μια νέα φόρμα, όπως φαίνεται στην εικόνα 40, που παρουσιάζει τις περιπτώσεις της οντότητας στην οποία ανήκει το υπό επεξεργασία σύνολο δεδομένων και τις τιμές που παίρνουν αυτές οι περιπτώσεις για κάθε ιδιότητα της οντότητας της

επιστημονικής οντολογίας. Ο χρήστης διαλέγει μια περίπτωση που τον ενδιαφέρει και αμέσως η τοποθεσία της περίπτωσης αυτής αναγράφεται στο πεδίο της τοποθεσίας του υπό επεξεργασίας συνόλου δεδομένων.

NUM	OF_CON	WD_SEA	OF_ABT	OF_CON	MTS_TI	MD_MO	NAME	OF_CON	OF_ORIG	OF_URL	MD_INP	MTS_TI	WP_CALL	MTS_TI	MTS_LAT	OF_SUPP	WD_PAR	MTS_LONG	WP_TYPE	OF_PURP	OF_CON	OF_PUBL	ACTION	
1					2 years	Mean, m...				http://a...Hallenb...	1986.11			1 month	65,000...				Significa...	7,000000	Combin...	CEOSAT		Set...
2				christef@	10 years	CLIMAR	Christos	NTUA		http://a...Hallenb...				3 hours	65,000...				Significa...	7,000000		See WM		Set...
3					5 years	Mean, m...				http://a...Hallenb...	1992.09			1 month	65,000...				Significa...	7,000000	Combin...	TOPEX		Set...

Εικόνα 40 - Φόρμα Επιλογής Περίπτωσης Συνόλου Δεδομένων

Task Specific Information

Prior Execution Info

Input DataSet Properties

NAME	HOSTNAME	UNIFORM RESOURCE IDE	CLASSNAME	ACTION
name0			OCEAN_FEATURE	Get Instance URI

Program Properties

NAME	HOSTNAME	UNIFORM RESOURCE IDENTIFIER	IS_SYNC
name0	my.host	my.un	<input type="checkbox"/>

Output DataSet Properties

NAME	HOSTNAME	UNIFORM RESOURCE IDE	CLASSNAME	ACTION
name1			OCEAN_FEATURE	Get Instance URI

Upon Execution Info

Make combinations of value parameters

Times Run:

Parameter List	Description:
param1	Type of Value: integer
param2	<input type="checkbox"/> Obligatory
param3	Has Constraint Type: Free of Constraints
param4	With Constraint:
param5	Use this param with x :
param6	<input type="checkbox"/> List of Values <input type="radio"/> Step + Initial Value <input type="radio"/> Single Value <input checked="" type="checkbox"/> Do not use it
param7	List
param8	

Step: Initial Value:

Single Value:

Actions

Update Reset

Εικόνα 41 - Περίπτωση Ροής Εργασίας που τρέχει στο background

Στην αντίθετη περίπτωση, αν η περίπτωση ροής εργασίας θα τρέχει στο background, τότε θα εμφανίζεται μια άλλη φόρμα, που φαίνεται στην εικόνα 41, όταν ο χρήστης επιλέγει ένα στοιχείο 'task'. Αυτή η φόρμα στην αρχή θα είναι η ίδια με την προηγούμενη αλλά θα παρουσιάζει και πληροφορία σχετικά με τις τιμές των παραμέτρου εισόδου του εν λόγω στοιχείου. Εκτός από τις πληροφορίες τοποθεσίας για τα σύνολα δεδομένων εισόδου, ο χρήστης θα πρέπει να ορίσει αν το συγκεκριμένο έργο θα τρέξει συγκεκριμένες φορές ή αν οι φορές εκτέλεσής του θα προκύπτουν από το συνδυασμό των τιμών των παραμέτρων του. Έπειτα για κάθε παράμετρο, θα δίνει μια ή περισσότερες τιμές ανάλογα με το τι θέλει να επιτύχει. Οι υποχρεωτικές

παράμετροι πρέπει οπωσδήποτε να πάρουν τιμές (εκτός και αν είναι τύπου ‘no-value’) ενώ οι προαιρετικές παράμετροι δεν είναι υποχρεωτικό να πάρουν τιμές. Για να ολοκληρώσει τον ορισμό του στοιχείου ‘task’, ο χρήστης πρέπει να πατήσει το κουμπί ‘Save’.

Γ.IV. Ένα Παράδειγμα Χρήσης των Κειμενογράφων

Έστω ότι έχουμε την οντολογία του κόσμου των αριθμών (‘world of numbers’ ontology), η οποία βρίσκεται στη διεύθυνση του διαδικτύου <http://www.number-ontology.org>, όπου ορίζεται η οντότητα της ‘ακολουθίας αριθμών’ (number sequence). Η ακολουθία αριθμών είναι ένα σύνολο από αριθμούς. Ο κάθε αριθμός μπορεί να είναι ακέραιος ή δεκαδικός.

Έστω, επίσης, ότι έχουμε ένα πρόγραμμα πρόσθεσης ακολουθίας αριθμών, το οποίο παίρνει ως είσοδο δύο ακολουθίες αριθμών, προσθέτει ένα προς ένα τα στοιχεία τους και παράγει μια νέα ακολουθία ως αποτέλεσμα. Αυτό το πρόγραμμα βρίσκεται στην τοποθεσία διαδικτύου: <http://www.program.org/bin/add.exe>. Τέλος, έστω ότι έχουμε ένα αντίστοιχο πρόγραμμα διαίρεσης, το οποίο παίρνει ως είσοδο δύο ακολουθίες αριθμών και μια παράμετρο που δηλώνει την ακρίβεια της διαίρεσης, διαιρεί ένα προς ένα τα στοιχεία τους και παράγει μια νέα ακολουθία ως αποτέλεσμα. Αυτό το πρόγραμμα βρίσκεται στην διεύθυνση διαδικτύου: <http://www.program.org/bin/divide.exe>. Και τα δύο προγράμματα δεν μπορούν να εκτελεστούν ταυτόχρονα από δύο χρήστες.

Εμείς θέλουμε να προσθέσουμε δύο ακολουθίες αριθμών, να προσθέσουμε άλλες δύο και να διαιρέσουμε τα δύο αποτελέσματα, που προκύπτουν, μεταξύ τους με ακρίβεια δεκαδικών ψηφίων 5, 10 και 15. Οι τέσσερις ακολουθίες αριθμών βρίσκονται στις διευθύνσεις: <http://www.seq1.org>, <http://www.seq2.org>, <http://www.seq3.org> και <http://www.seq4.org>. Οι δύο προσθέσεις μπορούν να γίνουν ταυτόχρονα και μετά θα ακολουθήσει η εκτέλεση της διαίρεσης. Συνεπώς το στοιχείο ‘workflow’ θα αποτελείται από ένα στοιχείο ‘sequence’. Το τελευταίο θα αποτελείται από ένα στοιχείο ‘parallel_sync’ και από ένα στοιχείο ‘task’. Το στοιχείο ‘parallel_sync’ θα αποτελείται από δύο στοιχεία ‘task’.

Για να ικανοποιηθεί η παραπάνω επιθυμία μας, θα πρέπει να δημιουργήσουμε και να αποθηκεύσουμε μια ροή εργασίας και να δημιουργήσουμε μια περίπτωση αυτής της ροής εργασίας που θα σπάσει σε τρεις περιπτώσεις.

Τα βήματα που θα ακολουθηθούν για τον ορισμό της ροής εργασίας στο Κειμενογράφο Ροών Εργασίας είναι τα ακόλουθα:

1. Συμπλήρωση των ‘login’ και ‘password’ πεδίων και πάτημα του κουμπιού ‘Ok’.
2. Πάτημα του κουμπιού ‘New Workflow’.
3. Επιλογή του ριζικού κόμβου του δένδρου και πάτημα του κουμπιού ‘Add a Sequence Element’ της ομάδας κουμπιών ‘Add First-Child Operations’.
4. Επιλογή του κόμβου ‘sequence’ του δένδρου και πάτημα του κουμπιού ‘Add a Parallel_sync Element’ της ομάδας κουμπιών ‘Add First-Child Operations’.
5. Επιλογή του κόμβου ‘parallel_sync’ και πάτημα του κουμπιού ‘Add a Task Element’ της ομάδας κουμπιών ‘Add First-Child Operations’.
6. Πάτημα του κουμπιού ‘Ok’ της φόρμας με όνομα ‘Task Information’.
7. Συμπλήρωση του ονόματος του πρώτου συνόλου δεδομένων εισόδου ως ‘I1’ και προσδιορισμός ότι: ο τύπος του συνόλου δεδομένων είναι ‘general’, το όνομα της

οντολογίας είναι ‘world of numbers’, η διεύθυνση της οντολογίας είναι <http://www.number-ontology.org>, και το όνομα της οντότητας στην οποία ανήκει το σύνολο δεδομένων είναι ‘Number Sequence’. Έπειτα πάτημα του κουμπιού ‘Save’.

8. Πάτημα του κουμπιού ‘New’.
9. Επανάληψη του βήματος 7 μόνο που αλλάζει το όνομα του δεύτερου συνόλου δεδομένων σε ‘I2’.
10. Πάτημα του κουμπιού ‘Continue’ της φόρμας με όνομα ‘Input DataSet Properties’. Επαναπάτημα του κουμπιού ‘Continue’ της νέας φόρμας.
11. Προσδιορισμός του ότι: το όνομα του προγράμματος είναι ‘Add Program’, έχει URI <http://www.program.org/bin/add.exe>, και πάτημα του κουμπιού ‘Ok’.
12. Επανάληψη του βήματος 7 μόνο που γίνεται αλλαγή στο όνομα του συνόλου δεδομένων σε ‘O1’, του τύπου του σε ‘produced-storable’ και του URI του σε <http://www.program.org/bin/O1>.
13. Πάτημα του κουμπιού ‘Continue’ και για τις δύο φόρμες που εμφανίζονται διαδοχικά.
14. Πάτημα του κουμπιού ‘End’.
15. Επιλογή του κόμβου που μόλις δημιουργήθηκε, δηλαδή του κόμβου ‘task’, και πάτημα του κουμπιού ‘Copy’ της ομάδας κουμπιών ‘Edit’.
16. Επιλογή πάλι του κόμβου ‘task’ και πάτημα του κουμπιού ‘Paste As Next Sibling’ της ομάδας κουμπιών ‘Edit’.
17. Επέκταση των κόμβων ‘task’, ‘input’ και ‘output’ του κόμβου που μόλις δημιουργήθηκε.
18. Επιλογή του πρώτου κόμβου ‘data-set’ του στοιχείου ‘input’ του προαναφερόμενου κόμβου ‘task’ και πάτημα του κουμπιού ‘Edit Properties’ της ομάδας κουμπιών ‘Edit’.
19. Αλλαγή του ονόματος του συνόλου δεδομένων ως ‘I3’ και πάτημα του κουμπιού ‘Ok’.
20. Επανάληψη των βημάτων 18 και 19 για το δεύτερο κόμβο ‘data-set’ με τη διαφορά ότι το όνομα θα γίνει ‘I4’.
21. Επανάληψη των βημάτων 18 και 19 για τον κόμβο ‘data-set’ του κόμβου ‘output’ με τη διαφορά ότι το όνομα θα γίνει ‘O2’ και η διεύθυνση <http://www.program.org/bin/O2>.
22. Επιλογή του κόμβου ‘parallel_sync’ και πάτημα του κουμπιού ‘Add a Task Element’ της ομάδα κουμπιών ‘Add a Next-Sibling Element’.
23. Επανάληψη των βημάτων 7 έως 14 με τις εξής αλλαγές: Το όνομα του πρώτου συνόλου δεδομένων εισόδου είναι ‘I5’ και αναφέρεται στο σύνολο δεδομένων ‘O1’, το όνομα του δεύτερου συνόλου δεδομένων είναι ‘I6’ και αναφέρεται στο σύνολο δεδομένων ‘O2’, το όνομα του προγράμματος είναι ‘Divide Program’ και βρίσκεται στη διεύθυνση <http://www.program.org/bin/divide.exe> και το όνομα του παραγόμενου συνόλου δεδομένων είναι ‘O3’ και θα είναι τύπου ‘produced’. Επιπλέον, γίνεται και ο ορισμός δύο παραμέτρων εισόδου. Η πρώτη παράμετρος θα λέγεται ‘accuracy’, θα είναι τύπου ‘Integer’ και θα έχει αρχική τιμή 5. Η δεύτερη παράμετρος θα λέγεται ‘result’ και θα είναι τύπου ‘special’.
24. Πάτημα του κουμπιού ‘Save’ της ομάδας κουμπιών ‘Workflow File Operations’.
25. Απόδοση του ονόματος ‘seq_num.xml’ σε αυτόν τον προσδιορισμό της ροής εργασίας και πάτημα του κουμπιού ‘Save’ της φόρμας ‘Save’.

Τα βήματα που θα ακολουθηθούν αφενός για τη δημιουργία μιας περίπτωσης της προαναφερόμενης ροής εργασίας και αφετέρου για τη εκκίνηση της εκτέλεσης αυτής της περίπτωσης είναι τα ακόλουθα:

1. Πάτημα του κουμπιού 'Execute' της ομάδας 'Workflow File Operations' του Κειμενογράφου Ροών Εργασίας. Συνεπακολούθως, θα εμφανιστεί ο Κειμενογράφος Περιπτώσεων Ροών Εργασίας. Με αυτόν θα αλληλεπιδρούμε τώρα.
2. Προσδιορισμός του ότι η περίπτωση ροής εργασίας θα τρέχει στο background και ότι θα σπάσει σε τόσες περιπτώσεις όσο είναι και το πλήθος των συνδυασμών των φορών εκτέλεσης όλων των έργων της δεδομένης περίπτωσης ροής εργασίας.
3. Επέκταση του γραφικού δένδρου ώστε να εμφανιστούν όλα τα στοιχεία 'task'.
4. Επιλογή του πρώτου στοιχείου 'task' του κόμβου 'parallel_sync'.
5. Συμπλήρωση στη φόρμα που εμφανίζεται, ότι η διεύθυνση του πρώτου συνόλου δεδομένων είναι η: <http://www.seq1.org> και ότι η διεύθυνση του δεύτερου συνόλου δεδομένων είναι η: <http://www.seq2.org> και πάτημα του κουμπιού 'Save'.
6. Επανάληψη των βημάτων 4 και 5 για το δεύτερο 'task' του στοιχείου 'parallel_sync' μόνο που στο πρώτο σύνολο δεδομένων δίνουμε τη διεύθυνση <http://www.seq3.org> και στο δεύτερο τη διεύθυνση <http://www.seq4.org>.
7. Επιλογή του στοιχείου 'task' του κόμβου 'sequence'.
8. Συμπλήρωση στη φόρμα που εμφανίζεται: ότι το έργο δεν θα συνδυάζει τις τιμές των παραμέτρων του και θα εκτελεστεί τρεις φορές, ότι η παράμετρος 'accuracy' θα πάρει τις τιμές 5, 10 και 15 και ότι η παράμετρος 'result' θα πάρει τις τιμές 'res1', 'res2' και 'res3'. Πατάμε το κουμπί 'Save'.
9. Πάτημα του κουμπιού 'Execute' του Κειμενογράφου Περιπτώσεων Ροών Εργασίας.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. C. Houstis and S. Lalis. ARION: An Advanced Lightweight Architecture for Accessing Scientific Collections. *RTD Information Society Technologies*, III.1.4, 2000.
2. C. Houstis and S. Lalis. ARION: A Scalable Architecture for a Digital Library of Scientific Collections. *8th Panhellenic Conference on Informatics*, November 2001.
3. C. Houstis, M. Pitikakis, K. Kritikos, A. Smardas, C. Gikas, L. Sidirourgos. ARION. An Advanced Lightweight System Architecture for Accessing Scientific Collections- System Design. IST Project No 2000-25289.
4. E. Marshall. "A Digital Library Definition", http://www.slais.ubc.ca/courses/libr500/fall1999/www-presentation/e_marshall/digitallibrary-definition.htm
5. Dr. T.B. Rajashakar. "Digital Library and Information Services in Enterprises: Their development and management", <http://144.10.72.183/is214/214-2001-2002/topic-1.h>.
6. M. Marazakis, D. Papadakis, C. Nikolau. The Aurora Architecture for Developing Network-Centric Applications by Dynamic Composition of Services. PhD Thesis at the Computer Science Department-University of Crete, Greece, 2000.
7. A. Sheth, D. Georgakopoulos, S. Joosten, Rusinkievitz, W. Scacchi, J. Wilden and A.Wolf. Report from the NSF Workshop on Workflow and Process Automation in Information Systems. Technical Report, University of Georgia, 1996. Available via URL, <http://LSDIS.cs.uga.edu/publications>.
8. N. Krishnakumar and A. Sheth. Dynamic Configuration of Distributed Systems. *IEEE Transactions Software Engineering*, 11(4), 1985.
9. D. Georgakopoulos, M. Hornik and A. Sheth. An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure. *Distributed and Parallel Databases*, 3(2): 119-154, April 1995.
10. V. Christophides, R. Hull, G. Karvounarakis, A. Kumar, G. Tong, M. Xiong: Beyond Discrete E-services : Composing Session-Oriented Services in Telecommunications. *TES 2001* : 58-73.
11. D.Fensel. *Ontologies: A Silver Bullet for Knowledge Management and E-commerce*. Spinger, 2001.
12. I. Foster, C. Kesselman (eds). *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann, 1998.
13. A. Sidiropoulos. Distributed Mechanisms of Indexing and Search. Master thesis at the Computer Science Department – University of Crete, 1999.
14. Overview of Standard Generalized Markup Language-SGML resources, <http://www.w3.org/MarkUp/SGML>.
15. Marc Standards homepage, <http://www.loc.gov/marc>.
16. International Standard Bibliographic Description of Electronic Resource-ISBD(ER) spec, <http://www.ifla.org/VII/s13/pubs/isbd.html>.
17. Dublic Core homepage, <http://dublincore.org>.
18. Federal Geographic Data Committee-FGDC homepage, <http://www.fgdc.gov>.
19. UDK spec, <http://www.computer.org/conferen/proceed/meta/1999/papers/45/wsoboda.html>.

20. Stanford Digital Library homepage, <http://diglib.stanford.edu>.
21. UC Berkeley Digital Library homepage, <http://http.cs.berkeley.edu/~wilensky>.
22. Ariadne project homepage, <http://www.isi.edu/info-agents/ariadne/index.html>.
23. Alexandria Digital Library homepage, <http://www.alexandria.ucsb.edu>.
24. Master Environmental Library homepage, <http://mel.dmsi.mil>.
25. Earth Observing System (EOS) Data and Information System (EOSDIS) homepage, http://romulus.gsfc.nasa.gov/eosinfo/EOSDIS_Site/index.html.
26. NASA homepage, <http://www.nasa.gov>.
27. DOD homepage, <http://www.unidata.ucar.edu/packages/dods>
28. Mathlab homepage, <http://www.mathworks.html>.
29. Workflow Management Coalition-WFMC homepage, <http://www.wfmc.org>.
30. F. Leymann and D. Roller. *Production Workflow : Concepts and Techniques*. Prentice Hall, 2000.
31. P. Lawrence. *WFMC Workflow Handbook 1997*. John Wiley & Sons, ISBN 0-471-96947, 1997.
32. G. Alonso, D. Agrawal, A. El Abbadi, C. Mohan, M. Kamath and R. Guenthoer. Exotica/FMQM: A Persistent Message-based Architecture for Distributed Workflow Management. In *Proc. IFIP Working Conference on Information Systems Development for Decentralized Organizations*, 1995.
33. S. Ceri, P. Grefen and G. Sanchez. WIDE: A Distributed Architecture for Workflow Management. In *Proc 7th Int'l Workshop on Research Issues in Data Engineering*, 1997.
34. Corba Technology homepage, <http://www.corba.org>.
35. A. Sheth, K.Kochut, J. Miller, D. Worah, S. Das, C. Lin, D. Palaniswami, J. Lynch and I. Schevchenko. Supporting State-Wide Immunization Tracing Using Multi-Paradigm Workflow Technology. In *Proc. VLDB Conference*, 1996.
36. A. Grasso, J. Meunier, D. Pagani and R. Pareschi. Distributed Coordination and Workflow on the World Wide Web. *Computer Supported Cooperative Work*, 6: 175-200, 1997.
37. J. Andreoli, S. Freeman and R. Pareschi. The Coordination Language Facility: Coordination of Distributed Objects. *Theory and Practice of Object Systems*, 2(2), 1996.
38. J. Andreoli, F. Pacull and R. Pareschi. XPECT: A Framework for Electronic Commerce. *IEEE Internet Computing*, 1(4), 1997.
39. J. Weissenfels, D. Wodtke, G. Weikum, A. Kotz Dittrich. The Mentor Architecture for Enterprise-wide Workflow Management. In: *NSF Workshop on Workflow and Process Automation in Information Systems*, Athens, GA, May 1996.
40. D. Georgakopoulos, M. Hornick, A. Sheth. An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure. *Distributed and Parallel Databases*, 3(2), 1995.
41. H. Groiss and J. Eder. *Integrating Workflow Systems and the World Wide Web*. P. Lawrence (ed.) – *Workflow Handbook 1997*. Handbook of the Workflow Management Coalition WFMC., Wiley & Sons, pp 157-163.
42. H. Hammainen, E. Eloranta, J. Alasuvanto. Distributed Form Management. *ACM Transactions on Information Systems*, 8(1), January 1990.
43. C. Bauzer Medeiros, G. Vossen, M. Weske. WASA: A Workflow-Based Architecture to Support Scientific Database and ExpertSystems (Extended Abstract). In *Proc. of 6th International Conference on Database and*

- ExpertSystems Applications (DEXA) 1995*, London, Springer LNCS 978, 574-583, 1995.
44. M. zur Muhlen and J. Becker. Workflow Process Definition Language-Development and Directions of a Meta-Language for Workflow Processes. *Proceedings of the 1999 Workflow Management Conference "Workflow-Based Applications"* in Muenster, November 1999.
 45. S. Carlsen. Conceptual Modeling and Composition of Flexible Workflow Models. PhD Thesis, Information Systems Group. Department of Computer and Information Science, Norwegian University of Science and Technology, Trondheim 1997.
 46. F. Leymann and W. Altenhuber. Managing Business Processes as an Information Resource. *IBM Systems Journal*, 33 (1994) 2, pp. 326-348.
 47. R. Medina-Mora, T. Winograd, R. Flores, F. Flores. The Action Workflow Approach to Workflow Management Technology. In: *CSCW'92: Proceedings of the Conference on Computer Supported Cooperative Work*, ACM Publishers, New York, pp. 281-288.
 48. N.S. Glance, D.S. Pagani, R. Pareschi. Generalized Process Structure Grammars (GPSG) for Flexible Representations of Work. In: M. Ackerman (Ed): *CSCW'96: Proceedings of the Conference on Computer Supported Cooperative Work*. Boston, MA, 1996, pp. 180-189.
 49. P. Senge: *The 5th Discipline-The art and practice of the learning organization*. Century Business Publishers, London, 1990.
 50. O.A. Buhkres, J. Chen, A.K. Elmagarmid, X. Liu and J.G. Mullen. Interbase: A Multi-Database Prototype System. In *Proceeding of the ACM Sigmod Conference on Management of Data*, 1993.
 51. N. Krishnakumar and A. Sheth. Managing Heterogeneous Multi-System Tasks to Support Enterprise-Wide Operations. *Distributed and Parallel Databases*, 3(2), April 1995.
 52. F. Leyman and D. Roller. Business Process Management with FlowMark. In *Proc. 39th IEEE Computer Society Int'l Conf. (CompCon)*, 1994.
 53. The IBM homepage. <http://www.ibm.com>.
 54. D. Chan, J. Vonk, G. Sanchez, P. Grefen, P. Apers. A Specification Language for the WIDE Workflow Model. In *Proc. 7th Int'l Workshop on Advanced Information Systems Engineering (CaiSE)*. 1997.
 55. W.M.P. van der Aalst and A. Kumar. XML Based Schema Definition for Support of Inter-Organizational Workflow. University of Colorado and University of Eindhoven Report, 2001.
 56. W.M.P. van der Aalst. The Application of Petri Nets to Workflow Management. *The Journal of Circuits-Systems and Computer*, 8(1): 21-66, 1998.
 57. N. Adam, V. Atluri and W. Huang. Modeling and Analysis of Workflows using Petri Nets. *Journal of Intelligent Information Systems*, 10(2): 131-158, 1998.
 58. C.A. Ellis and G.J. Nutt. Modelling and Enactment of Workflow Systems. In M. Ajmone Marsan editor, *Application and Theory of Petri Nets 1993*, 691:1-16, 1993.
 59. C. Reisig and G. Rosenberg, editors, *Lecture on Petri Nets I : Basic Models*, Volume 1491 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, 1998.

60. Workflow Definition Language 2,
<http://www.cs.helsinki.fi/group/design/wdl2-toteutus-1.1.html>.
61. Workman – Workflow Manager,
<http://www.cs.helsinki.fi/group/workflow/design>.
62. XLink specification, <http://www.w3.org/TR/xlink/>.
63. World Wide Web Consortium homepage, <http://www.w3.org/>.
64. *Workflow Process Definition Interface – XML Process Definition Language*, a Workflow Management Coalition Specification, Document Number WFMC-TC-1025.
65. Workflow Management Coalition. Workflow Client Application (Interface 2) Application Programming Interface (WAPI) Specification. Document Number WFMC TC-1009. Winchester 1997.
66. T. Malone, K. Crowston, J. Lee, B. Pentland. Tools for Inventing Organizations: Toward a Handbook of Organizational Processes. In: *1993 Proceedings of the 2nd IEEE Workshop on Enabling Technologies Infrastructure for Collaborative*, IEEE Computer Society Press, 1993.
67. J. Lee, M. Gruninger, Y. Jin, T. Malone, A. Tate, G. Yost et. al.. The PIF Process Interchange Format and Framework Version 1.2. *The Knowledge Engineering Review*, 13 (1998) 1, Cambridge University Press, March 1998, pp. 91-120.
68. M. R. Genesereth: Knowledge Interchange Format – draft proposed American National Standard (dpANS). NCITS.T2/98-004. Online Paper as of 1999-09-04. <http://logic.stanford.edu/kif/dpans.html>.
69. National Institute of Standards and Technology-NIST homepage, <http://www.nist.gov/>.
70. International Standardization Organization: Product data representation and exchange. Part 1: Overview and Fundamental Principles. ISO Standard 10303-1, 1992.
71. P. Dourish, J. Holmes, A. MacLean, P. Marquardsen, A. Zbyslaw. Freeflow: Mediating Between Representation and Action in Workflow Systems. In: M. Ackerman (Ed): *CSCW'96: Proceedings of the Conference on Computer Supported Cooperative Work*. Boston, MA, 1996, pp. 190-198.
72. Rational, Inc. et al.: UML Notation Guide version 1.1 OMG Document ad/97-08-05. Framingham, MA 1997.
73. Rational, Inc. et al.: UML Semantics version 1.1. OMG Document ad/97-08-04. Framingham, MA 1997.
74. Rational, Inc. et al.: UML Summary version 1.1. OMG Document ad/97-08-03. Framingham, MA 1997.
75. P. Hruby. Specification of Workflow Management Systems with UML. *Proceedings of the 1998 OOPSLA Workshop on Implementation and Application of Object-oriented Workflow Management Systems*, Vancouver, BC 1998.
76. O. Wiegert. Business Process Modeling and Workflow Definition with UML-Deficiencies and Actions to Improve. Presentation at the OMG Meeting Manchester, 1998-03-31.
77. Ora Lassila and Ralph R. Swick. Resource Description Framework (RDF) Model and Syntax Specification. Technical Report, W3C, 1999. W3C Recommendation. <http://www.w3.org/TR/REC-rdf-syntax>.

78. D. Brickley and R.V. Guha: Resource description framework (RDF) schema specification. Technical Report, W3C, 1999. W3C Proposed Recommendation. <http://www.w3.org/TR/PR-rdf-schema>.
 79. RDFSuite homepage, <http://139.91.183.30:9090/RDF/>.
 80. S. Alexaki, V. Christophides, G. Karvounarakis, D. Plexousakis, K. Tolle. The ICS-FORTH RDFSuite:Managing Voluminous RDF Description Bases. *2nd International Workshop on the Semantic Web, WWW10* (2001).
 81. G. Karvounarakis, S.Alexaki, V. Christophides, D.Plexousakis, M. Scholl. RQL: A Declarative Query Language for RDF. *The Eleventh International World Wide Web Conference (WWW2002)*, Honolulu, Hawaii, USA, May 7-11, 2002.
 82. Java Programming Language of Sun Corporation homepage, <http://java.sun.com>.
 83. RBAC standard homepage, <http://csrc.nist.gov/rbac>.
 84. Geographic Information Systems-GIS homepage, <http://www.gis.com>.
 85. Protégé-2000, <http://protege.stanford.edu>.
 86. OpenGIS, <http://www.opengis.org>.
 87. The XML specification web page, <http://www.w3.org/XML>.
 88. TCP tutorial, <http://www.ietf.org/rfc/rfc1180txt?number=1180>.
 89. Remote Procedure Call-RPC RFC, <http://www.ietf.org/rfc/rfc1150.txt?number=1150>.
 90. SSL Netscape Specification, <http://wp.netscape.com/eng/ssl3/ssl-toc.html>.
 91. Dom4j Java package homepage, <http://www.dom4j.org>.
 92. Xindice – an implementation of a XML Database – homepage, <http://xml.apache.org/xindice>.
 93. XML-DB:XUpdate spec, <http://www.xmldb.org/xupdate/xupdate-wd.html>.
 94. XPath specification, <http://www.w3.org/TR/xpath>.
 95. Vassilis Christophides, Richard Hull, Akhil Kumar: Querying and Splicing of XML Workflows. *CoopIS 2001*: 386-402.
 96. T. R. Gruber. A Translation Approach to Portable Ontology Specifications. In: *Knowledge Acquisition*. Vol.6, no.2, 1993. pp 199-221.
 97. M. Uschold, M. Healy, K. Williamson, P. Clark, S. Woods. Ontology reuse and application. In N. Guarino, (Ed.), *Formal Ontology in Information Systems*, 1998, pages 179-192, Trento, Italy.
 98. XRL, <http://xml.coverpages.org/xrl.html>.
 99. Workflow Management Coalition: Terminology and Glossary. Version 3.0. April 1999. Winchester 1999.
 100. Object Management Group: Workflow Management Facility. Document born 99-03-01. Framingham, MA 1999.
 101. Workflow Management Coalition. Interface 1: Process Definition Interchange Process Model. Document Number WFMC TC-1016-M. Winchester 1998.
 102. R. Bourret: “XML and Databases”. An Internet article.
- K. Nyberg. Workflow Definition Languages. *Tik-76.273 Seminar on Database Management*. Available via: <http://www.cs.hut.fi/~kng/workflowlang/>.