

Εισαγωγή

Περιγραφή

Όλοι οι ερευνητές συνηθίζουν να συλλέγουν λίστες από άρθρα, βιβλία και άλλους τύπους βιβλιογραφικών εγγραφών που τους ενδιαφέρουν και έχουν άμεση επαφή με το ερευνητικό τους πεδίο. Μια από τις πιο συνηθισμένες μορφές στις οποίες αποθηκεύονται αυτά τα δεδομένα είναι αυτή του BibTeX[1]. Μια εγγραφή σε αυτή τη γλώσσα έχει την ακόλουθη μορφή:

```
@Article{Something92,  
  author=      {Something, J.},  
  title=       {The process of Something},  
  journal=     {1971},  
  volume=     {2},  
  number=     {3},  
  pages=      {2-3},  
  month=      {November}  
}
```

Οι κάτοχοι όλων αυτών των εκατομμυρίων βιβλιογραφικών πληροφοριών που υπάρχουν είναι θετικοί στην ιδέα και πρόθυμοι στο να τις διαμοιραστούν με ένα πλήθος συνάδελφων τους που πιθανόν να βρίσκονται σε απόσταση πολλών χιλιομέτρων μακριά αλλά με την προϋπόθεση ότι αυτή η διαδικασία ανταλλαγής των δεδομένων δεν θα συνοδεύεται με την απαίτηση σπατάλης αρκετού χρόνου και εργασίας.

Μια λύση σε αυτό το πρόβλημα προτείνεται από την συγκεκριμένη εργασία που παρουσιάζει μια εφαρμογή στην οποία ο κάθε χρήστης μοιράζεται τα βιβλιογραφικά δεδομένα του μέσα από ένα ομότιμο δίκτυο.

Συγκεκριμένα η εφαρμογή παρέχει τις παρακάτω δυνατότητες :

1. Εισαγωγή, επεξεργασία και προβολή των βιβλιογραφικών δεδομένων μέσα από εύχρηστες και φιλικές προς τον χρήστη φόρμες .Οι παραπάνω διαδικασίες μπορούν να πραγματοποιηθούν με σύντομες κινήσεις χωρίς ιδιαίτερες απαιτήσεις σε προσπάθεια από τον χρήστη-ομότιμο μέλος
2. Εισαγωγή μη επεξεργασμένων δεδομένων. Σε αυτήν την περίπτωση η εφαρμογή μπορεί να διαβάζει βιβλιογραφικές λίστες γραμμένες σε BibTeX και να τις αποθηκεύει κατάλληλα ανάλογα με τον τύπο της κάθε καταχώρησης.

3. Δημοσίευση των δεδομένων στο ομότιμο δίκτυο από κάθε ομότιμο μέλος έτσι ώστε να είναι προσβάσιμα από τα άλλα ομότιμα μέλη που πιθανόν να αναζητήσουν βιβλιογραφικά δεδομένα σχετικά σε περιεχόμενο με αυτά.
4. Το κάθε ομότιμο μέλος δύναται να επιλέξει συγκεκριμένες εγγραφές προς δημοσίευση για το ομότιμο δίκτυο. Σε αυτήν την περίπτωση οι υπόλοιποι χρήστες μπορούν να βλέπουν μόνο αυτές τις εγγραφές.
5. Αναζήτηση βιβλιογραφικών δεδομένων στο ομότιμο δίκτυο βάσει κάποιων περιορισμών-συνθηκών που θέτουν οι χρήστες στοχεύοντας στην εύρεση των επιθυμητών εγγραφών με την μεγαλύτερη ακρίβεια . Από την στιγμή που επιτευχθεί αυτό, το ομότιμο μέλος μπορεί να προσθέσει αυτές στην τοπική του συλλογή για μελλοντική χρήση. Παράλληλα με την αναζήτηση εξάγονται από την εφαρμογή στατιστικές πληροφορίες σχετικές με την κάθε αναζήτηση .

Βάσεις βιβλιογραφικών πληροφοριών στο διαδίκτυο

Μεγάλος αριθμός βιβλιογραφικών πληροφοριών σε μορφή BibTeX είναι διαθέσιμος σε μεγάλες βιβλιογραφικές βάσεις στο διαδίκτυο. Μερικές εξ αυτών είναι:

- **DBLP**(Digital Bibliography & Library Project).Ο εξυπηρετητής DBLP παρέχει τις βιβλιογραφικές πληροφορίες για σημαντικά περιοδικά και πρακτικά πληροφορικής. Αρχικά ο εξυπηρετητής στράφηκε στα συστήματα βάσεων δεδομένων και στον λογικό προγραμματισμό(Logic Programming) (DBLP), τώρα επεκτείνεται βαθμιαία και προς άλλους τομείς της πληροφορικής. Μπορείτε τώρα να διαβάσετε την

Ο εξυπηρετητής συντάσσει ευρετήριο με περισσότερα από 610000 άρθρα και περιέχει αρκετές χιλιάδες συνδέσεις με τις αρχικές σελίδες των επιστημόνων υπολογιστών

<http://www.informatik.uni-trier.de/~ley/db/>

- **ACM**. Η ACM Digital Library είναι μια απέραντη συλλογή παραπομπών και πλήρους κειμένου από ACM άρθρα περιοδικών και ενημερωτικών δελτίων και πρακτικών διασκέψεων.

<http://portal.acm.org/dl.cfm>

- **CSB**.Πρόκειται για μια συλλογή βιβλιογραφιών της επιστημονικής λογοτεχνίας της πληροφορικής από τις διάφορες πηγές, που καλύπτουν τις περισσότερες πτυχές της πληροφορικής. Οι βιβλιογραφίες ενημερώνονται εβδομαδιαία από τις αρχικές θέσεις τους έτσι ώστε να παρέχονται πάντα οι πιο πρόσφατες εκδόσεις.

Η συλλογή περιέχει αυτήν την περίοδο περισσότερες από 2 εκατομμύρια αναφορές (συνήθως άρθρα σε περιοδικά, έγγραφα διασκέψεων και τεχνικές εκθέσεις), που συγκεντρώνονται σε περίπου 1500 βιβλιογραφίες, και αποτελείται από περισσότερα από 2,3 GBytes (530MB) BibTeX_ καταχωρήσεις.

Περισσότερες από 1 εκατομμύριο αναφορές περιέχουν URL για την απευθείας σύνδεση με την έκδοση του εγγράφου. Οι περιλήψεις είναι διαθέσιμες για περισσότερες από 800.000 καταχωρήσεις. Υπάρχουν περισσότερες από 2000 συνδέσεις με άλλες περιοχές που φέρουν βιβλιογραφικές πληροφορίες.

<http://iinwww.ira.uka.de/bibliography/>

- **CiteSeer.** Η CiteSeer είναι μια ψηφιακή βιβλιοθήκη επιστημονικής λογοτεχνίας και παράλληλα μηχανή αναζήτησης που εστιάζει πρώτιστα στη λογοτεχνία των υπολογιστών και την επιστήμη της πληροφορίας. Η CiteSeer στοχεύει να βελτιώσει τη διάδοση και την ανατροφοδότηση της επιστημονικής λογοτεχνίας και να παρέχει βελτιώσεις στην λειτουργικότητα, την χρησιμοποίηση, τη διαθεσιμότητα, το κόστος, την περιεκτικότητα, την αποδοτικότητα, και την επικαιρότητα στην πρόσβαση επιστημονικής και λόγιας γνώσης. Το πρότυπο CiteSeer είναι φορητό με μια πολύ παρόμοια μηχανή αναζήτησης, την SmealSearch, που δημιουργήθηκε για τα ακαδημαϊκά επιχειρησιακά έγγραφα.

<http://citeseer.ist.psu.edu>

Δομή κειμένου

Κεφάλαιο 1:P2P δίκτυα

Στο κεφάλαιο αυτό γίνεται μια παρουσίαση των βασικών αρχών και των οφελών των P2P δικτύων. Παρουσιάζονται οι P2P αρχιτεκτονικές καθώς και οι ιδιότητες που επηρεάζουν την επιλογή της πιο κατάλληλης αρχιτεκτονικής P2P για μια νέα εφαρμογή. Τέλος κατηγοριοποιούνται οι υπάρχουσες P2P εφαρμογές και αναφέρονται κάποια παραδείγματα αυτών.

Κεφάλαιο 2:JXTA

Αρχικά στο κεφάλαιο αυτό γίνεται λόγος στο γιατί θα μπορούσε να επιλεγεί το JXTA ως πλατφόρμα επικοινωνίας σε ένα P2P δίκτυο. Καθορίζεται τι είναι το JXTA, παρουσιάζονται, τα συστατικά, οι βασικές έννοιες και η αρχιτεκτονική δικτύου της τεχνολογίας αυτής.

Κεφάλαιο 3:Java

Στο εν λόγω κεφάλαιο γίνεται μια ιστορική αναδρομή για την γλώσσα προγραμματισμού Java και προβάλλονται τα χαρακτηριστικά και οι πρακτικές της χρήσεις.

Κεφάλαιο 4:BibTeX

Πρόκειται για ένα κεφάλαιο που πέρα από τον καθορισμό της BibTeX μορφής παρουσιάζει την μορφή και τα δυνατά πεδία μιας καταχώρησης αυτής της μορφής καθώς και τους δυνατούς τύπους καταχώρησης.

Κεφάλαιο 5:Παρόμοια Εφαρμογή

Στο συγκεκριμένο κεφάλαιο γίνεται μια σύντομη παρουσίαση μιας παρόμοιας εφαρμογής ,του Bibster.Περιγράφει το σύστημα με τα επιμέρους συστατικά της εφαρμογής αυτής και κάνει μια αναφορά για κάποιες βασικές της λειτουργίες συγκεκριμένα την σημασιολογική εξαγωγή βιβλιογραφικών μεταδεδομένων, την σημασιολογική ζήτηση πληροφοριών, την επιλογή ομότιμου μέλους βασισμένη στην πείρα και την σημασιολογική ανίχνευση διπλότυπης καταχώρησης.

Κεφάλαιο 6:BibSource

Στο τελευταίο κεφάλαιο γίνεται μια εκτενής παρουσίαση της βασικής εφαρμογής BibSource. Αναλύονται όλα τα σκευάσματα που προέκυψαν από τα στάδια της ανάπτυξης της εφαρμογής τον προσδιορισμό απαιτήσεων, την ανάλυση και την σχεδίαση. Τέλος γίνονται κάποιες προτάσεις για την μελλοντική επέκταση/ βελτίωση της εφαρμογής.

Κεφάλαιο 1:P2P Δίκτυα

Εισαγωγή

Το μοντέλο peer-to-peer και των πρακτόρων(agents)[2] προσφέρουν μια λύση για την αναζήτηση και την διανομή πόρων, συχνά χωρίς την ανάγκη μιας κεντρικής εξουσίας ή ενός κεντρικού εξυπηρετητή.

Προς το παρόν , η έμφαση είναι κυρίως πάνω στην διανομή πληροφορίας και υπολογιστικής ισχύος , όμως και άλλα είδη υπηρεσιών θα μπορούσαν να μελετηθούν .Τα P2P συστήματα αποσκοπούν σε εφαρμογές που απαιτούν υψηλό επίπεδο συνεργασίας και επικοινωνίας .Ο στόχος τους είναι να επηρεάσουν θετικά την διαθέσιμη υπολογιστική ισχύς ,το εύρος ζώνης και τις άλλες υπηρεσίες μεταξύ των ομότιμων.

Ο γενικότερος τρόπος με τον οποίο ο όρος agent χρησιμοποιείται, μολονότι δεν υπάρχει αποδεκτός ορισμός στο τι είναι ο agent, είναι για να δηλώσει ένα hardware ή (πιο συνήθως) ένα λογισμικό βασισμένο σε ένα σύστημα υπολογιστών που απολαμβάνει τις ακόλουθες ιδιότητες:

- *Αυτονομία*

Οι agents λειτουργούν χωρίς την άμεση επέμβαση των ανθρώπων ή άλλης εξωτερικής προγραμματιστικά παρέμβασης, και έχουν κάποιο έλεγχο των ενεργειών και της εσωτερικής κατάστασης τους

- *Κοινωνική δυνατότητα(social ability)*

Οι agents αλληλεπιδρούν με άλλους agents (και, ενδεχομένως με ανθρώπους) με την βοήθεια μια επικοινωνιακής γλώσσας των agents.

- *Ικανότητα άμεσου αντιδράσεως(reactivity)*

Οι agents διακρίνουν το περιβάλλον και τις αλλαγές του, και αντιδρούν σε αυτές όπως απαιτείται από τους στόχους τους.

- *Προ-δραστηριότητα(proactivity)*

Οι agents, πέρα από την ικανότητα να αντιδρούν στην αλλαγή των συνθηκών στο περιβάλλον τους ,μπορούν επίσης να αρχικοποιήσουν ενέργειες για να φτάσουν σε ένα ιδιαίτερο στόχο

- *Χρονική συνοχή(temporal continuity)*

Οι agents τρέχουν συνεχώς διαδικασίες (είτε εκτελώντας αυτές σε πρώτο πλάνο είτε στο υπόβαθρο), δεν εκτελούν μόνο τους υπολογισμούς ή τα σενάρια

(scripts) που αντιστοιχίζουν μια απλή είσοδο σε μια απλή έξοδο μια φορά και μετά τερματίζουν.

- *Προσανατολισμός στον στόχο(goal orientedness)*

Ένας agent είναι σε θέση να χειριστεί σύνθετες, υψηλού επιπέδου εργασίες. Η απόφαση πώς μια τέτοια εργασία είναι καλύτερα να διασπαστεί σε μικρότερες δευτερεύουσες εργασίες, και σε ποια σειρά και με ποιόν τρόπο αυτές οι δευτερεύουσες εργασίες είναι καλύτερα να εκτελεστούν, πρέπει να ληφθεί από τον ίδιο τον agent .

Ένας agent παρέχει ένα νέο μοντέλο για την υλοποίηση σύνθετων συστημάτων λογισμικού. Ο όρος χρησιμοποιείται για να περιγράψει συστήματα τα οποία παρουσιάζουν συμπεριφορές οι οποίες συνήθως σχετίζονται με “ενεργές υπάρξεις”. Μια τέτοια συμπεριφορά είναι η ικανότητα να εκτελούνται αυτόνομα ενέργειες σε ένα συγκεκριμένο περιβάλλον σε απόκριση των δραστηριοτήτων άλλων agents σύμφωνα πάντα με ένα συγκεκριμένο σχέδιο.

Το παράδειγμα του agent και το πρότυπο P2P είναι στενά συνδεδεμένες έννοιες: η συνεργασία και η επικοινωνία μεταξύ των ομότιμων κόμβων μπορεί να καθοδηγηθεί από agents που ανήκουν σε κάθε ομότιμο. Οι agents μπορούν να αρχικοποιούν εργασίες εκ μέρους των ομότιμων . Για παράδειγμα έξυπνοι agents μπορούν να χρησιμοποιούνται για να δώσουν προτεραιότητα σε εργασίες σε ένα δίκτυο, να αλλάξουν την ροή της κυκλοφορίας, να αναζητήσουν για αρχεία τοπικά ,να καθορίσουν την ανώμαλη συμπεριφορά και μετά να προβούν σε κατάλληλη ενέργεια, να ανταλλάξουν υπηρεσίες και να συναγωνιστούν για αυτές, να διαπραγματευτούν και τελικά να συνεργαστούν για να πετύχουν ένα ειδικό σκοπό.

Το αρχιτεκτονικό μοντέλο peer-to-peer

Μια ευρεία ποικιλία από ορισμούς του μοντέλου P2P έχουν προταθεί στην βιβλιογραφία. Οι επόμενοι δύο είναι χαρακτηριστικοί αυτών:

Ορισμός που δίνεται από τους συγγραφείς είναι :

-Το P2P μοντέλο αναφέρεται σε μια κλάση από συστήματα και εφαρμογές που χρησιμοποιεί κατανεμημένους πόρους για την εκτέλεση μιας σημαντικής λειτουργίας χωρίς κάποια κεντρική αρχή. Οι πόροι περιλαμβάνουν υπολογιστική ισχύ, δεδομένα (μνήμη και περιεχόμενο), εύρος ζώνης δικτύου, και παρουσία (presence) (υπολογιστές, ανθρώπους, και άλλους πόρους)

Ο ορισμός που έχει δώσει η ομάδα της Intel που ασχολείται με P2P δίκτυα είναι:

-Ο P2P μοντέλο είναι ένα δίκτυο βασισμένο στο μοντέλο αυτό και αφορά εφαρμογές όπου οι υπολογιστές μοιράζονται πόρους και υπηρεσίες μεταξύ τους μέσω της άμεσης ανταλλαγής (via direct exchange).

Οι δύο αυτοί ορισμοί μπορούν να θεωρηθούν σχεδόν ταυτόσημοι. Η διαφορά είναι ότι στον δεύτερο τονίζεται η δυνατότητα διαμοιρασμού των πόρων που υπάρχουν σε κάθε ομότιμο μέλος, ενώ στον πρώτο τονίζεται η δυνατότητα χρήσης των πόρων που προσφέρονται από διάφορα ομότιμα μέλη για την εκτέλεση μιας εργασίας το αποτέλεσμα της οποίας θα προκύψει με συνεργασία όλων αυτών των κόμβων.

Σε αντίθεση στο μοντέλο πελάτη /εξυπηρετητή , ο πελάτης κάνει αιτήσεις στον εξυπηρετητή με τον οποίο είναι δικτυωμένος. Ο εξυπηρετητής ανταποκρίνεται στις αιτήσεις και ενεργεί σε αυτές. Στο P2P μοντέλο , από την άλλη, κάθε υπολογιστής, αναφέρεται ως ένας ομότιμος και λειτουργεί σαν ένας πελάτης αλλά και ως ένας εξυπηρετητής.

Οι P2P εφαρμογές αναπτύσσουν σταδιακά λειτουργίες που αφορούν την αποθήκευση , τους υπολογισμούς, την μεταφορά μηνυμάτων , την ασφάλεια, και την διανομή αρχείων , διαμέσου άμεσων ανταλλαγών μεταξύ των ομότιμων. Ένας ομότιμος μπορεί να στείλει αιτήσεις , αλλά και να ανταποκρίνεται στις αιτήσεις από άλλους ομότιμους στο δίκτυο. Η ικανότητα να κάνεις άμεσες ανταλλαγές με τους άλλους χρήστες ελευθερώνει τους P2P χρήστες από την παραδοσιακή εξάρτηση από τους κεντρικούς εξυπηρετητές. Οι χρήστες έχουν ένα υψηλό βαθμό αυτονομίας και ελέγχου πάνω στις υπηρεσίες που χρησιμοποιούν.

Παρακάτω τονίζονται τα πιθανά οφέλη ενός P2P μοντέλου και περιγράφεται το είδος των P2P αρχιτεκτονικών και των ιδιοτήτων τους.

Οφέλη των P2P

Διάφορα οφέλη προκύπτουν από την P2P δικτύωση , συμπεριλαμβανομένων:

- Μείωση κόστους

Τα παραδοσιακά συστήματα όπως αυτά του πελάτη/ εξυπηρετητή εξυπηρετούν πολλούς πελάτες αλλά το μεγαλύτερο ποσοστό κόστους οφείλεται στο κεντρικό σύστημα. Αυτό το κόστος πιθανόν να είναι απαγορευτικό. Οι P2P αρχιτεκτονικές παρέχουν μια προσέγγιση όπου μοιράζεται το κόστος σε όλους τους ομότιμους κόμβους. Για παράδειγμα , το σύστημα Napster κάνει δυνατή την διανομή του κόστους της αποθήκευσης των αρχείων, αφού κάθε ομότιμο μέλος διατηρεί μια συλλογή αρχείων. Για τον λόγο αυτό, το P2P έχει αντίκτυπο στις συναλλαγές ανάμεσα στις επιχειρήσεις αλλά και ανάμεσα σε επιχειρήσεις και πελάτες. Η παραγωγικότητα αυξάνεται και το κόστος μειώνεται διαμέσου της εκμετάλλευσης των πόρων που προηγουμένως ήταν ανεκμετάλλευτοι.

- Αθροισμα πόρων

Κάθε κόμβος σε ένα P2P σύστημα συνεισφέρει πόρους όπως υπολογιστική ισχύς και χώρο αποθήκευσης. Οι συγκεντρωμένοι πόροι πρακτικά ωφελούν αυτές τις εφαρμογές που απαιτούν μεγάλη υπολογιστική ισχύ όπως για παράδειγμα ,εντατικές προσομοιώσεις ή μεγάλα καταναεμημένα συστήματα αρχείων. Το

ενδεχόμενο των οφελών από την συγκέντρωση πόρων είναι τεράστιο, αλλά αυτό προϋποθέτει να καθίσταται δυνατή η διαλειτουργικότητα μεταξύ των ποικίλων πλατφόρμων υλικού και λογισμικού.

- **Αυτονομία**

Για έλεγχο ή λόγους ασφάλειας, οι χρήστες μπορούν να θέσουν ότι όλα τα δεδομένα και η επεξεργασία σε αυτά πρέπει να εκτελεσθεί τοπικά. Τα P2P συστήματα υποστηρίζουν αυτό το επίπεδο αυτονομίας.

- **Διατήρηση απορρήτου**

Οι χρήστες επίσης μπορεί να θέλουν να διατηρήσουν το απόρρητο τους και να παραμένουν ανώνυμοι στους παροχείς υπηρεσιών. Αυτό είναι δύσκολο σε μια συγκεντρωτική αρχιτεκτονική γιατί ο εξυπηρετητής τυπικά είναι ικανός να αναγνωρίσει τον πελάτη, τουλάχιστον μέσω της διεύθυνσης του διαδικτύου. Οι P2P αρχιτεκτονικές καθιστούν ικανούς τους χρήστες να κρατούν τις πληροφορίες τους τοπικά επειδή όλες οι δραστηριότητες είναι τοπικές. Το FreeNet είναι ένα εξαιρετο παράδειγμα για το πώς η ανωνυμία μπορεί να εξασφαλιστεί σε μια P2P εφαρμογή.

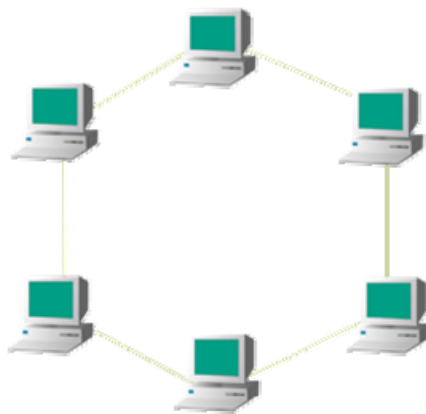
- **Δυναμικότητα**

Τα P2P συστήματα θεωρούν ως δεδομένο ότι το περιβάλλον του μοντέλου P2P είναι υψηλά δυναμικό, πράγμα το οποίο με την σειρά του υπονοεί ότι οι πόροι εισχωρούν στο σύστημα και το αφήνουν συνεχώς. Αυτό το περιβάλλον ταιριάζει πρακτικά στις καταναμημένες εφαρμογές όπου η επέκταση και οι αποκλίσεις/ μεταβολές των απαιτούμενων πόρων δεν είναι προβλέψιμες. Στην άμεση ανταλλαγή μηνυμάτων (Instant Messaging) , για παράδειγμα , οι αποκαλούμενες “λίστες φίλων”(buddy lists) χρησιμοποιούνται για να πληροφορήσουν χρήστες όταν άτομα με τους οποίους αυτοί επιθυμούν να επικοινωνήσουν γίνονται διαθέσιμοι. Χωρίς αυτήν την υποστήριξη , θα απαιτούνταν από τους χρήστες να “ψηφίζουν” για τους συνέταιρους συνομιλίας στέλνοντας τους περιοδικά μηνύματα. Παρομοίως, καταναμημένες εφαρμογές υπολογισμού όπως το distributed.net και το SETI@Home πρέπει να προσαρμόζονται για την αλλαγή των συμμετεχόντων.

P2P αρχιτεκτονικές

Οι ειδικοί τις διακρίνουν μεταξύ του καθαρού(pure) P2P και του υβριδικού(hybrid) P2P εξαρτώμενοι στο κατά πόσον όλοι οι συμμετέχοντες υπολογιστές είναι πράγματι ομότιμοι. Οι δύο παραπάνω κατηγορίες αρχιτεκτονικής P2P χαρακτηρίζονται επιπλέον και ως αποκεντρωτική και ημι-συγκεντρωτική αντίστοιχα. Μια καθαρή P2P ή αποκεντρωτική αρχιτεκτονική δεν περιέχει κανένα κεντρικό σημείο ελέγχου ή εστίασης. Τα ομότιμα μέλη θεωρούνται ότι είναι ισοδύναμα σταθερά και αυτόνομα (δηλαδή ανεξάρτητα, έξυπνα κ.λ.π) .Δεδομένα και υπολογισμός μπορεί να διανέμονται μεταξύ όλων των ομότιμων ,

και οι ομότιμοι μπορούν να επικοινωνούν άμεσα ο ένας με τον άλλο, οπότε και είναι συνεχώς ενήμεροι για την τοπολογία του δικτύου ή έμμεσα μέσω άλλων ομότιμων. Στην τελευταία περίπτωση, η οργάνωση των ομότιμων μπορεί είτε να έχει μια κανονική δομή, όπως μια ιεραρχία, ή να είναι συνολικά αδόμενητη. Μια έμμεση και αδόμενητη αρχιτεκτονική εξαλείφει την ανάγκη για οποιαδήποτε μορφή διαχείρισης που επιβάλλει μια ειδική δικτυακή διαμόρφωση. Ωστόσο η έμφαση και ο φόρτος εργασίας μετατοπίζονται στην αναζήτηση υπηρεσίας επειδή αλλαγές στην τοπολογία του δικτύου μπορούν να συμβούν χωρίς την ενημέρωση όλων των ομότιμων. Το FreeNet είναι ένα παράδειγμα από ένα τέτοιο σύστημα που χρησιμοποιείται για διανομή αρχείων και φτιάχτηκε χρησιμοποιώντας μια έμμεση αδόμενητη αρχιτεκτονική.



Pure P2P

Μια υβριδική P2P ή ημι-συγκεντρωτική αρχιτεκτονική είναι μια αρχιτεκτονική που περιέχει τουλάχιστον ένα κεντρικό σημείο ελέγχου ή εστίασης. Το κεντρικό σημείο μπορεί να χρησιμοποιηθεί για αυστηρό έλεγχο πάνω σε ολόκληρο το δίκτυο ή μπορεί να δρά απλά ως ένα κεντρικό σημείο αναφοράς για τους υπόλοιπους κόμβους. Στην πιο απλή περίπτωση το κεντρικό σημείο είναι ένα απλό ομότιμο μέλος που καταχωρεί σε ένα ευρετήριο (index) τους κόμβους του δικτύου. Το ευρετήριο μπορεί να είναι ένας κατάλογος για τα διαθέσιμα δεδομένα ή τις δυνατότητες επεξεργασίας στο δίκτυο. Ένας τέτοιος κεντρικός κόμβος εξαλείφει την ανάγκη για την αναζήτηση υπηρεσίας. Όταν ένας νέος κόμβος συνδέεται ενημερώνει τον κεντρικό κόμβο για την τοποθεσία και τις δυνατότητες του. Το Napster είναι ένα τυπικό παράδειγμα.

Στα υβριδικά μοντέλα όλοι οι ομότιμοι κόμβοι εκτός από τον κεντρικό εξυπηρετητή μπορεί να είναι είτε συνολικά αυτόνομοι ή χωρίς αυτονομία. Στην τελευταία περίπτωση, τα ομότιμα μέλη ουσιαστικά ελέγχονται από τον κεντρικό κόμβο, όπως στο παράδειγμα SETI@home.

Οι αρχιτεκτονικές με πολλαπλούς εξυπηρετητές ομότιμα μέλη έχουν επίσης ενδιαφέρον. Αυτές έχουν το πλεονέκτημα ότι έχουν υψηλότερη αξιοπιστία γιατί δεν υπάρχει ένα μοναδικό σημείο αποτυχίας. Ακόμα και αν χαλάσει ένα εξυπηρετητής υπάρχουν άλλοι που θα τον αντικαταστήσουν. Ο φόρτος εργασίας του κεντρικού σημείου επίσης διαμοιράζεται σε παραπάνω από ένα συστήματα.

Έχοντας πολλαπλούς κεντρικούς κόμβους θα μπορούσε να αντιμετωπιστεί οποιαδήποτε απώλεια στην ποιότητα της υπηρεσίας (Quality Of Service).



Hybrid P2P

Ιδιότητες των P2P

Η κατάλληλη αρχιτεκτονική P2P εξαρτάται από την εφαρμογή για την οποία θα χρησιμοποιηθεί. Η επιλογή της καθορίζεται από την έκταση στην οποία οι ιδιότητες της αρχιτεκτονικής ταιριάζουν σε αυτές που απαιτούνται από τις εφαρμογές. Μια λίστα από τους παράγοντες που επηρεάζουν την επιλογή της καλύτερης αρχιτεκτονικής P2P φαίνεται παρακάτω:

Η **αξιοπιστία** είναι αναμφισβήτητο το πιο σημαντικό χαρακτηριστικό για σχεδόν όλα τα συστήματα λογισμικού. Οι προσδοκίες ποιότητας των χρηστών κάνουν όχι μεγαλύτερη την ανοχή αποτυχιών λογισμικού σε κανονική βάση. Αναξιόπιστο λογισμικό μπορεί να είναι δαπανηρό στους τελικούς χρήστες και μπορεί να καταλήξει σε χάσιμο στην φήμη των developers του λογισμικού. Ένας ακριβής ορισμός της αξιοπιστίας ωστόσο έχει παραμείνει ασύλληπτος. Η κατακεκομμένη φύση των P2P συστημάτων διογκώνει τα προβλήματα λόγω της χαμηλής αξιοπιστίας. Ένα P2P σύστημα είναι πρακτικά επιρρεπές σε αποτυχίες δικτύου και έτσι μηχανισμοί χρειάζονται για να μειώσουν την επιρροή τους στο σύστημα.

Η **επεκτασιμότητα** είναι η ικανότητα από ένα σύστημα να λειτουργεί χωρίς μια αισθητή πτώση στην επίδοση του παρά την μεταβλητότητα στο ολικό λειτουργικό μέγεθος του. Για παράδειγμα θα μπορούσε ένα σύστημα που αρχικά στόχευε για 10 χρήστες να λειτουργήσει ικανοποιητικά για χιλιάδες χρήστες; Η επεκτασιμότητα με τα P2P συστήματα μετράται κατά μήκος διάφορων παραγόντων, συμπεριλαμβανόμενων για παράδειγμα του αριθμού των χρηστών που χρησιμοποιούν το σύστημα και του αριθμού των κόμβων μέσα στο σύστημα. Επομένως η επεκτασιμότητα θα μπορούσε να παίζει ένα σημαντικό ρόλο όταν σχεδιάζεται ένα σύστημα P2P.

Η **ασφάλεια** εκφράζει το επίπεδο ικανότητας από ένα σύστημα να προστατευτεί από αθέλητη ή θελημένη εισβολή. Η εξασφάλιση της ασφάλειας με τα P2P συστήματα είναι δυσκολότερη από τα πρότυπα συστήματα κεντρικού εξυπηρετητή. Υπάρχει συνήθως μια ανάγκη εξισορρόπησης μεταξύ από την μια, της

πιστοποίησης των συνέταιρων της επικοινωνία και της διανομής των πληροφοριών μόνο με τους ανθρώπους που εμπιστεύεστε και από την άλλη της υποστήριξης της ανωνυμίας των χρηστών.

Η **ακεραιότητα δεδομένων** σημαίνει ότι τα δεδομένα να μην μπορούν να διαφραρθούν ούτε να ακυρωθούν λόγω των προβλημάτων όπως της ασυμφωνίας ή δικτυακών προβλημάτων. Θα μπορούσε ένα τέτοιο πρόβλημα ακεραιότητας δεδομένων να συμβεί. Αυτό θα μπορούσε να οδηγήσει σε αποτυχίες συστημάτων ή λανθασμένες επιχειρηματικές αποφάσεις. Η διατήρηση της ακεραιότητας δεδομένων είναι δυσκολότερη με τα P2P συστήματα λόγω της συχνής ανταλλαγής δεδομένων μεταξύ των ομότιμων. Τα δεδομένα είναι πρακτικά ευαίσθητα στην διαφθορά κατά την διάρκεια της μεταφοράς εξαιτίας της ιδιότητας της αυτονομίας και της πολλαπλής ασυνέπειας των εκδόσεων των δεδομένων που μπορεί να συνυπάρχουν στο σύστημα. Μηχανισμοί αποφάσεων χρειάζονται να καθορίσουν την σωστή έκδοση.

Η **ανωνυμία** στις P2P εφαρμογές είναι η ικανότητα να κρύβεται η ταυτότητα του χρήστη, αλλά χωρίς την επιβαρυντική πιστοποίηση. Η ιδιότητα της ανωνυμίας μπορεί να επεκταθεί και στα δεδομένα. Για να κρυφτεί η ταυτότητα του χρήστη, ψευδώνυμα αντί των IP διευθύνσεων μπορούν να χρησιμοποιηθούν για να αναγνωριστούν οι χρήστες. Μια άλλη προσέγγιση είναι να χαρτογραφηθούν τα δεδομένα επικοινωνίας διαμέσου ανιδιοτελών και εμπιστευτικών τρίτων κοινών πληρεξουσιοτήτων. (party proxies).

Η **εξισορρόπηση φορτίου** παραπέμπει στην ιδιότητα η οποία εξασφαλίζει ότι τα ομότιμα μέλη στο σύστημα δεν λειτουργούν πάρα πολύ ή δεν χρησιμοποιούνται λίγο. Στόχος προφανώς είναι να επιτευχθεί μια αποδοτική χρήση των πόρων στο σύστημα. Για παράδειγμα, σε ένα δικτυακό σύστημα που αποτελείται από διάφορους εξυπηρετητές για να χειρίζονται τις αιτήσεις, το ανατιθέμενο φορτίο είναι εξίσου κατανεμημένο τόσο όσο είναι δυνατόν μεταξύ των εξυπηρετητών.

Η **αναζήτηση ομότιμου** είναι μια σημαντική ιδιότητα ενός P2P συστήματος που αφορά την ικανότητα του ομότιμου μέλους να ανακαλύπτει τους άλλους ομότιμους στο δίκτυο και τις διαθέσιμες υπηρεσίες τους. Τα συστήματα που βασίζονται στις ημι-συγκεντρωτικές αρχιτεκτονικές μπορούν να βασίζονται σε μεγάλο βαθμό σε ένα κεντρικό αρχείο δεικτών θέσης ομότιμου για να παρέχουν τις διευθύνσεις των άλλων ομότιμων στο δίκτυο. Με τις αποκεντρωτικές αρχιτεκτονικές η εργασία της ανακάλυψης διαδίδεται τυπικά κατά μήκος όλων των ομότιμων. Κάθε ομότιμο μέλος διατηρεί την δική του τοπική λανθάνουσα μνήμη με τις γνωστές διευθύνσεις ομότιμων. Η ανακάλυψη γίνεται με την αποστολή μηνυμάτων από το ένα ομότιμο μέλος στον επόμενο μέχρι το ζητούμενο ομότιμο μέλος να βρεθεί.

P2P εφαρμογές

Οι P2P εφαρμογές μπορούν να ταξινομηθούν σε τέσσερις ευρείες κατηγορίες

- την διανομή αρχείων
- την συνεργασία
- τις edge υπηρεσίες , και τέλος
- τον κατανεμημένο υπολογισμό

Παρουσιάζονται παρακάτω αντιπροσωπευτικά παραδείγματα σε κάθε κατηγορία.

Διανομή αρχείων

Οι τρεις πιο δημοφιλείς εφαρμογές διανομής αρχείων περιγράφονται παρακάτω. Αυτές καλύπτουν το φάσμα μεταξύ του καθαρού P2P και του υβριδικού P2P εξαρτώμενες από τον ρόλο των εξυπηρετητών που παίζουν στην εν λόγω αρχιτεκτονική.

Napster. Το Napster είναι ένα φαινόμενο που προκάλεσε πολλή συζήτηση για το μοντέλο P2P. Πρόκειται για ένα λογισμικό που επιτρέπει άμεση ανταλλαγή από MP3 αρχεία και άλλες επιπρόσθετες υπηρεσίες, όπως η άμεση ανταλλαγή μηνυμάτων. Το Napster είναι ένα παράδειγμα από μια κεντροποιημένη P2P αρχιτεκτονική , που μια ομάδα από εξυπηρετητές εκτελούν την διαδικασία ψαξίματος που απαιτείται από τους άλλους ομότιμους. Ένα ομότιμο μέλος πρέπει πρώτα να δημιουργήσει έναν λογαριασμό με τον εξυπηρετητή Napster και να λάβει μια λίστα από ομότιμους που προσφέρουν την λίστα του επιθυμητού αρχείου μουσικής. Μετά αυτός που κάνει την αίτηση μπορεί να επιλέξει οποιοδήποτε από αυτά τα ομότιμα μέλη για να κατεβάσει το αρχείο.

Gnutella. Αυτό είναι ένα παράδειγμα από μια καθαρή P2P εφαρμογή, δηλαδή δεν υπάρχουν εμπλεκόμενοι κεντρικοί εξυπηρετητές. Το Gnutella επιτρέπει διανομή οποιοδήποτε τύπου αρχείων, παρέχει απόλυτη ανωνυμία , είναι ανεκτικό σε σφάλματα σε σχέση με τις αποτυχίες των ομότιμων , με αποτέλεσμα η αξιοπιστία να αυξάνεται , και να προσαρμόζεται καλά στην δυναμική αλλαγή των διαμορφώσεων των ομότιμων. Το Gnutella πρωτόκολλο βασικά επιτρέπει στους συνδρομητές να μορφοποιούν ένα εικονικό δίκτυο. Η συνδρομή εγκαθίστανται με την σύνδεση σε κάποιο αρκετά γνωστό Gnutella host, που παρέχεται από εξειδικευμένους εξυπηρετητές . Ξεκινώντας από τον αρχικό , τα μηνύματα δρομολογούνται σε ένα σύνολο από ομότιμα μέλη χρησιμοποιώντας ένα βεβιασμένο/ αφύσικο μηχανισμό που τα γνωστοποιεί ευρέως μέσα σε ένα πλάτος-ο πρώτος τρόπος ο οποίος περιορίζει το πλήθος των αναπληρωμάτων (hops) μεταξύ των ομότιμων. Αν το μήνυμα που λήφθηκε από ένα ομότιμο μέλος είναι μια αίτηση αναζήτησης για ένα εξειδικευμένο αρχείο, τότε το ομότιμο μέλος ελέγχει τον τοπικό αποθηκευτικό του χώρο και αν το διαθέτει ανταποκρίνεται διαμέσου της ίδιας διαδρομής που ακολουθήθηκε από την αίτηση .

FreeNet. Ο στόχος είναι να παρέχει ένα P2P δίκτυο που είναι σχεδιασμένο να επιτρέπει την διανομή πληροφορίας πάνω από το Διαδίκτυο με έναν αποδοτικό τρόπο και χωρίς λογοκρισία. Το FreeNet είναι ένα παράδειγμα από μια απόλυτα αποκεντρωμένη αρχιτεκτονική. Ένας υπολογιστής στο Διαδίκτυο γίνεται ομότιμο μέλος από ένα freenet δίκτυο απλά τρέχοντας ένα κομμάτι λογισμικού που

καλείται FreeNet εξυπηρετητής ή FreeNet δαίμων (Daemon). Καθώς τα ομότιμα μέλη αλληλεπιδρούν, συσσωρεύουν γνώση σχετικά με τα άλλα ομότιμα μέλη, και αυτή η γνώση μετά επηρεάζεται κατά την διαδικασία της ανακάλυψης. Ωστόσο κανένα ομότιμο μέλος σε οποιοδήποτε σημείο δεν έχει γνώση σχετικά με ολόκληρο το σύστημα, και όλα τα ομότιμα μέλη συμπεριφέρονται ισοδύναμα.

Εκτός από τα παραπάνω δημοφιλή συστήματα, μια πληθώρα άλλων έχουν αναπτυχθεί βασισμένα στο P2P παράδειγμα. Καταγράφουμε σε λίστα μερικά από αυτά εδώ χωρίς εκτενή αναφορά: Akamai[3], FastTrack[4], iMesh[5], OceanStore[6], Pastry[7], Chord[8], CAN[9], JXTA[10], DHT (Distributed Hash Table)[11], Gridella[12], P_Grid[13], Random Walker[14], Farsite [15]κ.α

Συνεργασία

Το P2P μοντέλο εξουσιοδοτεί άτομα και ομάδες να δημιουργήσουν και να διαχειριστούν πραγματικού χρόνου εφαρμογές που συνεργάζονται. Τα εργαλεία συνεργασίας σε αυτήν την περίπτωση στοχεύουν στο ότι οι ομάδες έχουν προσπελάσει τα πιο ενημερωμένα δεδομένα. Απόρροια αυτού του στόχου είναι να βελτιώνεται η παραγωγικότητα μειώνοντας τον χρόνο για πολλαπλές ανασκοπήσεις από τους συμμετέχοντες και τις ομάδες του project σε διαφορετικές γεωγραφικές περιοχές. Οι αυστηρές απαιτήσεις αυτής της εφαρμογής θέτουν μια αξιосέβαστη πρόκληση για την επίτευξη της ασφάλειας και της ακεραιότητας. Οι πρωτοπόροι των συνεργατικών περιβαλλόντων περιλαμβάνουν τα Groove δίκτυα, Endeavors, Eugenia. Όπως με την διανομή αρχείων, η συνεργασία μπορεί να μειώσει την δικτυακή κυκλοφορία αφαιρώντας ηλεκτρονικά μηνύματα και μνήμη εξυπηρετητή που χρειάζεται για την αποθήκευση του project τοπικά. Κάποιες περιοχές επικοινωνίας που έχουν σε μεγάλο βαθμό επηρεαστεί από τα P2P είναι:

-Επικοινωνία: ICQ, IRC, Aimster-Άμεση ανταλλαγή μηνυμάτων (Instant Messaging)

-Διανομή περιεχομένου: διανομή επιχειρησιακού περιεχομένου σε επιτραπέζιους Η/Υ

-Αναζήτηση Διαδικτύου : OpenCOLA (<http://opencola.com/>)-Προσωπικός Διαχειριστής γνώσης (Personal Knowledge Manager)

Edge υπηρεσίες

Μια μορφή του P2P υπολογισμού, γνωστή και ως edge υπηρεσίες, ωθεί περιεχόμενο προς τους ομότιμους των οποίων οι πελάτες μπορεί να βρουν το περιεχόμενο σχετικό, που ελευθερώνεται λόγω του αποθηκευτικού χώρου στους εξυπηρετητές, εξισορροπώντας το φόρτο σε αυτούς τους εξυπηρετητές και διευκολύνοντας την δικτυακή κυκλοφορία σε κρίσιμα κεντροποιημένα συστήματα. Σαν αποτέλεσμα, ο συνολικός διαθέσιμος αποθηκευτικός χώρος αυξάνεται ενώ η διατήρηση του εξυπηρετητή και τα έξοδα του εύρους ζώνης μειώνονται. Αυτό

επίσης έχει μια θετική επιρροή στην παραγωγικότητα των υπαλλήλων, αφού αρχεία και περιεχόμενα διανέμονται γρηγορότερα. Ένα παράδειγμα των πλαισιωμένων υπηρεσιών είναι το Intel Share και το Learn Software (SLS).

Κατανεμημένος υπολογισμός

Το P2P μοντέλο μπορεί να βοηθήσει τις επιχειρήσεις ή τις επιστημονικές εφαρμογές με υπολογιστές μεγάλης κλίμακας αναγκών επεξεργασίας. Χρησιμοποιώντας ένα δίκτυο υπολογιστών, η P2P τεχνολογία μπορεί να χρησιμοποιήσει αδρανείς CPU MIPS και χώρο δίσκου, επιτρέποντας επιχειρήσεις και οργανισμούς να κατανέμουν μεγάλες υπολογιστικές δουλειές διαμέσου πολλαπλών υπολογιστών. Πρόσθετα, τα αποτελέσματα μπορούν να μοιράζονται άμεσα μεταξύ των συμμετεχόντων ομότιμων. Η συνδυασμένη ισχύς των προηγούμενων ανεκμετάλλετων υπολογιστικών πόρων μπορεί εύκολα να ξεπεράσει την κανονικά διαθέσιμη ισχύς από ένα σύστημα επιχείρησης χωρίς κατανεμημένο υπολογισμό. Τα αποτελέσματα είναι γρηγορότερα σε χρόνους ολοκλήρωσης και χαμηλότερα σε κόστος επειδή η τεχνολογία έχει το πλεονέκτημα της διαθέσιμης ισχύς στα συστήματα πελάτη. Παραδείγματα των κατανεμημένων εφαρμογών περιγράφονται στο SETI@home.

Κεφάλαιο 2:JXTA

Γιατί JXTA ;

Δεδομένου ότι το περιεχόμενο και ο αριθμός των δικτυωμένων συσκευών του Παγκόσμιου Ιστού συνεχίζουν να αυξάνονται με ταχύτατους ρυθμούς, το peer-to-peer πρότυπο γίνεται όλο και πιο δημοφιλές. Το λογισμικό που βασίζεται στις P2P τεχνολογίες περιλαμβάνει τη διανομή αρχείων, τον κατανεμημένο υπολογισμό, και τις υπηρεσίες στιγμιαίας ανταλλαγής μηνυμάτων (instant messenger). Κάθε μια από αυτές τις εφαρμογές μπορεί να επιτελεί διαφορετικό έργο, όλες όμως μοιράζονται κάποιες κοινές ιδιότητες, όπως την ανακάλυψη των ομότιμων, την αναζήτηση, και τη μεταφορά αρχείων ή δεδομένων. Αυτήν την περίοδο, η ανάπτυξη εφαρμογών είναι μη αποδοτική, με τους developers να λύνουν τα ίδια προβλήματα και να αντιγράφουν την παρόμοια υποδομή (infrastructure) υλοποίησης. Επιπλέον οι περισσότερες εφαρμογές δημιουργούνται για κάποια συγκεκριμένη πλατφόρμα και δεν έχουν την ικανότητα να επικοινωνήσουν και να μοιραστούν δεδομένα με άλλες εφαρμογές.

- Πρωταρχικός σκοπός του JXTA[10] είναι να παρέχει μια πλατφόρμα με τις βασικές λειτουργίες που είναι απαραίτητες για ένα P2P δίκτυο. Επιπλέον, η τεχνολογία JXTA επιδιώκει να ξεπεράσει τις πιθανές ανεπάρκειες(shortcomings) σε πολλά από τα υπάρχοντα P2P συστήματα.
- *Διαλειτουργικότητα* – Η τεχνολογία JXTA σχεδιάζεται για να καθιστά ικανά τα ομότιμα μέλη ,παρέχοντας τα διάφορες P2P υπηρεσίες ,για να εντοπίζουν και να επικοινωνούν το ένα με το άλλο.
- *Ανεξαρτησία πλατφόρμας* – Η τεχνολογία JXTA σχεδιάζεται για να είναι ανεξάρτητη από τις γλώσσες προγραμματισμού, τα πρωτόκολλα μεταφορών, και τις πλατφόρμες ανάπτυξης.
- *Πανταχού παρουσία(Ubiquity)* – Η τεχνολογία JXTA σχεδιάζεται για να είναι προσβάσιμη από οποιαδήποτε συσκευή με έναν ψηφιακό παλμό(digital heartbeat), όχι μόνο από PCs ή μια συγκεκριμένη πλατφόρμα ανάπτυξης.

Ένα κοινό χαρακτηριστικό των ομότιμων μελών σε ένα P2P δίκτυο είναι ότι συχνά βρίσκονται στην άκρη του κανονικού δικτύου. Επειδή αυτά υπόκεινται στην απρόβλεπτη συνδεσιμότητα με τις ενδεχομένως μεταβλητές διευθύνσεις δικτύων, είναι έξω από το τυποποιημένο πεδίο του dns. Το JXTA φιλοξενεί τα ομότιμα μέλη στην άκρη του δικτύου παρέχοντας ένα σύστημα για τα μοναδικά διεθνοδοτημένα ομότιμα μέλη το οποίο είναι ανεξάρτητο από τις παραδοσιακές υπηρεσίες ονόματος. Μέσω της χρήσης των JXTA IDs, ένα ομότιμο μέλος μπορεί να περιπλανηθεί στα δίκτυα, αλλάζοντας μέσα και διευθύνσεις δικτύου, όντας ακόμη και προσωρινά αποσυνδεδεμένο, και να είναι ακόμα προσπελάσιμο από άλλα ομότιμα μέλη.

Τι είναι το JXTA ;

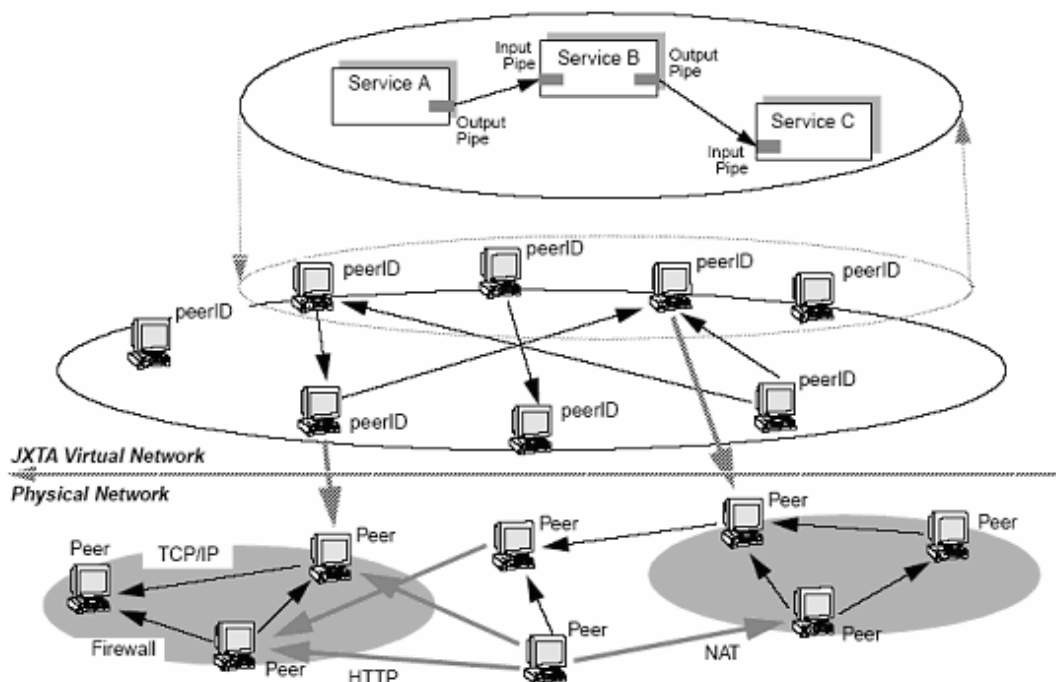
Το JXTA είναι ένα ανοικτό δίκτυο πρότυπο πλατφόρμας που σχεδιάστηκε για το peer-to-peer (P2P) πρότυπο. Στόχος του είναι να αναπτύξει τις βασικές δομικές μονάδες και τις υπηρεσίες για να επιτρέψει καινοτόμες εφαρμογές για τις ομότιμες ομάδες.

Ο όρος "JXTA" είναι σύντομος για αντιπαράθεση, όπως δίπλα-δίπλα. Είναι γνωστό ότι το P2P αντιπαράκειται στο πρότυπο πελάτη- εξυπηρετητή ή βασισμένο στο διαδίκτυο πρότυπο, που είναι το σημερινό παραδοσιακό καταναμημένο πρότυπο επικοινωνίας υπολογιστών.

Το JXTA παρέχει ένα κοινό σύνολο ανοικτών πρωτοκόλλων και μια ανοικτού πηγαιού κώδικα αναφορά υλοποίησης για την ανάπτυξη peer-to-peer εφαρμογών. Τα πρωτόκολλα JXTA τυποποιούν τον τρόπο με τον οποίο τα ομότιμα μέλη:

- Ανακαλύπτουν το ένα το άλλο
- Αυτό-οργανώνονται σε ομότιμες ομάδες
- Διαφημίζονται και ανακαλύπτουν υπηρεσίες δικτύων
- Επικοινωνούν το ένα με το άλλο
- Παρακολουθούν και ελέγχουν το ένα το άλλο

Τα πρωτόκολλα JXTA σχεδιάστηκαν έτσι ώστε να είναι ανεξάρτητα από τις γλώσσες προγραμματισμού και τα πρωτόκολλα μεταφοράς. Τα πρωτόκολλα μπορούν να υλοποιηθούν σε γλώσσα προγραμματισμού Java, C/C ++, Perl, και σε πολλές άλλες γλώσσες. Μπορούν να υλοποιηθούν πάνω από TCP/ IP, HTTP, Bluetooth, HomePNA, ή κάποιο άλλο πρωτόκολλο μεταφοράς.



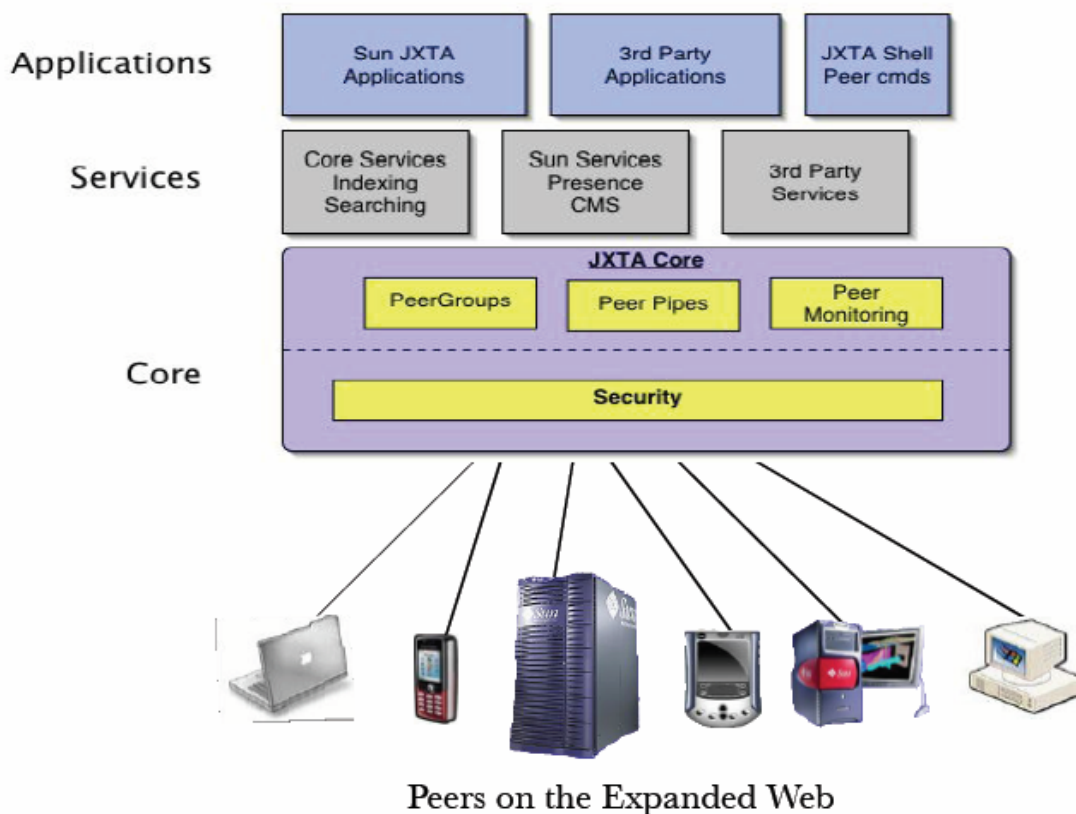
Τι μπορεί να γίνει εφικτό με την τεχνολογία JXTA ;

Τα πρωτόκολλα JXTA επιτρέπουν στους υπεύθυνους για την ανάπτυξη εφαρμογών να υλοποιήσουν και να επεκτείνουν P2P διαλειτουργικές υπηρεσίες και εφαρμογές. Επειδή τα πρωτόκολλα είναι ανεξάρτητα και από τη γλώσσα προγραμματισμού και από τα πρωτόκολλα μεταφοράς, διαφορετικές συσκευές με απολύτως διαφορετικές στοίβες λογισμικού μπορούν να επικοινωνήσουν μεταξύ τους. Χρησιμοποιώντας την τεχνολογία JXTA, οι υπεύθυνοι για την ανάπτυξη εφαρμογών μπορούν να υλοποιήσουν δικτυακές, διαλειτουργικές εφαρμογές που μπορούν να:

- Βρουν άλλα ομότιμα μέλη στο δίκτυο με τη δυναμική ανακάλυψη διαμέσου των firewalls
- Μοιραστούν εύκολα έγγραφα με οποιονδήποτε μέσα στο δίκτυο
- Βρίσκουν μέχρι και το πιο ασήμαντο περιεχόμενο επί των δικτυακών ηλεκτρονικών σελίδων.
- Δημιουργήσουν μια ομάδα ομότιμων μελών η οποία παρέχει μια υπηρεσία
- Παρακολουθούν δραστηριότητες ομότιμων μελών από απόσταση
- Επικοινωνούν με ασφάλεια με άλλα ομότιμα μέλη στο δίκτυο

JXTA Αρχιτεκτονική Επισκόπηση

Η αρχιτεκτονική του λογισμικού JXTA διαιρείται σε τρία επίπεδα, όπως απεικονίζεται στην συνέχεια.



- **Επίπεδο πλατφόρμας (πυρήνας JXTA)**

Το επίπεδο πλατφόρμας, γνωστό και ως πυρήνας JXTA, συμπυκνώνει τις ελάχιστες και ουσιαστικές αρχές (primitives) που είναι κοινές για την P2P δικτύωση. Αυτό περιλαμβάνει τις δομικές μονάδες για να επιτραπούν οι βασικοί μηχανισμοί στις P2P εφαρμογές, συμπεριλαμβανομένων της ανακάλυψης, της μεταφοράς (συμπεριλαμβανομένης της διαχείρισης firewall), της δημιουργίας των ομότιμων μελών και των ομότιμων ομάδων, και των σχετικών αρχών (primitives) ασφάλειας.

- **Επίπεδο υπηρεσιών**

Το επίπεδο υπηρεσιών περιλαμβάνει τις υπηρεσίες δικτύου που μπορεί να μην είναι απολύτως απαραίτητες για την λειτουργία ενός P2P δικτύου, αλλά είναι συνηθισμένες ή επιθυμητές στο P2P περιβάλλον. Παραδείγματα των δικτυακών υπηρεσιών περιλαμβάνουν την αναζήτηση και την δημιουργία ευρετηρίου, κατάλογο(directory), συστήματα αποθήκευσης, διαμοιρασμό αρχείων, καταναμημένα συστήματα αρχείων, συγκέντρωση και ενοικίαση πόρων(aggregation and renting), μετάφραση πρωτοκόλλου(protocol translation), επικύρωση, και PKI (Public Key Infrastructure) υπηρεσίες.

- **Επίπεδο εφαρμογών**

Το επίπεδο εφαρμογών περιλαμβάνει την υλοποίηση των ολοκληρωμένων εφαρμογών, όπως την P2P στιγμιαία ανταλλαγή μηνυμάτων, τον διαμοιρασμό εγγράφων και πόρων, την διαχείριση και παράδοση ψυχαγωγικού περιεχομένου, τα P2P συστήματα ηλεκτρονικού ταχυδρομείου, τα διανεμημένα συστήματα δημοπρασίας, και πολλά άλλα.

Το όριο μεταξύ των υπηρεσιών και των εφαρμογών δεν είναι αυστηρό. Μια εφαρμογή σε έναν πελάτη μπορεί να θεωρηθεί ως μια υπηρεσία σε έναν άλλο πελάτη. Ολόκληρο το σύστημα σχεδιάζεται για να είναι αρθρωτό, επιτρέποντας στους υπεύθυνους για την ανάπτυξη να επιλέξουν μια συλλογή από υπηρεσίες και εφαρμογές που ανταποκρίνονται στις ανάγκες τους.

Συστατικά JXTA

Το δίκτυο JXTA αποτελείται από μια σειρά συνδεδεμένων κόμβων, ή τα **ομότιμα μέλη**. Τα ομότιμα μέλη μπορούν να αυτό-οργανωθούν σε ομάδες ομότιμων μελών, οι οποίες παρέχουν ένα κοινό σύνολο υπηρεσιών. Παραδείγματα των υπηρεσιών που θα μπορούσαν να παρασχεθούν από μια ομάδα ομότιμων μελών περιλαμβάνουν τις εφαρμογές διανομής εγγράφων ή συνομιλίας.

Τα ομότιμα μέλη JXTA διαφημίζουν τις υπηρεσίες τους με έγγραφα XML αποκαλούμενα **διαφημίσεις**. Οι διαφημίσεις επιτρέπουν σε άλλα ομότιμα μέλη στο δίκτυο να μάθουν πώς να συνδέονται και να αλληλεπιδρούν με τις υπηρεσίες ενός ομότιμου μέλους.

Τα ομότιμα μέλη JXTA χρησιμοποιούν τις **διοχετεύσεις (pipes)** για να στείλουν **μηνύματα** το ένα στο άλλο. Οι διοχετεύσεις είναι ένας ασύγχρονος και μονοκατευθυντικός μηχανισμός μεταφοράς μηνύματος που χρησιμοποιείται για

την επικοινωνία σε μια υπηρεσία. Τα μηνύματα είναι απλά έγγραφα XML των οποίων ο φάκελος περιέχει τις πληροφορίες για την δρομολόγηση, την αφομοίωση(digest), και την πιστοποίηση. Οι διοχετεύσεις περιορίζονται από συγκεκριμένα *σημεία τέλους*, όπως μια θύρα(port) TCP και την σχετική διεύθυνση IP.

Αυτές οι έννοιες περιγράφονται λεπτομερώς παρακάτω.

Βασικά χαρακτηριστικά της αρχιτεκτονικής JXTA

Τρία ουσιώδεις χαρακτηριστικά της αρχιτεκτονικής JXTA την διακρίνουν από τα άλλα κατανεμημένα δικτυακά πρότυπα:

- Η χρήση των XML εγγράφων (διαφημίσεις) για την περιγραφή των πόρων
- Αφαίρεση των διοχετεύσεων σε ομότιμα μέλη, και των ομότιμων μελών σε σημεία τέλους(endpoints) χωρίς την εξάρτηση πάνω σε μια κεντρική αρχή ονομασίας/ διευθυνσιοδότησης όπως το DNS
- Ένα ομοιόμορφο σχήμα διευθυνσιοδότησης (ταυτότητες των ομότιμων μελών)

Έννοιες JXTA

Σε αυτό το σημείο καθορίζεται η βασική ορολογία JXTA και περιγράφονται τα αρχικά συστατικά της πλατφόρμας JXTA.

Ομότιμα Μέλη

Ένα *ομότιμο μέλος* είναι οποιαδήποτε δικτυωμένη συσκευή που υλοποιεί ένα ή περισσότερα από τα πρωτόκολλα JXTA. Τα ομότιμα μέλη μπορεί να είναι αισθητήρες, τηλέφωνα, και PDAs, καθώς επίσης και PCs, εξυπηρετητές, και υπερυπολογιστές. Κάθε ομότιμο μέλος λειτουργεί ανεξάρτητα και ασύγχρονα από όλα τα άλλα ομότιμα μέλη, και προσδιορίζεται μοναδικά από μια ταυτότητα ομότιμου μέλους(Peer ID).

Τα ομότιμα μέλη δημοσιεύουν μια ή περισσότερες διεπαφές δικτύου (network interfaces) για τη χρήση των πρωτοκόλλων JXTA. Κάθε δημοσιευμένη διεπαφή διαφημίζεται ως *σημείο τέλους ομότιμου μέλους(peer endpoint)*, το οποίο προσδιορίζει μοναδικά τη διεπαφή δικτύου. Τα σημεία τέλους ομότιμου μέλους χρησιμοποιούνται από τα ομότιμα μέλη για να εγκαταστήσουν τις άμεσες από σημείο σε σημείο συνδέσεις μεταξύ δύο ομότιμων μελών.

Τα ομότιμα μέλη δεν απαιτείται να έχουν άμεσες από σημείο σε σημείο συνδέσεις δικτύων μεταξύ τους. Ενδιάμεσα ομότιμα μέλη μπορούν να χρησιμοποιηθούν για να δρομολογήσουν μηνύματα σε ομότιμα μέλη τα οποία διαχωρίζονται λόγω των φυσικών συνδέσεων δικτύου ή της διαμόρφωσης του δικτύου (π.χ., NATS, firewalls, proxies).

Τα ομότιμα μέλη διαμορφώνονται για να ανακαλύπτουν αυθόρμητα το ένα το άλλο στο δίκτυο και να διαμορφώνουν προσωρινές ή μόνιμες σχέσεις αποκαλούμενες ομάδες ομότιμων μελών (peergroups).

Ομότιμες ομάδες

Μια *ομότιμη ομάδα* είναι μια συλλογή από ομότιμα μέλη που έχουν συμφωνήσει πάνω σε ένα κοινό σύνολο υπηρεσιών. Τα ομότιμα μέλη αυτό-οργανώνονται σε ομότιμες ομάδες, κάθε μια από τις οποίες προσδιορίζεται από μια μοναδική ταυτότητα ομότιμης ομάδας. Κάθε ομότιμη ομάδα μπορεί να καθιερώσει την πολιτική ιδιότητας μέλους της από ανοικτή (οποιοσδήποτε μπορεί να ενωθεί) μέχρι υψηλά αξιόπιστη και προστατευόμενη (αρκετά διαπιστευτήρια /πιστοποιητικά απαιτούνται για να ενωθεί κάποιος).

Τα ομότιμα μέλη πιθανόν να ανήκουν σε περισσότερες από μια ομότιμες ομάδες ταυτόχρονα. Εξ ορισμού, η πρώτη ομάδα που αρχικοποιείται είναι η ομότιμη ομάδα δικτύου(Net Peer Group). Όλα τα ομότιμα μέλη ανήκουν στην ομότιμη ομάδα δικτύου. Τα ομότιμα μέλη μπορούν να επιλέξουν να προσχωρήσουν σε πρόσθετες ομότιμες ομάδες.

Τα πρωτόκολλα JXTA περιγράφουν πώς τα ομότιμα μέλη μπορούν να δημοσιεύσουν, να ανακαλύψουν, να ενωθούν, και να ελέγξουν ομότιμες ομάδες ` δεν υπαγορεύουν πότε ή γιατί οι ομότιμες ομάδες δημιουργούνται.

Υπάρχουν διάφορα κίνητρα για τη δημιουργία ομότιμων ομάδων:

- *Για την δημιουργία ενός ασφαλούς περιβάλλοντος*

Οι ομάδες δημιουργούν μια τοπική περιοχή ελέγχου στην οποία μια συγκεκριμένη πολιτική ασφάλειας μπορεί να επιβληθεί. Η πολιτική ασφάλειας μπορεί να είναι τόσο απλή όσο η ανταλλαγή απλού κειμένου ονόματος χρήστη/ κωδικού πρόσβασης, ή τόσο περίπλοκη όπως η κρυπτογράφηση δημόσιου κλειδιού(public key cryptography). Τα όρια ομότιμης ομάδας επιτρέπουν στα ομότιμα μέλη της να έχουν πρόσβαση και να δημοσιεύουν προστατευμένο περιεχόμενο. Οι ομότιμες ομάδες διαμορφώνουν λογικές περιοχές των οποίων τα όρια περιορίζουν την πρόσβαση στους πόρους της ομότιμης ομάδας.

- *Για την δημιουργία ενός περιβάλλοντος κοινού ενδιαφέροντος*

Οι ομάδες επιτρέπουν την καθιέρωση μιας τοπικής περιοχής ειδίκευσης. Παραδειγματος χάριν, τα ομότιμα μέλη μπορούν να ομαδοποιηθούν μαζί για να υλοποιήσουν ένα έγγραφο που μοιράζεται το δίκτυο ή μια ΚΜΕ που μοιράζεται το δίκτυο. Οι ομότιμες ομάδες εξυπηρετούν την υποδιαίρεση του δικτύου σε αφηρημένες περιοχές που παρέχουν έναν απόλυτο κοινού ενδιαφέροντος μηχανισμό . Τα όρια της ομότιμης ομάδας καθορίζουν το πεδίο αναζήτησης όταν αναζητείται ένα περιεχόμενο της ομάδας.

- *Για την δημιουργία ενός περιβάλλοντος ελέγχου*

Οι ομότιμες ομάδες επιτρέπουν στα ομότιμα μέλη να ελέγχουν ένα σύνολο ομότιμων μελών για οποιοδήποτε ειδικό σκοπό (π.χ., παλμό(heartbeat), αυτοεξέταση κυκλοφορίας(traffic introspection), ή υπευθυνότητα(accountability)).

Οι ομάδες διαμορφώνουν επίσης μια ιεραρχική σχέση γονέα-παιδιού, στην οποία κάθε ομάδα έχει έναν μοναδικό γονέα. Αιτήματα αναζήτησης διαδίδονται μέσα στην ομάδα. Η διαφήμιση για την ομάδα δημοσιεύεται στην ομάδα γονέα πέρα από την ίδια την ομάδα.

Μια ομότιμη ομάδα παρέχει ένα σύνολο υπηρεσιών αποκαλούμενων υπηρεσίες ομότιμης ομάδας. Το JXTA καθορίζει ένα βασικό σύνολο ομότιμων υπηρεσιών ομάδας. Πρόσθετες υπηρεσίες μπορούν να αναπτυχθούν για την παράδοση συγκεκριμένων υπηρεσιών. Για να αλληλεπιδράσουν δύο ομότιμα μέλη μέσω μιας υπηρεσίας, πρέπει και τα δύο να είναι μέρος της ίδιας ομότιμης ομάδας.

Οι βασικές υπηρεσίες ομότιμης ομάδας περιλαμβάνουν τα εξής:

- *Η υπηρεσία ανακάλυψης* –η υπηρεσία ανακάλυψης χρησιμοποιείται από τα ομότιμα μέλη της για την αναζήτηση πόρων ομότιμης ομάδας, όπως ομότιμα μέλη, ομότιμες ομάδες, διασωληνώσεις/ διοχετεύσεις και υπηρεσίες.
- *Η υπηρεσία ιδιότητας μέλους* –η υπηρεσία ιδιότητας μέλους χρησιμοποιείται από τα ενεργά μέλη για να απορρίψουν ή να δεχτούν μια νέα αίτηση ιδιότητας μέλους ομάδας. Τα ομότιμα μέλη που επιθυμούν να προσχωρήσουν σε μια ομότιμη ομάδα πρέπει πρώτα να εντοπίσουν ένα ενεργό μέλος, και να ζητήσουν έπειτα να ενωθούν. Η αίτηση για την ένωση είτε απορρίπτεται είτε γίνεται αποδεκτή από το σύνολο των ενεργών μελών της. Η υπηρεσία ιδιότητας μέλους μπορεί να επιβάλει μια ψηφοφορία ομότιμων μελών ή να εκλέξει μια αντιπροσωπευτική ομάδα για να δεχτεί ή να απορρίψει τις νέες αιτήσεις ιδιότητας μέλους.
- *Η υπηρεσία πρόσβασης* –η υπηρεσία πρόσβασης χρησιμοποιείται για να επικυρώσει αιτήματα που υποβάλλονται από ένα ομότιμο μέλος σε άλλο. Το ομότιμο μέλος που δέχεται την αίτηση παίρνει τα credentials(πιστοποιητικά) των αιτούμενων ομότιμων μελών και τις πληροφορίες σχετικά με το αίτημα που υποβλήθηκε για να καθορίσει εάν η πρόσβαση επιτρέπεται. [Σημείωση:όλες οι ενέργειες μέσα στην ομότιμη ομάδα δεν χρειάζεται να ελεγχθούν με την υπηρεσία πρόσβασης αλλά μόνο εκείνες οι ενέργειες που περιορίζονται σε μερικά ομότιμα μέλη.]
- *Η υπηρεσία διασωλήνωσης* –η υπηρεσία διασωλήνωσης χρησιμοποιείται για να δημιουργήσει και να διαχειριστεί τις συνδέσεις διασωλήνωσης μεταξύ των μελών της ομότιμης ομάδας.
- *Η υπηρεσία αναλυτή (Resolver)* –η υπηρεσία αναλυτή χρησιμοποιείται για να στείλει γενικά ερωτήματα αιτήματα σε άλλα ομότιμα μέλη. Τα ομότιμα μέλη μπορούν να καθορίσουν και να ανταλλάξουν ερωτήματα για να βρουν

οποιοσδήποτε πληροφορίες που μπορεί να χρειάζονται (π.χ., η κατάσταση μιας υπηρεσίας ή η κατάσταση ενός σημείου τέλους διασώληνωσης).

- *Η υπηρεσία ελέγχου* – η υπηρεσία ελέγχου χρησιμοποιείται για να επιτρέψει σε ένα ομότιμο μέλος να ελέγξει άλλα μέλη από την ίδια ομότιμη ομάδα.

Όλες οι παραπάνω υπηρεσίες δεν είναι απαραίτητο να υλοποιηθούν από κάθε ομότιμη ομάδα. Μια ομότιμη ομάδα είναι ελεύθερη να υλοποιήσει μόνο τις υπηρεσίες που βρίσκει χρήσιμες, και στηρίζεται στην εξ ορισμού ομότιμη ομάδα δικτύου για να παρέχει τις γενικές υλοποιήσεις των μη κρίσιμων βασικών υπηρεσιών.

Δικτυακές υπηρεσίες

Τα ομότιμα μέλη συνεργάζονται και επικοινωνούν για να δημοσιεύσουν, να ανακαλύψουν, και να επικαλεσθούν *δικτυακές υπηρεσίες*. Τα ομότιμα μέλη μπορούν να δημοσιεύσουν πολλές υπηρεσίες. Τα ομότιμα μέλη ανακαλύπτουν τις υπηρεσίες δικτύων μέσω του ομότιμου πρωτοκόλλου ανακάλυψης (Peer Discovery Protocol). Τα πρωτόκολλα JXTA αναγνωρίζουν δύο επίπεδα δικτυακών υπηρεσιών:

- *Υπηρεσίες ομότιμου μέλους*

Μια υπηρεσία ομότιμου μέλους είναι προσβάσιμη μόνο στο ομότιμο μέλος που δημοσιεύει αυτήν την υπηρεσία. Εάν αποτύχει αυτό το ομότιμο μέλος, η υπηρεσία αποτυγχάνει επίσης. Πολλά στιγμιότυπα της υπηρεσίας μπορούν να τρέχουν σε διαφορετικά ομότιμα μέλη, αλλά κάθε στιγμιότυπο δημοσιεύει τη διαφήμισή του.

- *Υπηρεσίες ομότιμης ομάδας*

Μια υπηρεσία ομότιμης ομάδας αποτελείται από μια συλλογή στιγμιότυπων (που ενδεχομένως συνεργάζονται το ένα με το άλλο) της υπηρεσίας που τρέχει σε πολλά μέλη της ομότιμης ομάδας. Εάν οποιοσδήποτε ομότιμο μέλος αποτύχει, η συλλογική υπηρεσία ομότιμης ομάδας δεν επηρεάζεται (υποθέτοντας ότι η υπηρεσία είναι ακόμα διαθέσιμη από ένα άλλο ομότιμο μέλος). Οι υπηρεσίες ομότιμης ομάδας δημοσιεύονται ως τμήμα της διαφήμισης της ομότιμης ομάδας.

Οι υπηρεσίες μπορούν είτε να προ-εγκατεστημένες επάνω σε έναν ομότιμο μέλος είτε να φορτώνονται από το δίκτυο. Προκειμένου να τρέξει πραγματικά μια υπηρεσία, ένα ομότιμο μέλος πιθανόν να πρέπει να εντοπίσει μια υλοποίηση κατάλληλη για το περιβάλλον εκτέλεσης του ομότιμου μέλους. Η διαδικασία, της εύρεσης, του κατεβάσματος και της εγκατάστασης μιας υπηρεσίας από το δίκτυο είναι παρόμοια με την εκτέλεση μιας αναζήτησης στο διαδίκτυο για μια ιστοσελίδα, που ανακτά τη σελίδα, και έπειτα εγκαθιστά την απαραίτητη σύνδεση.

Μονάδες -(Modules)

Οι μονάδες JXTA είναι μια αφαίρεση που χρησιμοποιείται για να αντιπροσωπεύσει οποιοδήποτε κομμάτι "κώδικα" που χρησιμοποιείται για να υλοποιήσει μια συμπεριφορά στο κόσμο του JXTA. Οι δικτυακές υπηρεσίες είναι το πιο κοινό παράδειγμα της συμπεριφοράς που μπορεί να αρχικοποιηθεί (instantiated) σε ένα ομότιμο μέλος. Η αφαίρεση μονάδας δεν διευκρινίζει τι είναι αυτός ο "κώδικας" : μπορεί να είναι μια κλάση της Java, ένα jar αρχείο της Java, μια δυναμική βιβλιοθήκη DLL, ένα σύνολο μηνυμάτων XML, ή ένα σενάριο(script). Η υλοποίηση της συμπεριφοράς της μονάδας αφήνεται στους υλοποιητές μονάδων. Παραδείγματος χάριν, οι μονάδες μπορούν να χρησιμοποιηθούν για να αντιπροσωπεύσουν διαφορετικές υλοποιήσεις μιας δικτυακής υπηρεσίας σε διαφορετικές πλατφόρμες, όπως η πλατφόρμα της Java, των Microsoft Windows, ή του λειτουργικού περιβάλλοντος Solaris.

Οι μονάδες παρέχουν μια γενική αφαίρεση που επιτρέπει σε έναν ομότιμο μέλος να αρχικοποιήσει μια νέα συμπεριφορά. Όσο τα ομότιμα μέλη κοιτάζουν βιαστικά ή συνενώνονται σε μια νέα ομότιμη ομάδα, μπορεί να βρουν νέες συμπεριφορές που πιθανόν να θέλουν να αρχικοποιήσουν. Παραδείγματος χάριν, κατά την συνένωση σε μια ομότιμη ομάδα, ένα ομότιμο μέλος μπορεί να πρέπει να μάθει μια νέα υπηρεσία αναζήτησης που χρησιμοποιείται μόνο σε αυτήν την ομότιμη ομάδα. Προκειμένου να ενωθεί σε αυτήν την ομάδα, το ομότιμο μέλος πρέπει να αρχικοποιήσει αυτή την νέα υπηρεσία αναζήτησης. Το πλαίσιο(framework) εργασίας μονάδας καθιστά δυνατή την αντιπροσώπευση και την διαφήμιση των ανεξάρτητων από πλατφόρμα- συμπεριφορών, και επιτρέπει στα ομότιμα μέλη να περιγράψουν και να αρχικοποιήσουν οποιοδήποτε τύπο υλοποίησης μιας συμπεριφοράς. Παραδείγματος χάριν, ένα ομότιμο μέλος έχει τη δυνατότητα να αρχικοποιήσει είτε μια Java είτε μια C υλοποίηση της συμπεριφοράς.

Η δυνατότητα να περιγραφεί και να δημοσιευθεί ανεξάρτητη από πλατφόρμα- συμπεριφορά είναι ουσιαστική για την υποστήριξη ομότιμων ομάδων αποτελούμενων από διαφορετικά ομότιμα μέλη. Οι διαφημίσεις μονάδας επιτρέπουν στα ομότιμα μέλη JXTA να περιγράψουν μια συμπεριφορά με έναν ανεξάρτητο από την πλατφόρμα τρόπο. Η πλατφόρμα JXTA χρησιμοποιεί τις διαφημίσεις μονάδας για να αυτό-περιγραφθεί.

Η αφαίρεση μονάδας περιλαμβάνει μια κλάση μονάδας, μια προδιαγραφή μονάδας, και μια υλοποίηση μονάδας:

- *Κλάση μονάδας (Module Class)*

Η κλάση μονάδας χρησιμοποιείται αρχικά για να διαφημίσει την ύπαρξη μιας συμπεριφοράς. Ο ορισμός της κλάσης αντιπροσωπεύει μια προσδοκώμενη συμπεριφορά και μια προσδοκώμενη δέσμευση για να υποστηρίξει την μονάδα. Κάθε κλάση μονάδας προσδιορίζεται από μια μοναδική ταυτότητα, την ModuleClassID.

- *Προδιαγραφή μονάδας (Module Specification)*

Η προδιαγραφή μονάδας χρησιμοποιείται αρχικά για πρόσβαση σε μια μονάδα. Περιέχει όλες τις απαραίτητες πληροφορίες για πρόσβαση ή ζήτηση της μονάδας. Παραδείγματος χάριν, στην περίπτωση μιας υπηρεσίας, η προδιαγραφή μονάδας μπορεί να περιέχει μια διαφήμιση διασώληνωσης που χρησιμοποιείται για επικοινωνία με την υπηρεσία.

Η προδιαγραφή μιας μονάδας είναι μια προσέγγιση για την παροχή της λειτουργικότητας που μια κλάση μονάδας θεωρεί δεδομένη. Μπορεί να υπάρχουν πολλές προδιαγραφές μονάδας για μια δεδομένη κλάση μονάδας. Κάθε προδιαγραφή μονάδας προσδιορίζεται από μια μοναδική ταυτότητα, την ModuleSpecID. Η ModuleSpecID περιέχει την ταυτότητα της ModuleClass (δηλ., Η ModuleClassID ενσωματώνεται σε μια ModuleSpecID), δείχνοντας τη σχετική κλάση μονάδας.

Η προδιαγραφή μιας μονάδας θεωρεί δεδομένη τη δικτυακή συμβατότητα. Όλες οι υλοποιήσεις μιας δεδομένης προδιαγραφής μονάδας πρέπει να χρησιμοποιούν τα ίδια πρωτόκολλα και να είναι συμβατές, αν και μπορούν να γραφτούν σε διαφορετική γλώσσα.

- *Υλοποίηση μονάδας (Module Implementation)*

Η υλοποίηση μονάδας είναι η υλοποίηση μιας δεδομένης προδιαγραφής μονάδας. Μπορεί να υπάρχουν πολλές υλοποιήσεις μονάδας για μια δεδομένη προδιαγραφή μονάδας. Κάθε υλοποίηση μονάδας περιέχει την ModuleSpecID της σχετικής προδιαγραφής που υλοποιεί.

Οι μονάδες χρησιμοποιούνται από τις υπηρεσίες ομότιμων ομάδων, και μπορούν επίσης να χρησιμοποιηθούν από αυτόνομες υπηρεσίες. Οι υπηρεσίες JXTA μπορούν να χρησιμοποιούν την αφαίρεση μονάδας για να προσδιορίσουν την ύπαρξη της υπηρεσίας (η κλάση μονάδας του), την προδιαγραφή της υπηρεσίας (η προδιαγραφή μονάδας του), ή μια υλοποίηση της υπηρεσίας (μια υλοποίηση μονάδας). Κάθε ένα από αυτά τα συστατικά έχει μια σχετική διαφήμιση, η οποία μπορεί να δημοσιευθεί και να ανακαλυφθεί από τα άλλα JXTA ομότιμα μέλη.

Για παράδειγμα, θεωρείστε την υπηρεσία ανακάλυψης JXTA. Αυτή έχει μια μοναδική ModuleClassID, προσδιορίζοντας την ως μια υπηρεσία ανακάλυψης – η αφηρημένη λειτουργικότητα της. Μπορούν να υπάρχουν πολλαπλές προδιαγραφές της υπηρεσίας ανακάλυψης, κάθε μια ενδεχομένως μη συμβατή με την άλλη. Κάποια μπορεί να χρησιμοποιεί διαφορετικές στρατηγικές προσαρμοσμένες στο μέγεθος της ομάδας και την διασπορά κατά μήκος του δικτύου, ενώ άλλη να πειραματίζεται με νέες στρατηγικές. Κάθε προδιαγραφή έχει μια μοναδική ModuleSpecID, η οποία αναφέρει την υπηρεσία ανακάλυψης ModuleClassID. Για κάθε προδιαγραφή, μπορεί να υπάρχουν πολλές υλοποιήσεις, κάθε μια από τις οποίες περιέχει την ίδια ModuleSpecID.

Συνεπώς , μπορεί να υπάρχουν πολλές προδιαγραφές μιας δεδομένης κλάσης μονάδας, και κάθε μια πιθανόν να μην είναι συμβατή. Εντούτοις, όλες οι εφαρμογές οποιασδήποτε δεδομένης προδιαγραφής θεωρούνται ότι είναι συμβατές.

Διασωληνώσεις (Pipes)

Τα ομότιμα μέλη JXTA χρησιμοποιούν τις *διασωληνώσεις-διοχετεύσεις* για να στείλουν τα μηνύματα το ένα στον άλλο. Οι διασωληνώσεις είναι ένας ασύγχρονος και μονοκατευθυντικός μηχανισμός μεταφοράς μηνύματος που χρησιμοποιείται για την επικοινωνία υπηρεσίας. Όλες οι διασωληνώσεις υποστηρίζουν τη μεταφορά οποιουδήποτε αντικειμένου, συμπεριλαμβανομένων του δυαδικού κώδικα, των αλφαριθμητικών δεδομένων, και των αντικείμενων βασισμένων στην τεχνολογία της Java.

Τα σημεία τέλους των διασωληνώσεων αναφέρονται ως *διασωλήνωση εισαγωγής(input pipe)* (το άκρο λήψης) και *διασωλήνωση εξαγωγής(output pipe)* (το άκρο αποστολής). Τα σημεία τέλους διασωληνώσεων δυναμικά περιορίζονται στα σημεία τέλους του ομότιμου μέλους κατά τον χρόνο εκτέλεσης. Τα σημεία τέλους ομότιμου μέλους αντιστοιχούν στις διαθέσιμες διεπαφές δικτύων ομότιμου μέλους (π.χ., μια πόλη TCP και η σχετική διεύθυνση IP) που μπορούν να χρησιμοποιηθούν για να στείλουν και να λάβουν μήνυμα. Οι διασωληνώσεις JXTA μπορούν να έχουν σημεία τέλους που συνδέονται με διάφορα ομότιμα μέλη σε διαφορετικές στιγμές, ή δεν είναι καθόλου συνδεδεμένα.

Οι διασωληνώσεις είναι εικονικά κανάλια επικοινωνίας και μπορούν να συνδέσουν ομότιμα μέλη που δεν έχουν μια άμεση φυσική σύνδεση. Σε αυτήν την περίπτωση, ένα ή περισσότερα ενδιάμεσα σημεία τέλους ομότιμου μέλους χρησιμοποιούνται για να αναμεταδώσουν τα μηνύματα μεταξύ των δύο σημείων τέλους της διασωλήνωσης .

Οι διασωληνώσεις προσφέρουν δύο τρόπους επικοινωνίας, την από σημείο σε σημείο(point to point) και την διάδοσης(propagate). Ο πυρήνας JXTA παρέχει επίσης αξιόπιστες unicast διασωληνώσεις, μια αξιόπιστη παραλλαγή της από σημείο σε σημείο διασωλήνωσης.

- *Από σημείο σε σημείο διασωλήνωση*

Μια από σημείο σε σημείο διασωλήνωση συνδέει ακριβώς δύο σημεία τέλους διασωλήνωσης μαζί: μια διασωλήνωση εισαγωγής σε έναν ομότιμο μέλος λαμβάνει τα μηνύματα που στάλθηκαν από την διασωλήνωση εξαγωγής ενός άλλου ομότιμου μέλους.

- *Διασωληνώσεις διάδοσης*

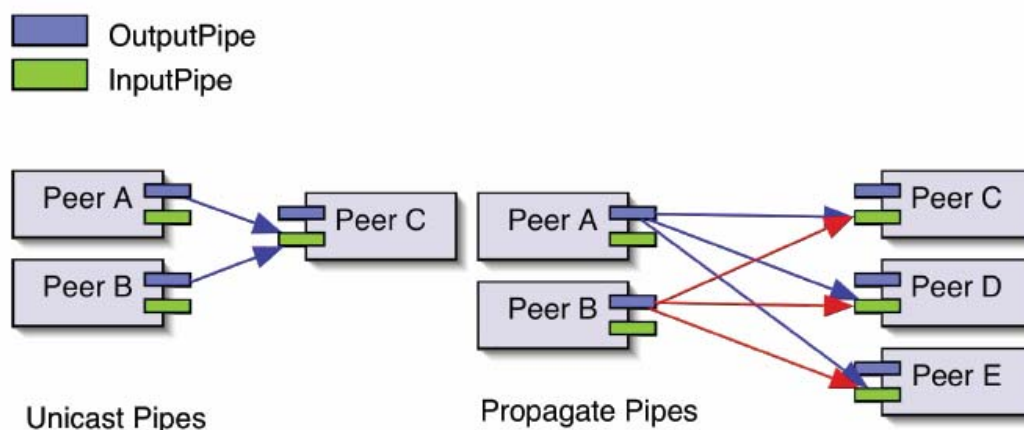
Μια διασωλήνωση διάδοσης συνδέει μια διασωλήνωση εξαγωγής με τις πολλαπλές διασωληνώσεις εισαγωγής. Τα μηνύματα ρέουν από την διασωλήνωση εξαγωγής-παραγωγής (η πηγή διάδοσης) στις διασωληνώσεις

εισαγωγής. Όλη η διάδοση γίνεται στο πλαίσιο μιας ομότιμης ομάδας. Αυτό σημαίνει ότι η διασωλήνωση εξαγωγής και όλες οι διασωληνώσεις εισαγωγής πρέπει να ανήκουν στην ίδια ομότιμη ομάδα.

- *Αξιόπιστες Διασωληνώσεις Unicast*

Μια αξιόπιστη unicast διασωλήνωση είναι ένας τύπος από σημείο σε σημείο διασωλήνωσης που παρέχει ένα αξιόπιστο κανάλι επικοινωνίας.

Πρόσθετοι τύποι υπηρεσιών διασωληνώσεις μπορούν να δημιουργηθούν χρησιμοποιώντας τις βασικές διασωληνώσεις πυρήνα.



Μηνύματα

Το μήνυμα είναι ένα αντικείμενο που στέλνεται μεταξύ των ομότιμων μελών JXTA. Είναι η βασική μονάδα ανταλλαγής δεδομένων μεταξύ των ομότιμων μελών. Τα μηνύματα στέλνονται και παραλαμβάνονται από την υπηρεσία διασωληνώσεων (Pipe Service) και από την υπηρεσία σημείου τέλους (Endpoint Service). Τυπικά, οι εφαρμογές χρησιμοποιούν την υπηρεσία διασωλήνωσης για να δημιουργήσουν, να στείλουν, και να λάβουν μηνύματα. (Γενικά, οι εφαρμογές δεν αναμένεται να χρειάζονται να χρησιμοποιήσουν την υπηρεσία σημείου τέλους άμεσα. Εάν, ωστόσο, μια εφαρμογή χρειάζεται να καταλάβει ή να ελέγξει την τοπολογία του JXTA δικτύου, η υπηρεσία σημείου τέλους μπορεί να χρησιμοποιηθεί.)

Το μήνυμα είναι μια διαταγμένη ακολουθία καθορισμένων και τυποποιημένων περιεχομένων αποκαλούμενων στοιχεία μηνύματος. Κατά συνέπεια ένα μήνυμα είναι ουσιαστικά ένα σύνολο ζευγαριών ονόματος / τιμής. Το περιεχόμενο μπορεί να είναι ένας αυθαίρετος τύπος.

Τα πρωτόκολλα JXTA καθορίζονται ως ένα σύνολο μηνυμάτων που ανταλλάσσονται μεταξύ των ομότιμων μελών. Κάθε δέσμευση (binding) πλατφόρμας λογισμικού περιγράφει πώς ένα μήνυμα μετατρέπεται σε και από μια απλή δομή δεδομένων όπως ένα αντικείμενο τεχνολογίας Java ή μια δομή στην C.

Υπάρχουν δύο αναπαραστάσεις για τα μηνύματα: η XML[16] και η δυαδική(binary). Η JXTA J2SE δέσμευση πλατφόρμας χρησιμοποιεί μια δυαδική μορφή για να ενθυλακώνει το ωφέλιμο φορτίο του μηνύματος. Οι υπηρεσίες μπορούν να χρησιμοποιήσουν το πιο κατάλληλο σχήμα-format για κάποια μεταφορά (π.χ., μια υπηρεσία που απαιτεί μια συμπαγή αναπαράσταση για ένα μήνυμα μπορεί να χρησιμοποιήσει την δυαδική αναπαράσταση, ενώ άλλες υπηρεσίες μπορούν να χρησιμοποιήσουν XML). Τα δυαδικά δεδομένα μπορούν να κωδικοποιηθούν χρησιμοποιώντας ένα σχέδιο κωδικοποίησης Base64 στο σώμα ενός μηνύματος XML.

Η χρήση των μηνυμάτων XML για να καθορίσουν τα πρωτόκολλα επιτρέπει σε πολλά διαφορετικά είδη ομότιμων μελών να συμμετέχουν σε ένα πρωτόκολλο. Επειδή τα δεδομένα είναι σε μορφή ετικετών, κάθε ομότιμο μέλος είναι ελεύθερο να υλοποιήσει το πρωτόκολλο με τρόπο που να ταιριάζει στις ικανότητες και στον ρόλο του. Εάν ένα ομότιμο μέλος χρειάζεται μόνο κάποιο υποσύνολο του μηνύματος, οι ετικέτες δεδομένων XML επιτρέπουν σε αυτό το ομότιμο μέλος να προσδιορίσει τα μέρη του μηνύματος που τον ενδιαφέρουν. Παραδείγματος χάριν, ένα ομότιμο μέλος που περιορίζεται ιδιαίτερα και δεν έχει την επαρκή χωρητικότητα για να επεξεργαστεί μερικό ή το περισσότερο μέρος του μηνύματος, μπορεί να χρησιμοποιήσει τις ετικέτες δεδομένων για να εξαγάγει τα μέρη που μπορεί να επεξεργαστεί, και μπορεί να αγνοήσει το υπόλοιπο.

Διαφημίσεις

Όλοι οι πόροι του δικτύου JXTA –όπως τα ομότιμα μέλη, οι ομότιμες ομάδες, οι διασωληνώσεις, και οι υπηρεσίες – αντιπροσωπεύονται από μια *διαφήμιση*. Οι διαφημίσεις είναι δομές μεταδεδωμένων σε ουδέτερη γλώσσα που αναπαρίστανται ως έγγραφα XML. Τα πρωτόκολλα JXTA χρησιμοποιούν τις διαφημίσεις για να περιγράψουν και να δημοσιεύσουν την ύπαρξη ομότιμων πόρων. Τα ομότιμα μέλη ανακαλύπτουν πόρους ψάχνοντας για τις αντίστοιχες διαφημίσεις τους, και μπορούν να αποθηκεύσουν οποιοσδήποτε ανακαλυμμένες διαφημίσεις τοπικά.

Κάθε διαφήμιση δημοσιεύεται με μια διάρκεια ζωής(lifetime) που διευκρινίζει τη διαθεσιμότητα του σχετικού πόρου της. Οι διάρκειες ζωής επιτρέπουν τη διαγραφή των άχρηστων πόρων χωρίς την απαίτηση οποιουδήποτε κεντρικού ελέγχου. Μια διαφήμιση μπορεί να αναδημοσιευτεί (πριν η αρχική διαφήμιση λήξει) για να επεκτείνει τη διάρκεια ζωής ενός πόρου.

Τα πρωτόκολλα JXTA καθορίζουν τους ακόλουθους τύπους διαφημίσεων:

- *Διαφήμιση ομότιμου μέλους* – περιγράφει το ομότιμο μέλος πόρο. Η αρχική χρήση αυτής της διαφήμισης είναι να κρατήσει συγκεκριμένες πληροφορίες για το ομότιμο μέλος, όπως το όνομα, την ταυτότητα ομότιμου μέλους, τα διαθέσιμα σημεία τέλους του, και οποιοσδήποτε ιδιότητες χρόνου εκτέλεσης(run-time attributes) που οι μεμονωμένες υπηρεσίες ομάδας θέλουν να δημοσιεύσουν (όπως ότι είναι ένα ομότιμο μέλος ραντεβού για την ομάδα).

- *Η διαφήμιση ομότιμης ομάδας* – περιγράφει την ομότιμη ομάδα-συγκεκριμένοι πόροι, όπως το όνομα, την ταυτότητα ομότιμη ομάδας, την περιγραφή, την προδιαγραφή, και τις παραμέτρους υπηρεσίας.
- *Η διαφήμιση διασωλήνωσης* – περιγράφει μια διασωλήνωση κανάλι επικοινωνίας, και χρησιμοποιείται από την υπηρεσία διασωλήνωσης για να δημιουργήσει τα σχετικά σημεία τέλους διασωλήνωσης εισαγωγής και εξαγωγής. Κάθε διαφήμιση διασωλήνωσης περιέχει μια προαιρετική συμβολική ταυτότητα, ένα τύπο διασωλήνωσης (από σημείο σε σημείο, διάδοσης(propagate), αξιοπιστίας(secure), κ.λπ....) και μια μοναδική ταυτότητα διασωλήνωσης.
- *Η διαφήμιση κλάσης μονάδας* – περιγράφει μια κλάση μονάδας. Ο αρχικός σκοπός της είναι τυπικά να τεκμηριώσει την ύπαρξη μιας κλάση μονάδας. Περιλαμβάνει ένα όνομα, την περιγραφή, και την μοναδική ταυτότητα (ModuleClassID).
- *Η διαφήμιση προδιαγραφής μονάδας(Module Spec Advertisement)* – καθορίζει μια προδιαγραφή μονάδας. Κύριος σκοπός της είναι να παρέχει τις αναφορές στην τεκμηρίωση που απαιτείται προκειμένου να δημιουργηθούν οι προσαρμοσμένες υλοποιήσεις αυτής της προδιαγραφής. Μια δευτερεύουσα χρήση είναι, προαιρετικά, να κατασταθούν τα εκτελούμενα στιγμιότυπα χρησιμοποιήσιμα από απόσταση, δημοσιεύοντας πληροφορίες όπως μια διαφήμιση διασωλήνωσης. Περιλαμβάνει το όνομα, την περιγραφή, την μοναδική ταυτότητα (ModuleSpecID), την διαφήμιση διασωλήνωσης, και το πεδίο παραμέτρου που περιέχει τις αυθαίρετες παραμέτρους για να μπορεί να ερμηνευθεί από κάθε εφαρμογή.
- *Η διαφήμιση Υλοποίησης μονάδας* – καθορίζει μια υλοποίηση μιας δεδομένης προδιαγραφής μονάδας. Περιλαμβάνει το όνομα, το σχετικό ModuleSpecID, καθώς επίσης και τον κώδικα, το πακέτο, και τα πεδία παραμέτρων που επιτρέπουν σε έναν ομότιμο μέλος να ανακτήσει τα απαραίτητα δεδομένα για να εκτελέσει την υλοποίηση.
- *Η διαφήμιση ομότιμου μέλους ραντεβού* – περιγράφει έναν ομότιμο μέλος που ενεργεί όπως ένα ομότιμο μέλος ραντεβού για μια δεδομένη ομότιμη ομάδα.
- *Η ομότιμη διαφήμιση πληροφοριών* – περιγράφει τον πόρο πληροφοριών ομότιμου μέλους. Η αρχική χρήση αυτής της διαφήμισης είναι να κρατήσει συγκεκριμένες πληροφορίες για την τρέχουσα κατάσταση ενός ομότιμου μέλους, όπως το πότε συνδέθηκε ,το πλήθος μηνυμάτων εντός και εκτός ορίων του, την χρονική στιγμή που λήφθηκε το τελευταίο μήνυμα και την χρονική στιγμή που στάλθηκε το τελευταίο μήνυμα.

Κάθε διαφήμιση αναπαρίστανται από ένα έγγραφο XML[16]. Οι διαφημίσεις αποτελούνται από μια σειρά ιεραρχικά διατεταγμένων στοιχείων. Κάθε στοιχείο μπορεί να περιλαμβάνει τα δεδομένα του ή πρόσθετα στοιχεία . Ένα στοιχείο μπορεί επίσης να έχει ιδιότητες. Οι ιδιότητες είναι αλφαριθμητικά ζεύγη ονόματος-τιμής. Μια ιδιότητα χρησιμοποιείται για να αποθηκεύσει τα μεταδεδομένα, τα οποία βοηθούν να περιγράψουν τα δεδομένα μέσα στο στοιχείο.

Ένα παράδειγμα μιας διαφήμισης διασωλήνωσης ακολουθεί.

```
<?xml version="1.0"?>
<!DOCTYPE jxta:PipeAdvertisement>
<jxta:PipeAdvertisement xmlns:jxta="http://jxta.org">
<Id>
urn:jxta:uuid-
59616261646162614E504720503250338E3E786229EA460DADC1A176B69B731504
</Id>
<Type>
JxtaUnicast
</Type>
<Name>
TestPipe.end1
</Name>
</jxta: PipeAdvertisement>
```

Η πλήρης προδιαγραφή των διαφημίσεων JXTA δίνεται στην *προδιαγραφή πρωτοκόλλων JXTA (JXTA Protocols Specification)*. Οι υπηρεσίες ή οι υλοποιήσεις ομότιμου μέλους μπορούν να δημιουργήσουν υποκατηγορίες με οποιεσδήποτε από τις ανωτέρω διαφημίσεις για να δημιουργήσουν τις δικές τους διαφημίσεις.

Ασφάλεια

Δυναμικά P2P δίκτυα όπως το δίκτυο JXTA χρειάζεται να υποστηρίξουν τα διαφορετικά επίπεδα πρόσβασης των πόρων. Τα JXTA ομότιμα μέλη λειτουργούν με ένα πρότυπο ρόλο-βασισμένο στην εμπιστοσύνη, στον οποίο ένα μεμονωμένο ομότιμο μέλος δρα κάτω από την εξουσία που εκχωρείται σε αυτό από άλλο έμπιστο ομότιμο μέλος για να εκτελέσει μια ιδιαίτερη εργασία.

Πέντε βασικές απαιτήσεις ασφάλειας πρέπει να παρέχονται:

1. **Εμπιστευτικότητα** – εγγυάται ότι τα περιεχόμενα ενός μηνύματος δεν αποκαλύπτονται σε αναρμόδια άτομα.
2. **Πιστοποίηση** – εγγυάται ότι ο αποστολέας είναι αυτός ο οποίος ή αυτή που ζήτησε να είναι .
3. **Εξουσιοδότηση** – εγγυάται ότι ο αποστολέας εξουσιοδοτείται για να στείλει ένα μήνυμα.

4. **Ακεραιότητα δεδομένων** – εγγυάται ότι το μήνυμα δεν τροποποιήθηκε τυχαία ή σκόπιμα κατά την μεταφορά.
5. **Αντικρουότητα (Refutability)** – εγγυάται ότι το μήνυμα διαβιβάστηκε από έναν κατάλληλα προσδιορισμένο αποστολέα και δεν είναι μια επανάληψη ενός προηγούμενως διαβιβασθέντος μηνύματος. "

Τα μηνύματα XML παρέχουν τη δυνατότητα να προσθέσετε μεταδεδομένα όπως πιστοποιητικά (credentials), βεβαιώσεις(certificates), συνοψίσεις (digests), και δημόσια κλειδιά για μηνύματα JXTA, που επιτρέπουν σε αυτές τις βασικές απαιτήσεις ασφάλειας να ικανοποιηθούν. Οι digests μηνυμάτων εγγυώνται την ακεραιότητα δεδομένων των μηνυμάτων. Τα μηνύματα μπορούν επίσης να κρυπτογραφηθούν (χρησιμοποιώντας δημόσια κλειδιά) και να υπογραφούν (χρησιμοποιώντας τα certificates) για την αξιοπιστία και την αντικρουότητα. Τα credentials μπορούν να χρησιμοποιηθούν για να παρέχουν την εξουσιοδότηση και την πιστοποίηση του μηνύματος.

Ένα credential είναι μια ένδειξη(token) που χρησιμοποιείται για να προσδιορίσει έναν αποστολέα, και μπορεί να χρησιμοποιηθεί για να επικυρώσει το δικαίωμα ενός αποστολέα να στείλει ένα μήνυμα σε ένα συγκεκριμένο σημείο τέλους. Το credential είναι ένα μη διαφανής ένδειξη που πρέπει να παρουσιάζεται κάθε φορά που ένα μήνυμα στέλνεται. Η σταλμένη διεύθυνση που τοποθετείται σε έναν φάκελο JXTA μηνύματος επαληθεύεται με την ταυτότητα του αποστολέα στο credential. Κάθε υλοποίηση credential διευκρινίζεται ως μια αυτόματη(plug-in) διαμόρφωση, η οποία επιτρέπει σε πολλές διαμορφώσεις credential να συνυπάρχουν στο ίδιο δίκτυο.

Είναι πρόθεση των πρωτοκόλλων JXTA να είναι συμβατά με τους ευρέως αποδεκτούς μηχανισμούς ασφάλειας επιπέδου μεταφοράς για τις βασισμένες σε μηνύματα αρχιτεκτονικές, όπως το επίπεδο ασφάλειας υποδοχών (SSL, Secure Sockets Layer) και το πρωτόκολλο ασφάλειας Διαδικτύου (IPSec, Internet Protocol Security). Ωστόσο, τα ασφαλή πρωτόκολλα μεταφοράς όπως το SSL και το IPSec παρέχουν μόνο την ακεραιότητα και την εμπιστευτικότητα της μεταφοράς μηνυμάτων μεταξύ δύο επικοινωνούντων ομότιμων μελών. Προκειμένου να παρέχεται ασφαλής μεταφορά σε ένα δίκτυο πολλών βημάτων (multi-hop) όπως το JXTA, μια σχέση εμπιστοσύνης πρέπει να καθιερωθεί μεταξύ όλων των ενδιαμέσων ομότιμων μελών. Η ασφάλεια επιτυγχάνεται αν όλες οι συνδέσεις επικοινωνίας είναι ασφαλείς.

IDs

Τα ομότιμα μέλη, οι ομότιμες ομάδες, οι διασωληνώσεις και οι άλλοι JXTA πόροι πρέπει να είναι μοναδικά προσδιορισμένοι. Μια ταυτότητα ID JXTA προσδιορίζει μοναδικά μια οντότητα και χρησιμεύει ως ένας κανονιστικός τρόπος για την αναφορά σε αυτή την οντότητα. Έως τώρα , υπάρχουν έξι τύποι οντοτήτων JXTA που έχουν καθορίσει τους τύπους ID JXTA: τα ομότιμα μέλη, η ομότιμη ομάδα, οι

διασωληνώσεις, τα περιεχόμενα, οι κλάσεις μονάδας, και οι προδιαγραφές μονάδας.

Τα URNs χρησιμοποιούνται για να εκφράσουν τα JXTA IDs. Τα URNs είναι μια μορφή του URI που "... προορίζονται να χρησιμεύσουν ως συνεχείς, ανεξάρτητοι θέσης, προσδιοριστές πόρων ". Όπως και οι άλλες μορφές του URI, τα JXTA IDs παρουσιάζονται ως κείμενο.

Ένα παράδειγμα ID JXTA ομότιμου μέλους είναι:

```
urn:jxta:uuid-  
59616261646162614A78746150325033F3BC76FF13C2414CBC0AB663666DA53903
```

Ένα παράδειγμα ID JXTA διασωλήνωσης είναι:

```
urn:jxta:uuid-  
59616261646162614E504720503250338E3E786229EA460DADC1A176B69B731504
```

Μοναδικά IDs παράγονται τυχαία από τη J2SE δέσμευση(binding) πλατφόρμας JXTA. Υπάρχουν δύο εφεδρικές JXTA IDs: η μηδενική (NULL) ταυτότητα και η ID ομότιμης ομάδας δικτύου(Net Peer Group).

Αρχιτεκτονική Δικτύου

Οργάνωση δικτύων

Το δίκτυο JXTA είναι ένα ειδικό, πολλών βημάτων(multi-hop) , προσαρμόσιμο δίκτυο που αποτελείται από τα συνδεδεμένα ομότιμα μέλη. Συνδέσεις μέσα στο δίκτυο μπορεί να είναι προσωρινές, και η δρομολόγηση μηνυμάτων μεταξύ των ομότιμων μελών είναι μη ντετερμινιστική. Τα ομότιμα μέλη μπορεί να ενώνονται ή να αφήνουν το δίκτυο από την μια στιγμή στην άλλη, με αποτέλεσμα οι διαδρομές να αλλάζουν συχνά.

Τα ομότιμα μέλη μπορούν να πάρουν οποιαδήποτε μορφή εφ' όσον μπορούν να επικοινωνήσουν χρησιμοποιώντας τα πρωτοκόλλα JXTA. Η οργάνωση του δικτύου δεν υποχρεούται από το πλαίσιο εργασίας JXTA, αλλά τυπικά στην πράξη τέσσερα είδη ομότιμων μελών είναι που χρησιμοποιούνται:

- *Ελάχιστο ομότιμο μέλος άκρου*

Ένα ελάχιστο ομότιμο μέλος άκρου μπορεί να στείλει και να λάβει μηνύματα, αλλά δεν εναποθηκεύει τις διαφημίσεις ή τη διαδρομή των μηνυμάτων για άλλα ομότιμα μέλη. Τα ομότιμα μέλη στις συσκευές με περιοριστικούς πόρους (π.χ., ένα PDA ή ένα κινητό τηλέφωνο) θα μπορούσαν να είναι τα ελάχιστα ομότιμα μέλη άκρου.

- *Πλήρως-χαρακτηρισμένο ομότιμο μέλος άκρου*

Ένα πλήρως-χαρακτηρισμένο ομότιμο μέλος μπορεί να στείλει και να λάβει μηνύματα, και τυπικά να εναποθηκεύσει τις διαφημίσεις. Ένα απλό ομότιμο μέλος απαντά στα αιτήματα ανακάλυψης με πληροφορίες που βρήκε στις εναποθηκευμένες διαφημίσεις του, αλλά δεν διαβιβάζει οποιαδήποτε αιτήματα ανακαλύψεων. Τα περισσότερα ομότιμα μέλη συνήθως είναι ομότιμα μέλη άκρου.

- *Ομότιμο μέλος ραντεβού*

Ένα ομότιμο μέλος ραντεβού είναι όπως οποιοδήποτε άλλο ομότιμο μέλος, και διατηρεί μια περιοχή αποθήκευσης για τις διαφημίσεις. Ωστόσο, τα ομότιμα μέλη ραντεβού επιπλέον προωθούν τις αιτήσεις ανακάλυψης για να βοηθήσουν τα άλλα ομότιμα μέλη κατά την ανακάλυψη πόρων. Όταν ένα ομότιμο μέλος προσχωρεί/ ενώνεται σε μια ομότιμη ομάδα, ψάχνει αυτόματα έναν ομότιμο μέλος ραντεβού. Εάν κανένα ομότιμο μέλος ραντεβού δεν βρεθεί, αυτό το ομότιμο μέλος δυναμικά γίνεται ένα ομότιμο μέλος ραντεβού για αυτήν την ομότιμη ομάδα. Κάθε ομότιμο μέλος ραντεβού διατηρεί μια λίστα από τα άλλα γνωστά ομότιμα μέλη ραντεβού και κατά συνέπεια και τα ομότιμα μέλη που το χρησιμοποιούν ως ομότιμο μέλος ραντεβού.

Κάθε ομότιμη ομάδα διατηρεί το δικό της σύνολο ομότιμων μελών ραντεβού, και μπορεί να έχει τόσα ομότιμα μέλη ραντεβού όσα χρειάζονται. Μόνο τα ομότιμα μέλη ραντεβού που είναι μέλη μιας ομότιμης ομάδας θα δουν τα αιτήματα αναζήτησης της συγκεκριμένης ομότιμης ομάδας.

Τα ομότιμα μέλη άκρου στέλνουν τα αιτήματα αναζήτησης και ανακάλυψης στα ομότιμα μέλη ραντεβού, τα οποία με την σειρά τους προωθούν τα αιτήματα, όταν αυτά δεν μπορούν να τα απαντήσουν, σε άλλα γνωστά ομότιμα μέλη ραντεβού. Η διαδικασία ανακαλύψεων συνεχίζεται έως ότου ένα ομότιμο μέλος έχει την απάντηση ή το αίτημα πεθάνει. Τα μηνύματα έχουν ένα προεπιλεγμένο χρόνο ζωής(TTL, time-to-live) επτά βημάτων (hops).Οι αναδράσεις(Loopbacks)αποτρέπονται με τη διατήρηση λίστας των ομότιμων μελών κατά μήκος της πορείας του μηνύματος .

- *Ομότιμο μέλος αναμετάδοσης(relay peer or router peer)*

Ένα ομότιμο μέλος αναμετάδοσης διατηρεί τις πληροφορίες σχετικά με τις διαδρομές σε άλλα ομότιμα μέλη και δρομολογεί τα μηνύματα στα ομότιμα μέλη. Ένα ομότιμο μέλος πρώτα κοιτάζει στην τοπική περιοχή αποθήκευσης του για τις πληροφορίες διαδρομής. Εάν δεν βρεθούν, το ομότιμο μέλος στέλνει τις ερωτήσεις στα ομότιμα μέλη αναμετάδοσης ζητώντας πληροφορίες διαδρομής. Τα ομότιμα μέλη αναμετάδοσης έτσι προωθούν τα μηνύματα εκ μέρους των ομότιμων μελών που δεν μπορούν άμεσα να απευθυνθούν σε ένα άλλο ομότιμο μέλος (π.χ., NAT περιβάλλοντα), γεφυρώνοντας τα διαφορετικά φυσικά και/ ή λογικά δίκτυα. Οποιοδήποτε ομότιμο μέλος μπορεί να υλοποιήσει τις υπηρεσίες που απαιτούνται για να είναι ένα ομότιμο μέλος

αναμετάδοσης ή ραντεβού. Οι υπηρεσίες αναμετάδοσης και ραντεβού μπορούν να υλοποιηθούν στο ίδιο ομότιμο μέλος.

Διανεμημένο Ευρετήριο Κοινού πόρου(SRDI, Shared Resource Distributed Index)

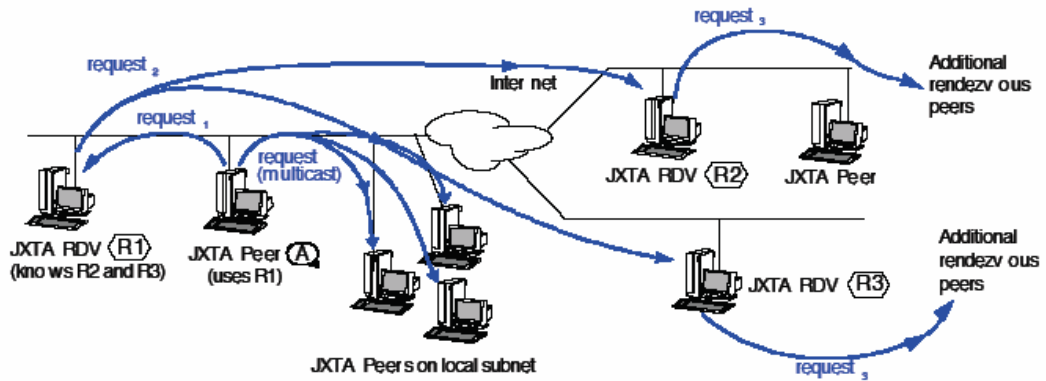
Η JXTA 2.0 J2SE πλατφόρμα υποστηρίζει μια υπηρεσία ενός διανεμημένου ευρετηρίου κοινού πόρου (SRDI) για να παρέχει ένα περισσότερο αποδοτικό μηχανισμό για τα προς διάδοση αιτήματα ερώτησης μέσα στο δίκτυο JXTA. Τα ομότιμα μέλη ραντεβού διατηρούν ένα ευρετήριο των διαφημίσεων που δημοσιεύτηκαν από τα ομότιμα μέλη άκρων. Όταν τα ομότιμα μέλη άκρων δημοσιεύουν νέες διαφημίσεις, χρησιμοποιούν την υπηρεσία SRDI για να ωθήσει τους δείκτες διαφημίσεων στα ομότιμα μέλη ραντεβού τους. Με αυτήν την ιεραρχία ομότιμου μέλους ραντεβού- ομότιμου μέλους άκρου, οι ερωτήσεις διαδίδονται μεταξύ μόνο των ομότιμων μελών ραντεβού, πράγμα το οποίο μειώνει σημαντικά τον αριθμό ομότιμων μελών που συμμετέχουν στην αναζήτηση μιας διαφήμισης.

Κάθε ομότιμο μέλος ραντεβού διατηρεί ένα δικό του κατάλογο με τα γνωστά ομότιμα μέλη ραντεβού του στην ομότιμη ομάδα. Ένα ομότιμο μέλος ραντεβού μπορεί να ανακτήσει πληροφορίες ραντεβού από ένα προκαθορισμένο σύνολο αυτοδύναμης έναρξης(bootstrapping), ή προκυπτουσών (seeding) ομότιμων μελών ραντεβού. Τα ομότιμα μέλη ραντεβού περιοδικά επιλέγουν έναν δεδομένο τυχαίο αριθμό ομότιμων μελών ραντεβού και τους στέλνουν μια τυχαία λίστα γνωστών τους ομότιμων μελών ραντεβού. Τα ομότιμα μέλη ραντεβού επίσης περιοδικά καθαρίζουν τα ομότιμα μέλη ραντεβού που δεν ανταποκρίνονται. Κατά συνέπεια, διατηρούν ένα ασαφή-συνεπές δίκτυο των γνωστών ομότιμων μελών ραντεβού.

Όταν ένας ομότιμο μέλος δημοσιεύει μια νέα διαφήμιση, η διαφήμιση τοποθετείται στο ευρετήριο από την υπηρεσία SRDI χρησιμοποιώντας κλειδιά όπως το όνομα ή την ταυτότητα της διαφήμισης. Μόνο οι δείκτες της διαφήμισης ωθούνται στα ομότιμα μέλη ραντεβού από το SRDI, ελαχιστοποιώντας το ποσό των δεδομένων που χρειάζεται να αποθηκευτεί στα ομότιμα μέλη ραντεβού. Το ομότιμο μέλος ραντεβού ωθεί επίσης το δείκτη σε πρόσθετα ομότιμα μέλη ραντεβού (που επιλέγονται από τον υπολογισμό μιας hash συνάρτησης του ευρετηρίου διαφημίσεων).

Ερωτήσεις

Ένα παράδειγμα διαμόρφωσης παρουσιάζεται. Το ομότιμο μέλος A είναι ένα ομότιμο μέλος άκρου και είναι διαμορφωμένο για να χρησιμοποιεί το ομότιμο μέλος R1 ως ομότιμο μέλος ραντεβού. Όταν το ομότιμο μέλος A ξεκινά ένα αίτημα ανακάλυψης ή αναζήτησης, αυτό στέλνεται αρχικά στο ομότιμο μέλος ραντεβού του – το R1, σε αυτό το παράδειγμα – αλλά και μέσω πολλαπλής διανομής στα άλλα ομότιμα μέλη στο ίδιο υποδίκτυο.



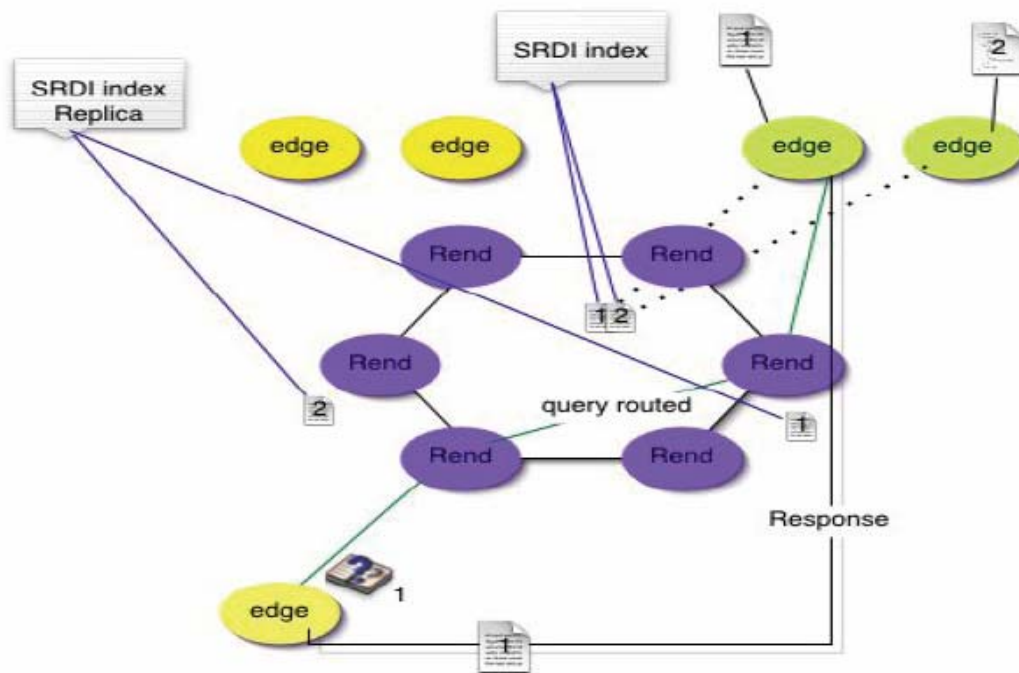
Request propagation via rendezvous peers.

Οι τοπικές ερωτήσεις γειτονιάς (δηλ., μέσα σε ένα υποδίκτυο) διαδίδονται στα γειτονικά ομότιμα μέλη χρησιμοποιώντας αυτό που ένα μέσο ορίζει ως ραδιοφωνικής μετάδοσης (broadcast) ή πολλαπλής διανομής (multicast) μέθοδο. Τα ομότιμα μέλη που λαμβάνουν την ερώτηση ανταποκρίνονται άμεσα στο αιτούντα ομότιμο μέλος, εάν περιέχουν την πληροφορία στην περιοχή αποθήκευσής τους.

Οι ερωτήσεις πέρα από την τοπική γειτονιά στέλνονται στο συνδεδεμένο ομότιμο μέλος ραντεβού. Το ομότιμο μέλος ραντεβού προσπαθεί να ικανοποιήσει την ερώτηση με την τοπική μνήμη του. Εάν περιέχει την ζητούμενη πληροφορία, απαντά άμεσα στο αιτούντα ομότιμο μέλος και δεν διαδίδει περαιτέρω το αίτημα. Εάν περιέχει το δείκτη για τον πόρο στο SRDI του, θα ειδοποιήσει το ομότιμο μέλος που δημοσίευσε τον πόρο και εκείνο το ομότιμο μέλος θα αποκριθεί άμεσα στο αιτούντα ομότιμο μέλος. (θυμηθείτε ότι το ομότιμο μέλος ραντεβού αποθηκεύει μόνο το δείκτη για τη διαφήμιση, και όχι την ίδια τη διαφήμιση).

Εάν το ομότιμο μέλος ραντεβού δεν περιέχει τις ζητούμενες πληροφορίες, ένας προεπιλεγμένος αλγόριθμος περιπατητή (walker) περιορισμένης-έκτασης (limited-range walker) χρησιμοποιείται για να συνοδεύσει το σύνολο ομότιμων μελών ραντεβού που ψάχνουν για ένα ομότιμο μέλος ραντεβού που περιέχει το δείκτη. Μια αρίθμηση βήματος χρησιμοποιείται για να διευκρινίσει το μέγιστο αριθμό φορών που το αίτημα μπορεί να διαβιβαστεί. Μόλις φθάσει η ερώτηση στο ομότιμο μέλος, αυτό απαντά άμεσα στο δημιουργό της ερώτησης.

Το ακόλουθο σχήμα απεικονίζει μια λογική άποψη για το πώς η υπηρεσία SRDI εργάζεται. Το ομότιμο μέλος 2 δημοσιεύει μια νέα διαφήμιση, και ένα SRDI μήνυμα στέλνεται στο ομότιμο μέλος ραντεβού του, το R3. Οι δείκτες θα αποθηκευτούν στο R3, και μπορεί να προωθηθούν σε άλλα ομότιμα μέλη ραντεβού στην ομότιμη ομάδα. Στην συνέχεια, το ομότιμο μέλος 1 στέλνει ένα αίτημα ερώτησης για αυτόν τον πόρο στο ομότιμο μέλος ραντεβού του, το R1. Το ομότιμο μέλος ραντεβού R1 θα ελέγξει την τοπική περιοχή αποθήκευσης των SRDI καταχωρήσεων της, και θα διαδώσει την ερώτηση εάν δεν τον βρίσκει. Όταν ο πόρος βρεθεί στο ομότιμο μέλος 2, το ομότιμο μέλος 2 θα αποκριθεί άμεσα στο R1 με τη ζητούμενη διαφήμιση.



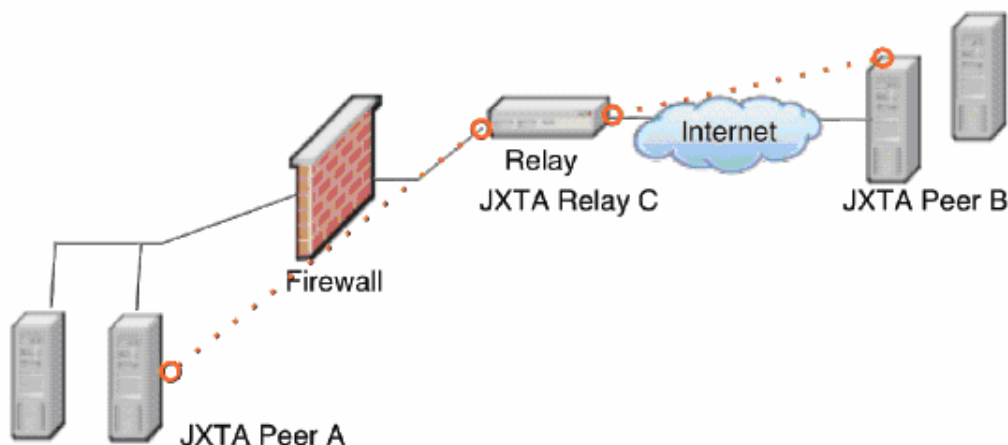
Firewalls και NAT

Ένα ομότιμο μέλος πίσω από ένα firewall μπορεί να στείλει ένα μήνυμα άμεσα σε έναν ομότιμο μέλος έξω από ένα firewall. Αλλά ένα ομότιμο μέλος έξω από το firewall δεν μπορεί να εγκαταστήσει μια σύνδεση άμεσα με έναν ομότιμο μέλος πίσω από το firewall .

Για να επικοινωνήσουν τα ομότιμα μέλη JXTA το ένα με τον άλλο μέσω ενός firewall, οι ακόλουθοι όροι πρέπει να ισχύουν:

- Τουλάχιστον ένα ομότιμο μέλος στην ομότιμη ομάδα μέσα από το firewall πρέπει να γνωρίζει τουλάχιστον έναν ομότιμο μέλος έξω από το firewall.
- Το ομότιμο μέλος μέσα και το ομότιμο μέλος έξω από το firewall πρέπει να γνωρίζουν ο ένας τον άλλον και πρέπει να υποστηρίζουν HTTP
- Το firewall πρέπει να επιτρέπει τις μεταφορές δεδομένων HTTP.

Το παρακάτω σχήμα απεικονίζει ένα χαρακτηριστικό σενάριο δρομολόγησης μηνύματος μέσω ενός firewall . Σε αυτό το σενάριο, τα ομότιμα μέλη JXTA A και B θέλουν να ανταλλάξουν ένα μήνυμα, αλλά το firewall δεν επιτρέπει την άμεση επικοινωνία τους. Το JXTA ομότιμο μέλος A κάνει αρχικά μια σύνδεση με το ομότιμο μέλος C χρησιμοποιώντας ένα πρωτόκολλο όπως το HTTP που μπορεί να διαπεράσει το firewall . Το ομότιμο μέλος C κάνει έπειτα μια σύνδεση με το ομότιμο μέλος B, χρησιμοποιώντας ένα πρωτόκολλο όπως το TCP/IP. Μια εικονική σύνδεση γίνεται έτσι μεταξύ των ομότιμων μελών A και B.



Πρωτόκολλα JXTA

Το JXTA ορίζει μια σειρά μορφών XML μηνύματος, ή *πρωτοκόλλων*, για την επικοινωνία μεταξύ των ομότιμων μελών. Τα ομότιμα μέλη χρησιμοποιούν αυτά τα πρωτόκολλα για να ανακαλύπτουν το ένα το άλλο, να διαφημίσουν και να ανακαλύψουν πόρους του δικτύου και τα μηνύματα επικοινωνίας και διαδρομής.

Υπάρχουν έξι πρωτόκολλα JXTA:

- Το πρωτόκολλο ανακάλυψης ομότιμου μέλους (*PDP, Peer Discovery Protocol*) – χρησιμοποιείται από τα ομότιμα μέλη για να διαφημίσουν τους ίδιους τους πόρους τους (π.χ. ομότιμα μέλη, ομότιμες ομάδες, διασωληνώσεις, ή υπηρεσίες) και για να ανακαλύψουν πόρους από τα άλλα ομότιμα μέλη. Κάθε πόρος ομότιμου μέλους περιγράφεται και δημοσιεύεται χρησιμοποιώντας μια διαφήμιση.
- Το πρωτόκολλο πληροφοριών ομότιμου μέλους (*PIP, Peer Information Protocol*) – χρησιμοποιείται από τα ομότιμα μέλη για να λάβουν τις πληροφορίες κατάστασης (χρόνος λειτουργίας(uptime), κατάσταση, πρόσφατη κίνηση, κλπ.) άλλων ομότιμων μελών.
- Το πρωτόκολλο αναλυτής ομότιμου μέλους (*PRP, Peer Resolver Protocol*) – επιτρέπει στα ομότιμα μέλη να στείλουν μια γενική ερώτηση σε ένα ή περισσότερα ομότιμα μέλη και να λάβουν μια απάντηση (ή πολλαπλές απαντήσεις) στην ερώτηση. Οι ερωτήσεις μπορούν να κατευθυνθούν σε όλα τα ομότιμα μέλη σε μια ομότιμη ομάδα ή σε συγκεκριμένα ομότιμα μέλη μέσα στην ομάδα. Αντίθετα από το PDP και το PIP, τα οποία χρησιμοποιούνται για την ζήτηση συγκεκριμένης προκαθορισμένης πληροφορίας, το πρωτόκολλο αυτό επιτρέπει στις υπηρεσίες ομότιμου μέλους να καθορίσουν και να ανταλλάξουν οποιαδήποτε αυθαίρετη πληροφορία χρειάζονται.

- Το πρωτόκολλο δέσμευσης διασωλήνωσης (PBP, *Pipe Binding Protocol*) – χρησιμοποιείται από τα ομότιμα μέλη για να καθιερώσουν ένα εικονικό κανάλι επικοινωνίας, ή μια διασωλήνωση, μεταξύ ενός ή περισσότερων ομότιμων μελών. Το PBP χρησιμοποιείται από ένα ομότιμο μέλος για να δεσμεύσει δύο ή περισσότερα άκρα της σύνδεσης (σημεία τέλους διασωλήνωσης).
- Το πρωτόκολλο δρομολόγησης σημείου τέλους (ERP, *Endpoint Routing Protocol*) – χρησιμοποιείται από τα ομότιμα μέλη για να βρουν τις διαδρομές στις θύρες προορισμού άλλων ομότιμων μελών. Οι πληροφορίες διαδρομής περιλαμβάνουν μια διατεταγμένη ακολουθία ταυτοτήτων ομότιμων μελών αναμετάδοσης που μπορούν να χρησιμοποιηθούν για να σταλθεί ένα μήνυμα στον προορισμό του. (Παραδείγματος χάριν, το μήνυμα μπορεί να παραδοθεί με την αποστολή του στο ομότιμο μέλος A που το αναμεταδίδει στο ομότιμο μέλος B που με την σειρά του το αναμεταδίδει στον τελικό προορισμό.)
- Το πρωτόκολλο ομότιμου μέλους ραντεβού (RVP, *Rendezvous Protocol*) – είναι ένας μηχανισμός με τον οποίο τα ομότιμα μέλη μπορούν να συμμετέχουν σε μια υπηρεσία διάδοσης. Μέσα σε μια ομότιμη ομάδα, τα ομότιμα μέλη μπορούν να είναι ομότιμα μέλη ραντεβού ή ομότιμα μέλη που ακούνε τα ομότιμα μέλη ραντεβού. Το RVP επιτρέπει σε έναν ομότιμο μέλος να στείλει μηνύματα σε όλα τα στιγμιότυπα ακρόασης υπηρεσίας. Το RVP χρησιμοποιείται από το πρωτόκολλο αναλυτή ομότιμου μέλους (PRP) και το πρωτόκολλο δέσμευσης διασωλήνωσης (PBP) για να διαδώσει μηνύματα.

Όλα τα πρωτόκολλα JXTA είναι ασύγχρονα, και είναι βασισμένα σε ένα πρότυπο ερώτησης / απάντησης. Ένα ομότιμο μέλος JXTA χρησιμοποιεί ένα από τα πρωτόκολλα για να στείλει μια ερώτηση σε ένα ή περισσότερα ομότιμα μέλη στην ομότιμη ομάδα του. Μπορεί να λάβει μηδέν, μια, ή περισσότερες απαντήσεις στην ερώτηση του. Παραδείγματος χάριν, ένα ομότιμο μέλος μπορεί να χρησιμοποιήσει το PDP για να στείλει μια ερώτηση ανακάλυψης ψάχνοντας για όλα τα γνωστά ομότιμα μέλη στην προεπιλεγμένη Ομότιμη Ομάδα Δικτύου (Net Peer Group). Σε αυτήν την περίπτωση, τα πολλαπλά ομότιμα μέλη πιθανώς να αποκριθούν με τις απαντήσεις ανακάλυψης. Σε ένα άλλο παράδειγμα, ένα ομότιμο μέλος μπορεί να στείλει ένα αίτημα ανακάλυψης ζητώντας μια συγκεκριμένη διασωλήνωση που ονομάζεται "aardvark". Εάν αυτή η διασωλήνωση δεν βρεθεί, τότε μηδέν απαντήσεις ανακάλυψης θα σταλούν ως απάντηση.

Τα ομότιμα μέλη JXTA δεν απαιτείται να υλοποιήσουν και τα έξι πρωτόκολλα, χρειάζεται μόνο να υλοποιήσουν τα πρωτόκολλα που θα χρησιμοποιήσουν.

Πρωτόκολλο ανακάλυψης ομότιμου μέλους

Το πρωτόκολλο ανακάλυψης ομότιμου μέλους (PDP, *Peer Discovery Protocol*) χρησιμοποιείται για να ανακαλύψει οποιουδήποτε δημοσιευμένους πόρους ομότιμου μέλους. Οι πόροι αναπαρίστανται σαν διαφημίσεις. Ένας πόρος μπορεί

να είναι ένα ομότιμο μέλος, μια ομότιμη ομάδα, μια διασωλήνωση, μια υπηρεσία, ή οποιοσδήποτε άλλος πόρος που έχει μια διαφήμιση.

Το PDP επιτρέπει σε έναν ομότιμο μέλος να βρει διαφημίσεις σε άλλα ομότιμα μέλη. Το PDP είναι το προεπιλεγμένο πρωτόκολλο ανακάλυψης για όλους τους χρήστες που καθορίζονται στις ομότιμες ομάδες και την προεπιλεγμένη ομότιμη ομάδα δικτύου (net peer group). Οι επί παραγγελία υπηρεσίες ανακάλυψης μπορεί να επιλέξουν να χρησιμοποιήσουν την ισχύ του PDP. Εάν μια ομότιμη ομάδα δεν έχει την υπηρεσία ανακάλυψης της, το PDP χρησιμοποιείται για να εξετάσει τα ομότιμα μέλη για διαφημίσεις.

Υπάρχουν πολλοί τρόποι να ανακαλύψει κάποιο ομότιμο μέλος διανεμημένες πληροφορίες. Η τρέχουσα πλατφόρμα δέσμευσης προγράμματος JXTA J2SE χρησιμοποιεί έναν συνδυασμό της IP πολλαπλής διανομής στο τοπικό υποδίκτυο και της χρήση των ομότιμων μελών ραντεβού, μια τεχνική βασισμένη στο σύρσιμο δικτύου (network-crawling). Τα ομότιμα μέλη ραντεβού παρέχουν το μηχανισμό αποστολής αιτημάτων από έναν γνωστό ομότιμο μέλος στο επόμενο ("crawling" γύρω από το δίκτυο) για να ανακαλύψουν δυναμικά τις πληροφορίες. Ένα ομότιμο μέλος μπορεί να προ-διαμορφωθεί με ένα προκαθορισμένο σύνολο ομότιμων μελών ραντεβού. Ένα ομότιμο μέλος μπορεί επίσης να επιλέξει να κάνει αυτοδύναμη εκκίνηση το ίδιο εντοπίζοντας δυναμικά τα ομότιμα μέλη ραντεβού ή τους δικτυακούς πόρους στο περιβάλλον εγγύτητάς του.

Τα ομότιμα μέλη παράγουν τα μηνύματα/ αιτήματα ερώτησης ανακάλυψης για να ανακαλύψουν διαφημίσεις μέσα σε μια ομότιμη ομάδα. Αυτό το μήνυμα περιέχει το πιστοποιητικό (credential) ομότιμης ομάδας του εξετάζον ομότιμου μέλους και προσδιορίζει τον εξετάζον ομότιμο μέλος στον παραλήπτη του μηνύματος. Τα μηνύματα μπορούν να σταλούν σε οποιοδήποτε ομότιμο μέλος μέσα σε μια περιοχή ή σε έναν ομότιμο μέλος ραντεβού.

Ένα ομότιμο μέλος μπορεί να λάβει μηδέν, μια, ή περισσότερες απαντήσεις σε ένα αίτημα ερώτησης ανακάλυψης. Το μήνυμα απάντησης επιστρέφει μια ή περισσότερες διαφημίσεις.

Πρωτόκολλο πληροφοριών ομότιμου μέλους

Μόλις εντοπιστεί ένα ομότιμο μέλος, οι ικανότητες και η κατάσταση του μπορεί να ζητηθούν. Το πρωτόκολλο πληροφοριών ομότιμου μέλους (PIP) παρέχει ένα σύνολο μηνυμάτων για να ληφθούν πληροφορίες κατάστασης ομότιμου μέλους. Αυτές οι πληροφορίες μπορούν να χρησιμοποιηθούν για εμπορική ή εσωτερική ανάπτυξη των εφαρμογών JXTA. Παραδείγματος χάριν, στις εμπορικές εφαρμογές αυτές οι πληροφορίες μπορούν να χρησιμοποιηθούν για να καθορίσουν τη χρήση μιας ομότιμης υπηρεσίας και για να τιμολογήσουν τους καταναλωτές της υπηρεσίας για τη χρησιμοποίησή της. Σε μια εσωτερική εφαρμογή του IT (Information Technology), οι πληροφορίες μπορούν να χρησιμοποιηθούν από το τμήμα IT για να ελέγξει την συμπεριφορά ενός κόμβου και την δικτυακή κυκλοφορία για να βελτιώσει τη γενική απόδοση του. Έτσι αυτές οι πληροφορίες

εκτός από την παροχή των πληροφοριών κατάστασης, μπορούν να αξιοποιηθούν για την παροχή του ελέγχου του τμήματος IT του κόμβου του ομότιμου μέλους .

Το μήνυμα ring PIP στέλνεται σε έναν ομότιμο μέλος να ελέγξει εάν το ομότιμο μέλος είναι ενεργό και για να πάρει πληροφορίες για το ομότιμο μέλος. Το μήνυμα ring διευκρινίζει κατά πόσον μια πλήρη απάντηση (διαφήμιση ομότιμου μέλους) ή μια απλή αναγνώριση (για το αν είναι ενεργό και το χρόνο λειτουργίας) πρέπει να επιστραφεί.

Το μήνυμα PeerInfo χρησιμοποιείται για να στείλει ένα μήνυμα σε απάντηση σε ένα μήνυμα ring. Περιέχει το πιστοποιητικό(credential) του αποστολέα, την ταυτότητα ομότιμου μέλους-πηγή και την ταυτότητα ομότιμου μέλους-προορισμού, τον χρόνο που λειτουργεί το ομότιμο μέλος (uptime), και την διαφήμιση ομότιμου μέλους.

Πρωτόκολλο αναλυτής ομότιμου μέλους

Το πρωτόκολλο αναλυτής ομότιμου μέλους (PRP) επιτρέπει στα ομότιμα μέλη να στείλουν γενικά αιτήματα ερώτησης σε άλλα ομότιμα μέλη και να αναγνωρίσουν τις αντίστοιχες απαντήσεις. Τα αιτήματα ερώτησης μπορούν να σταλούν σε έναν συγκεκριμένο ομότιμο μέλος, ή μπορούν να διαδοθούν μέσω των υπηρεσιών του ομότιμου μέλους ραντεβού στο πλαίσιο μιας ομότιμης ομάδας. Το PRP χρησιμοποιεί την υπηρεσία ραντεβού για να διαδώσει μια ερώτηση σε πολλαπλά ομότιμα μέλη, και χρησιμοποιεί μηνύματα unicast για να στείλει τις ερωτήσεις σε συγκεκριμένα ομότιμα μέλη.

Το PRP είναι ένα πρωτόκολλο αναζήτησης(foundation) που υποστηρίζει τα γενικά αιτήματα ερώτησης. Και το PIP και το PDP χιτίζονται χρησιμοποιώντας το PRP, και παρέχουν συγκεκριμένη ερώτηση/ αιτήματα: στο PIP χρησιμοποιείται για να ζητηθούν συγκεκριμένες πληροφορίες κατάστασης και στο PDP χρησιμοποιείται για να ανακαλύψει πόρους ομότιμων μελών. Το PRP μπορεί να χρησιμοποιηθεί για οποιαδήποτε γενική ερώτηση που μπορεί να απαιτηθεί από μια εφαρμογή. Παραδειγματος χάριν, το PRP επιτρέπει στα ομότιμα μέλη να ορίσουν και να ανταλλάξουν ερωτήσεις για να βρουν ή να ψάξουν πληροφορίες υπηρεσιών όπως την κατάσταση της υπηρεσίας, την κατάσταση ενός σημείου τέλους διασώληνωσης, κ.λπ.

Το μήνυμα ερώτησης αναλυτής χρησιμοποιείται για να στείλει ένα αίτημα ερώτησης αναλυτή σε μια υπηρεσία σε ένα άλλο μέλος μιας ομότιμης ομάδας. Το μήνυμα ερώτησης αναλυτή περιέχει το πιστοποιητικό(credential) του αποστολέα, μια μοναδική ταυτότητα ερώτησης, μια συγκεκριμένη υπηρεσία χειριστή, και την ερώτηση. Κάθε υπηρεσία μπορεί να καταχωρήσει έναν χειριστή στην υπηρεσία αναλυτή ομότιμης ομάδας για να επεξεργαστεί τα αιτήματα ερώτησης αναλυτή και να παράγει τις απαντήσεις. Το μήνυμα απάντησης αναλυτή χρησιμοποιείται για να στείλει ένα μήνυμα σε απάντηση σε ένα μήνυμα ερώτησης αναλυτή. Το μήνυμα απάντησης αναλυτή περιέχει το πιστοποιητικό του αποστολέα, μια μοναδική ταυτότητα ερώτησης, έναν συγκεκριμένο χειριστή υπηρεσίας και την

απάντηση. Πολλαπλά μηνύματα ερώτησης αναλυτή μπορούν να σταλούν. Ένα ομότιμο μέλος μπορεί να λάβει μηδέν, μια, ή περισσότερες απαντήσεις σε ένα αίτημα ερώτησης.

Τα ομότιμα μέλη μπορούν επίσης να συμμετέχουν στο διανεμημένο ευρετήριο κοινού πόρου(SRDI). Το SRDI παρέχει έναν γενικό μηχανισμό όπου οι υπηρεσίες JXTA μπορούν να χρησιμοποιήσουν ένα διανεμημένο ευρετήριο κοινών πόρων με άλλα ομότιμα μέλη που είναι ομαδοποιημένα ως ένα σύνολο των ικανότερων ομότιμων μελών όπως τα ομότιμα μέλη ραντεβού. Αυτοί οι δείκτες μπορούν να χρησιμοποιηθούν για να οδηγήσουν τις ερωτήσεις προς την κατεύθυνση όπου η ερώτηση είναι πολύ πιθανόν να απαντηθεί, και να αναμεταδώσουν μηνύματα στα ομότιμα μέλη που ενδιαφέρονται για τα μεταδιδόμενα μηνύματα. Το PRP στέλνει ένα μήνυμα αναλυτή SRDI στον καθορισμένο χειριστή, σε ένα ή περισσότερα ομότιμα μέλη της ομότιμης ομάδας. Το μήνυμα αναλυτή SRDI στέλνεται σε έναν συγκεκριμένο χειριστή, και περιέχει ένα αλφαριθμητικό που θα ερμηνευθεί από τον προοριζόμενο χειριστή.

Πρωτόκολλο δέσμευσης διασωλήνωσης

Το πρωτόκολλο δέσμευσης διασωλήνωσης (PBP) χρησιμοποιείται από τα ομότιμα μέλη ομάδας για να δεσμεύσουν μια διαφήμιση διασωλήνωσης σε ένα σημείο τέλους διασωλήνωσης. Η εικονική σύνδεση διασωλήνωσης μπορεί να τοποθετηθεί σε στρώσεις επάνω σε οποιοδήποτε αριθμό συνδέσεων μεταφοράς φυσικού δικτύου όπως το TCP/IP .Κάθε τέλος της διασωλήνωσης λειτουργεί για να διατηρήσει την εικονική σύνδεση και για να την επανεγκαθιδρύσει, εάν είναι απαραίτητο, δεσμεύοντας ή βρίσκοντας τα τρέχοντα σημεία τέλους ορίου της διασωλήνωσης .

Μια διασωλήνωση μπορεί να αντιμετωπισθεί ως μια αφηρημένη καθορισμένη ουρά αναμονής μηνυμάτων, που δημιουργεί , ανοίγει/ αναλύει (δεσμεύει), κλείνει (αποδεσμεύει), διαγράφει, στέλνει, και λαμβάνει διαδικασίες. Οι πραγματικές υλοποιήσεις διασωλήνωσης μπορεί να διαφέρουν, αλλά όλες οι συμμορφωμένες υλοποιήσεις χρησιμοποιούν το PBP για να δεσμεύσουν την διασωλήνωση σε ένα σημείο τέλους. Κατά τη διάρκεια που η αφαίρεση δημιουργεί την διαδικασία, ένα τοπικό ομότιμο μέλος δεσμεύει ένα σημείο τέλους διασωλήνωσης σε μια διασωλήνωση μεταφοράς.

Το μήνυμα ερώτησης PBP στέλνεται από ένα σημείο τέλους διασωλήνωσης ομότιμου μέλους για να βρει ένα σημείο τέλους διασωλήνωσης ομότιμου μέλους που διευκρινίζεται στην ίδια διαφήμιση διασωλήνωσης. Το μήνυμα ερώτησης μπορεί να ζητήσει πληροφορίες που δεν κρατιούνται από την περιοχή αποθήκευσης. Αυτό χρησιμοποιείται για να ληφθούν οι πιο ενημερωμένες πληροφορίες από ένα ομότιμο μέλος. Το μήνυμα ερώτησης μπορεί επίσης να περιέχει μια προαιρετική ταυτότητα ομότιμου μέλους, η οποία εάν υπάρχει δείχνει ότι μόνο το καθορισμένο ομότιμο μέλος πρέπει να απκριθεί στην ερώτηση.

Το μήνυμα απάντησης PBP στέλνεται πίσω στο αιτούντα ομότιμο μέλος από κάθε ομότιμο μέλος συνδεδεμένο στην διασωλήνωση. Το μήνυμα περιέχει την

ταυτότητα διασωλήνωσης, το ομότιμο μέλος όπου μια αντίστοιχη InputPipe έχει δημιουργηθεί, και μια λογική τιμή που δείχνει εάν το InputPipe υπάρχει στο καθορισμένο ομότιμο μέλος.

Πρωτόκολλο δρομολόγησης σημείου τέλους

Το πρωτόκολλο δρομολόγησης σημείου τέλους (ERP) καθορίζει ένα σύνολο μηνυμάτων αιτήματος /ερώτησης που χρησιμοποιούνται για να βρεθούν πληροφορίες δρομολόγησης. Αυτές οι πληροφορίες διαδρομών απαιτούνται για να σταλθεί ένα μήνυμα από ένα ομότιμο μέλος (πηγή) σε ένα άλλο (προορισμός). Όταν ένα ομότιμο μέλος καλείται να στείλει ένα μήνυμα σε μια δεδομένη διεύθυνση σημείου τέλους ομότιμου μέλους, κοιτάζει αρχικά στην τοπική μνήμη για να καθορίσει εάν έχει μια διαδρομή για αυτό το ομότιμο μέλος. Εάν δεν βρίσκει μια διαδρομή, στέλνει ένα αίτημα αναλυτή ερώτησης διαδρομής στα διαθέσιμα ομότιμα μέλη αναμετάδοσης ζητώντας πληροφορίες διαδρομής. Όταν ένα ομότιμο μέλος αναμετάδοσης λάβει μια ερώτηση διαδρομής, ελέγχει εάν ξέρει την διαδρομή. Εάν την γνωρίζει, επιστρέφει τις πληροφορίες διαδρομής ως μια απαρίθμηση των βημάτων.

Οποιοδήποτε ομότιμο μέλος μπορεί να ρωτήσει ένα ομότιμο μέλος αναμετάδοσης για πληροφορίες διαδρομής, και οποιοσδήποτε ομότιμο μέλος σε μια ομότιμη ομάδα μπορεί να γίνει ένας αναμεταδότης. Τα ομότιμα μέλη αναμεταδοτές τυπικά αποθηκεύουν τις πληροφορίες διαδρομής.

Οι πληροφορίες διαδρομής περιλαμβάνουν την ομότιμη ταυτότητα της πηγής, την ομότιμη ταυτότητα του προορισμού, ένα χρόνο ζωής(TTL, Time-To-Live) για την διαδρομή, και μια διαταγμένη ακολουθία ταυτοτήτων ομότιμων μελών- πύλη (gateway) . Η ακολουθία των ταυτοτήτων ομότιμων μελών μπορεί να μην είναι πλήρης, αλλά πρέπει να περιέχει τουλάχιστον τον πρώτο αναμεταδοτή.

Τα αιτήματα ερώτησης διαδρομής στέλνονται από ένα ομότιμο μέλος σε ένα ομότιμο μέλος αναμετάδοσης για να ζητήσει πληροφορίες διαδρομής. Η ερώτηση μπορεί να δείξει μια προτίμηση στο να παρακαμφθεί το περιεχόμενο της μνήμης του δρομολογητή και να αναζητήσει δυναμικά μια νέα διαδρομή.

Τα μηνύματα απάντησης διαδρομής στέλνονται από ένα ομότιμο μέλος αναμετάδοσης σε απάντηση στα αιτήματα πληροφοριών διαδρομής. Αυτό το μήνυμα περιέχει την ταυτότητα ομότιμου μέλους-προορισμού, την ταυτότητα ομότιμου μέλους και την διαφήμιση ομότιμου μέλους του δρομολογητή που ξέρει μια διαδρομή για τον προορισμό, και μια διαταγμένη ακολουθία ενός ή περισσότερων αναμεταδοτών.

Πρωτόκολλο ομότιμου μέλους ραντεβού

Το πρωτόκολλο ομότιμου μέλους ραντεβού (RVP) είναι αρμόδιο για τη διάδοση των μηνυμάτων μέσα σε μια ομότιμη ομάδα. Ενώ διαφορετικές ομότιμες ομάδες μπορεί να έχουν διαφορετικά μέσα για να διαδώσουν τα μηνύματα, το

πρωτόκολλο ομότιμου μέλους ραντεβού καθορίζει ένα απλό πρωτόκολλο που επιτρέπει:

- Στα ομότιμα μέλη να συνδεθούν με την υπηρεσία (να είσαι σε θέση να διαδώσουν τα μηνύματα και να λάβουν τα μεταδομένα μηνύματα)
- Έλεγχο της διάδοση του μηνύματος (TTL, ανίχνευση ανάδρασης(loopback detection) για να μην περνά το μήνυμα από τα ίδια σημεία, κ.λπ.).

Το RVP χρησιμοποιείται από το ομότιμο μέλος πρωτόκολλο αναλυτή (PRP) και από το πρωτόκολλο δέσμευσης διασωλήνωσης (PBP) προκειμένου να διαδοθούν τα μηνύματα.

Κεφάλαιο 3:Java

Ιστορική Αναδρομή

Το 1990, ο μηχανικός λογισμικού της Sun Microsystems Patric Naughton που ασχολούνταν εκείνη την εποχή με την υποστήριξη όλων των διαφορετικών APIs (Abstract Programming Interfaces) που είχε στη διάθεσή της η Sun Microsystems, σκεφτόταν σοβαρά τη μετακίνησή του στην ανταγωνιστική εταιρία NeXT μετά από αντίστοιχη πρόταση. Όταν ανακοινώθηκε η απόφασή του για αποχώρηση από τη Sun στον τότε γενικό διευθυντή της Scott McNealy, και φίλο του Naughton, αυτός δεν τη δέχθηκε και του έδωσε την ευκαιρία να φτιάξει μία λίστα με όλα τα «κακώς κείμενα» στην εταιρία, καθώς επίσης και με τις προτάσεις του σχετικά με το τι ήταν αναγκαίο να γίνει.

Ο Naughton πραγματικά έφτιαξε αυτή τη λίστα με το σκεπτικό ότι αυτόν δεν θα τον απασχολούσε τίποτα από αυτά που θα παρατηρούσε και θα πρότεινε. Στο γράμμα που έστειλε στο γενικό διευθυντή αργότερα, υπογράμμιζε πως αν η Sun αποφάσιζε να ευθυγραμμίσει την τακτική της στο μέλλον, πολλοί από τους προγραμματιστές που απασχολούνταν για την υποστήριξη των Windows Systems (X-Windows, Motif κλπ.) θα μπορούσαν να χρησιμοποιηθούν πιο αποδοτικά. Ο τρόπος που παρουσίασε ο Naughton τις προτάσεις του ήταν όπως ακριβώς θα έβλεπε κάποιος εξωτερικός παρατηρητής την κατάσταση.

Ακριβώς αυτό το αμερόληπτο γράμμα, έκανε τον γύρο αθόρυβα σε πολλούς μηχανικούς λογισμικού που εργάζονταν στη Sun και προς μεγάλη έκπληξη του Naughton, το ηλεκτρονικό του ταχυδρομείο κατακλυζόταν καθημερινά από μηνύματα συναδέλφων που υποστήριζαν τις θέσεις του. Η αποδοχή των προτάσεών του από τους συναδέλφους του μηχανικούς μετέφερε ουσιαστικά τους προβληματισμούς του Naughton στους διευθύνοντες της εταιρίας.

Το αποτέλεσμα ήταν η Sun να κάνει μία αρκετά γενναιόδωρη και ταυτόχρονα πρωτότυπη προσφορά εργασίας στον Naughton. Του πρότεινε να δημιουργήσει μία ομάδα μηχανικών και προγραμματιστών με απόλυτη ελευθερία να ασχοληθούν με οτιδήποτε προτιμούσαν. Το μόνο τους παραδοτέο θα έπρεπε να είναι κάτι πραγματικά πρωτότυπο.

Η ομάδα Green (όπως ήταν ο κωδικός που της δόθηκε) απέκτησε πραγματικά απόλυτη ανεξαρτησία κινήσεων. Το εκπληκτικό είναι ότι το πρώτο πράγμα που απασχόλησε την ομάδα ήταν οι παιχνιδομηχανές που κυκλοφορούσαν στο εμπόριο και είχαν γνωρίσει τεράστια επιτυχία (Nintendo Game Boy, Sega). Το ορατό πρόβλημα για το μέλλον ήταν ότι οι μηχανές αυτές είχαν λόγω της κατασκευής της κεντρικής μονάδας επεξεργασίας τους περιορισμένες δυνατότητες επέκτασης. Έτσι, αν κάποιος κατασκευαστής αποφάσιζε να αναπτύξει κάποιο πρωτοποριακό προϊόν, θα συναντούσε σίγουρα δυσκολίες. Με αυτές τις σκέψεις κατά νου και με το δεδομένο ότι οι περισσότεροι μικροί επεξεργαστές περιορίζονταν μόνο σε κάποιο συγκεκριμένο εύρος εφαρμογών, πρότειναν μια νέα

προσέγγιση στον τρόπο προγραμματισμού με στόχο να επιφέρουν την πρωτοπορία στον χώρο των προϊόντων που ήδη κυκλοφορούσαν.

Την ίδια εποχή στα εργαστήρια της Sun γεννιόταν μία νέα γλώσσα προγραμματισμού που ο James Gosling, προϊστάμενος του Naughton, ονόμασε Oak, εμπνευσμένος από τη βελανιδιά που έβλεπε έξω από το παράθυρό του. Αυτή η γλώσσα προγραμματισμού αποσκοπούσε στο να εκμεταλλευτεί ένα πυρήνα χαρακτηριστικών όλων των ολοκληρωμένων στις παιχνιδομηχανές χειρός και να προετοιμάσει το έδαφος για μία νέα, ενιαία πλατφόρμα υλοποίησης προγραμμάτων.

Καθώς η υλοποίηση της γλώσσας Oak συνεχιζόταν, η ομάδα Green συνέχιζε ανεξάρτητα την εκτεταμένη έρευνα για την προσέλκυση του κοινού προς τα ηλεκτρονικά παιχνίδια και τον τρόπο που αλληλεπιδρούσαν με τα μηχανήματα. Σαν αποτέλεσμα αυτής της έρευνας, η ομάδα δημιούργησε μία μικρή, κινητή συσκευή (ακριβώς όπως και το Gameboy) με το κωδικό όνομα “*7”. Το *7 παρουσίαζε έναν κινούμενο χαρακτήρα (που σχεδιάστηκε από τον Joe Palrang και ονομάστηκε Duke), ο οποίος βοηθούσε τους χρήστες να εξερευνήσουν μία απλή στη χρήση αλλά πλούσια σε γραφικά διεπαφή. Η σκέψη γύρω από την οποία κινήθηκε η υλοποίηση είναι ότι μία διεπαφή χρήστη θα πρέπει να είναι φιλική και να ψυχαγωγεί ταυτόχρονα το χρήστη πάνω σε μία εύκολα μετακινούμενη, προσωπική συσκευή. Ο Duke αργότερα θα γινόταν η μασκότ και το σήμα κατατεθέν της Java[17].



*Η αρχική συσκευή του *7*



*Μία οθόνη του *7, που δείχνει μία πειραματική διεπαφή χρήστη*

Σε μία προσπάθεια να γίνει η ομάδα Green ακόμα πιο αυτόνομη, η Sun την ανεξαρτητοποιεί σε θυγατρική εταιρία με το όνομα First Person. Η νεοσυσταθείσα εταιρία είχε πολλές νέες ιδέες αλλά δεν ήξεραν ακόμη πώς να τις χρησιμοποιήσει για να κάνει ένα άνοιγμα προς την αγορά. Ο μόνος τρόπος έδειχνε να είναι η ανερχόμενη τεχνολογία της αλληλεπιδραστικής τηλεόρασης (interactive television). Παρ’ όλα αυτά, κάποιες προσπάθειες προς την κατεύθυνση

αυτή δεν καρποφόρησαν κυρίως λόγω λανθασμένων διοικητικών χειρισμών, σχετικά με τα πνευματικά δικαιώματα κυρίως, οπότε και εγκαταλείφθηκε η ιδέα.



*Ο Duke, ο χαρακτήρας που σχεδιάστηκε αρχικά για το *7 και κατέληξε μασκότε της Java*

Είχε φτάσει πλέον η εποχή (1993) που το Εθνικό Κέντρο για Εφαρμογές Υπερουπολογιστών (National Center for Supercomputing Applications) εισήγαγε στην αγορά το Mosaic, την πρώτη εφαρμογή πλοήγησης (browser) για τον Παγκόσμιο Ιστό (Web). Η συνέχεια, ήταν η καταγιστική νέα τεχνολογία του Παγκόσμιου Ιστού η οποία αύξανε την κίνηση στο Διαδίκτυο (Internet) όλο και περισσότερο. Με δεδομένη αυτή την κατεύθυνση της αγοράς προς τα δίκτυα, η First Person αποφάσισε στις αρχές του 1994 να επικεντρώσει τις προσπάθειές της προς τις εφαρμογές πολυμέσων συνεχούς σύνδεσης (online multimedia).

Παράλληλα, η γλώσσα Oak ήταν έτοιμη να παρουσιαστεί όταν ήρθε η ιδέα να δημιουργηθεί ένα λειτουργικό σύστημα γραμμένο σε Oak. Πραγματικά, η κίνηση αυτή έγινε και στη συνέχεια ο Naughton πρότεινε την ελεύθερη διάθεση του κώδικα στο Internet, ιδέα που επίσης υιοθετήθηκε με τη δημιουργία ενός μεταγλωττιστή για Oak γραμμένο στην ίδια τη γλώσσα. Για την υποστήριξη του νέου λειτουργικού συστήματος, ο Naughton, συνεργάστηκε με την ομάδα ανάπτυξης της Oak, δημιούργησαν την πρώτη εφαρμογή πλοήγησης για τον Παγκόσμιο ιστό γραμμένη σε Oak, και την ονόμασαν WebRunner. Το πρώτο applet στην ιστορία δημιουργήθηκε τότε και παρουσίαζε τον Duke να χαιρετά τους δημιουργούς του.

Η Sun υποστήριξε την απόφαση να μοιραστεί ελεύθερα ο κώδικας της Oak στο Internet αλλά αφού πρώτα άλλαξε το όνομά της σε Java για περισσότερο εμπορικούς λόγους. Άλλα ονόματα που είχαν προταθεί για τη Java ήταν Neon, Lyric, Pepper και Silk.



Διαφημιστικό του WebRunner

Το μόνο που χρειαζόταν πλέον η τεράστια κοινότητα του Διαδικτύου ήταν μία πλατφόρμα που να τρέχει τις εφαρμογές που έχουν γραφτεί σε Java. Αυτή δεν ήταν άλλη από τον WebRunner ο οποίος επίσης μετονομάστηκε σε HotJava λόγω νομικών προβλημάτων με τα κατατεθέντα σήματα. Η συνέχεια ήρθε με την υποστήριξη από τη Netscape της Java στη δημοφιλή εφαρμογή πλοήγησής της (Navigator).

Από τότε μέχρι σήμερα η Java έχει επεκταθεί πάρα πολύ, κάνοντας τον Duke μία από τις πιο δημοφιλείς φυσιογνωμίες στο Διαδίκτυο. Από την εποχή εκείνη έχουν συμβεί πάρα πολλά στο Διαδίκτυο και στην τεχνολογία για επιχειρήσεις, και αυτά είναι μόνο η αρχή: το JDK™, το sandbox, τα applets, χιλιάδες προσπάθειες προσανατολισμένες σε τεχνολογίες Java, η αρχιτεκτονική JavaBeans™, το Java Studio™, το Netscape Communicator, χιλιάδες παροχές υπηρεσιών Διαδικτύου, 60 εκατομμύρια χρήστες του Διαδικτύου, 56K και καλωδιακά modems, το ηλεκτρονικό εμπόριο, τα servlets, οι Java Foundation Classes, τα συστατικά Enterprise JavaBeans™, το JavaOS for Business™, και δεσμεύσεις από μεγάλες εταιρίες όπως η IBM.

Τα παραπάνω είναι κάτι περισσότερο από έναν απλό κατάλογο προϊόντων λογισμικού και γεγονότων. Είναι μία ένδειξη του πόσο γρήγορα αναπτύχθηκε αυτή η τεχνολογία κατά τη διάρκεια των τελευταίων ετών. Η τεχνολογία της Java είναι ένα από τα σπουδαιότερα υπολογιστικά περιβάλλοντα αυτή τη στιγμή, ένας ποταμός που έχει πλημμυρίσει τις όχθες του και κατευθύνεται τώρα προς οποιεσδήποτε βιομηχανίες ή επιχειρήσεις.

Στους μήνες που ακολούθησαν την αρχική ανακοίνωση της Sun, η τιμή της μετοχής της ανέβηκε στα ύψη, πράγμα για το οποίο εν μέρει ευθύνεται και η τεχνολογία Java. Κατά το τέλος του πρώτου πλήρους χρόνου τεχνολογίας Java, το τμήμα JavaSoft είχε παραχωρήσει 38 άδειες και είχε προσελκύσει 6.000 άτομα που ασχολούνταν με την ανάπτυξη λογισμικού στην εμπορική έκθεσή της, το πρώτο συνέδριο JavaOne Developer Conference™. Κατά το τέλος του δεύτερου χρόνου, η JavaSoft είχε παραχωρήσει σχεδόν 100 άδειες και το συνέδριο JavaOne προσέλκυε 10.000 άτομα.

Στον εορτασμό των τρίτων γενεθλίων της τεχνολογίας Java, το μετονομασμένο τμήμα Java της Sun απασχολούσε 800 άτομα, υποστηριζόταν από 150 άδειες και πολλές εκατοντάδες χιλιάδων μηχανικών λογισμικού παγκοσμίως. Κατάφερε να φιλοξενήσει 15.000 άτομα στο JavaOne (η μεγαλύτερη σύναξη προγραμματιστών που έγινε ποτέ παγκοσμίως), κατάφερε να υπερκεράσει τον προηγούμενο ετήσιο στόχο της για 10.000 downloads του JDK σε μία μέρα, και υπερηφανευόταν για μία ιδιόκτητη πολυάσχολη γραμμή T3.

Χαρακτηριστικά της Java

Η γλώσσα προγραμματισμού Java σχεδιάστηκε για να δώσει λύσεις σε μία σειρά πρακτικών προβλημάτων που εμφανίζονταν σε όλες τις ήδη υπάρχουσες γλώσσες. Η γλώσσα στην οποία στηρίχτηκε αρκετά η προσπάθεια δημιουργίας αυτής της

νέας γλώσσας ήταν η C++, μία αντικειμενοστραφής (object oriented) γλώσσα η οποία όμως πέρα από τα αρχικά προβλήματα, κυρίως σε πρακτικά ζητήματα, με τους υπάρχοντες μεταφραστές, παρουσίασε ουσιαστικά προβλήματα στην πορεία. Η λύση που προτάθηκε από τους μηχανικούς της Sun ήταν η δημιουργία μιας εντελώς νέας γλώσσας που θα είναι αντικειμενοστραφής και θα βασίζεται στη C++.

Ο επίσημος ορισμός της Java, όπως αυτός δίνεται από τη Sun, έχει ως εξής:

Java: A simple, object oriented, network-savvy, interpreted, robust, secure, architecture neutral, portable, high-performance, multithreaded, dynamic language.

Java: Μία απλή, αντικειμενοστραφής, έχουσα δικτυακή γνώση, μεταφράσιμη, εύρωστη, ασφαλής, ανεξάρτητη αρχιτεκτονικής, μεταφέριμη, υψηλής απόδοσης, πολυνηματική, δυναμική γλώσσα.

Ακολούθως με βάση τον παραπάνω ορισμό θα παρουσιάσουμε μία σύντομη και συνοπτική περιγραφή των χαρακτηριστικών αυτής της νέας γλώσσας.

Απλότητα (Simple)

Στην προσπάθεια δημιουργίας της Java λήφθηκε σοβαρά υπόψη η απλότητα. Η νέα γλώσσα δε θα έπρεπε να αγνοεί την πραγματικότητα στον προγραμματισμό: οι περισσότεροι προγραμματιστές γνώριζαν και χρησιμοποιούσαν τη C ενώ όσοι ενδιαφέρονταν για αντικειμενοστραφείς εφαρμογές, τη C++. Δε θα ήταν επομένως ώριμο να εισάγουν μία εντελώς νέα γλώσσα χωρίς καμία σχέση με αυτό που γνώριζαν πολύ καλά οι προγραμματιστές.

Ωστόσο, καταβλήθηκε προσπάθεια ώστε η Java να απαλλαγεί από χαρακτηριστικά των C/C++ που είτε δε χρησιμοποιούνταν ευρέως, είτε περισσότερο δυσκόλευαν τον προγραμματιστή, παρά του πρόσφεραν επιπλέον δυνατότητες. Τέτοια χαρακτηριστικά είναι, ενδεικτικά, η υπερφόρτωση τελεστών (operator overloading) και η πολλαπλή κληρονομικότητα (multiple inheritance) από τη C++.

Επιπλέον το κυριότερο πρόβλημα της C είναι η αποδοτική διαχείριση της μνήμης. Η δέσμευση ή αποδέσμευση μνήμης είναι μία ιδιαίτερα επίπονη διαδικασία που αποσπά τον προγραμματιστή από το σκοπό του, απασχολώντας τον με ζητήματα ρουτίνας. Η Java για το λόγο αυτό εισήγαγε την αυτόματη συλλογή «απορριμμάτων» (garbage collection), δηλαδή την αυτόματη αποδέσμευση μνήμης στην οποία δεν υπάρχουν αναφορές (referenced). Πλέον ο προγραμματιστής απαλλάσσεται από την ανάγκη να γνωρίζει και να ασχολείται με τη διαχείριση της μνήμης στην πλατφόρμα στην οποία εργάζεται.

Τέλος, ένα άλλο χαρακτηριστικό που πηγάζει από την απλότητα αφορά στο μέγεθος. Ένας από τους στόχους δημιουργίας της Java ήταν να μπορεί να τρέχει σε μικρές μηχανές χειρός. Έτσι, οι βιβλιοθήκες της καθώς και όλα τα προγράμματα γραμμένα σε αυτή είναι αρκετά μικρά, πράγμα που τις δίνει μεγάλο πλεονέκτημα σε ένα ανομοιογενές δικτυακό περιβάλλον στο οποίο έχει σημασία ο χρόνος που απαιτείται προκειμένου να «κατέβει» η εφαρμογή.

Αντικειμενοστραφής Γλώσσα (Object Oriented)

Η έννοια του αντικειμενοστραφούς (object oriented) είναι από τις πιο πολυχρησιμοποιημένες στη βιομηχανία παραγωγής λογισμικού. Εντούτοις, αντιπροσωπεύει μία από τις δυναμικότερες έννοιες στον προγραμματισμό, υλοποιώντας καθαρά και αποδοτικά τις διεπαφές και τις επαναχρησιμοποιούμενα συστατικά (components) λογισμικού.

Ο αντικειμενοστραφής σχεδιασμός είναι γενικότερα μία τεχνική που επικεντρώνει στο σχεδιασμό με αντικείμενα - δεδομένα (objects) και στη διεπαφή αυτών προς τα έξω παρά με τα εργαλεία που θα χρησιμοποιηθούν για τη δημιουργία τους. Είναι περισσότερο ενδιαφέρον, δηλαδή, οι ιδιότητες ενός αντικείμενου και πώς αυτό αλληλεπιδρά με άλλα αντικείμενα στον ίδιο ή σε κάποιο άλλο χώρο (πρόγραμμα ή πλατφόρμα). Είναι ουσιαστικά αυτό που κάνει η τεχνολογία plug-and-play.

Η ιδέα για την υλοποίηση των αντικειμενοστραφών χαρακτηριστικών της Java προήλθαν ουσιαστικά αυτούσια από τη C++, ενώ χρησιμοποιήθηκαν και επεκτάσεις από την Objective C για τη δυναμικότερη ανάλυση των μεθόδων-συναρτήσεων (dynamic method resolution).

Δικτυακή Γνώση (Network-aware)

Η Java έχει ενσωματωμένη μία μεγάλη συλλογή από βιβλιοθήκες που υλοποιούν άμεσα τα γνωστότερα TCP/IP πρωτόκολλα, όπως το HTTP και το FTP. Αυτό κάνει τη δημιουργία δικτυακών συνδέσεων ευκολότερη από αυτή των C/C++. Οι εφαρμογές της Java μπορούν να προσπελάσουν μία οποιαδήποτε άλλη εφαρμογή ή γενικότερα πόρο στο δίκτυο με χρήση των URLs (Uniform Resource Locators) και με την ίδια ευκολία που μπορεί κάποιος προγραμματιστής να προσπελάσει το τοπικό σύστημα αρχείων.

Ευρωστία (Robust)

Η Java σχεδιάστηκε ως μία γλώσσα της οποίας τα προγράμματα πρέπει να είναι αξιόπιστα από διάφορες απόψεις. Έτσι, δίνεται ιδιαίτερη έμφαση στον έλεγχο κατά τη μεταγλώττιση πιθανών αιτιών που θα μπορούσαν να προκαλέσουν λάθος. Επιπλέον, γίνεται κατά την εκτέλεση του προγράμματος ένας δυναμικός έλεγχος για να εντοπιστούν άλλα λάθη εκτέλεσης.

Για να γίνει σαφές το πρόβλημα που παρουσιάζουν άλλες γλώσσες, αναφέρουμε ενδεικτικά το πρόβλημα του μεταφραστή της C, όσον αφορά στη δυνατότητα έμμεσης δήλωσης συναρτήσεων. Σε μεγάλης κλίμακας εφαρμογές μία δηλωμένη συνάρτηση μπορεί, κατά λάθος, να θεωρηθεί και υλοποιημένη χωρίς στην πραγματικότητα να υπάρχει κώδικας για τη συγκεκριμένη συνάρτηση ή να είναι προγενέστερη έκδοση της ίδιας, καταλήγοντας σε λάθος κατά την εκτέλεση (runtime error). Σε γλώσσες που χρησιμοποιούν τύπους, όπως η C++, ο έλεγχος τύπων και δηλώσεων γίνεται κατά τη φάση της μεταγλώττισης. Ωστόσο ο μεταγλωττιστής της C++ δεν είναι απαλλαγμένος από όλες τις επιρροές της C.

Η ουσιαστικότερη όμως διαφορά των Java και C/C++ είναι ότι η μεν πρώτη χρησιμοποιεί ένα μοντέλο υλοποίησης των δεικτών που εξαλείφει την πιθανότητα λανθασμένης εγγραφής ή αλλοίωσης των δεδομένων στη μνήμη, οι άλλες χρησιμοποιούν την τεχνική αριθμητικής των δεικτών (pointer arithmetic - οι δείκτες αντιμετωπίζονται σαν διευθύνσεις μνήμης πάνω στις οποίες μπορούν να εφαρμοστούν όλες οι επιτρεπτές αριθμητικές πράξεις όπως πρόσθεση, αφαίρεση, μετατόπιση). Για τον ίδιο σκοπό η Java χρησιμοποιεί πραγματικούς πίνακες (true arrays) που εγγυώνται ότι γίνεται πάντα ο έλεγχος του δείκτη και δε θα γίνει απόπειρα εγγραφής σε περιοχή μνήμης πέρα από τα όρια του πίνακα. Επίσης δεν υποστηρίζει την μετατροπή αριθμού σε δείκτη (κάτι που περιπλέκει αρκετές φορές τους προγραμματιστές).

Με δεδομένο ότι δε θα γίνει λάθος στο χειρισμό της μνήμης, ο προγραμματιστής μπορεί να είναι ήσυχος ότι κανείς δε θα μπορέσει να του αλλοιώσει τα δεδομένα και επομένως δε θα χρειαστεί να λάβει επιπλέον μέτρα ασφάλειας. Τέλος, ο μεταφραστής δεν αφήνει μεγάλα περιθώρια για επιλογές στον προγραμματιστή όσον αφορά σε τύπους συναρτήσεων. Αυτό το φαινομενικά αρνητικό χαρακτηριστικό, εξασφαλίζει ότι λάθη τύπων θα ελεγχθούν και θα βρεθούν κατά τη διάρκεια της μεταγλώττισης και στο εξής δε θα πρέπει να προβληματίζουν τον προγραμματιστή ως πιθανές αιτίες λαθών.

Ασφάλεια (Security)

Ένας από τους στόχους της Java είναι να μπορεί να τρέχει πάνω από δικτυακά και κατανομημένα περιβάλλοντα. Με τα γνωστά προβλήματα ασφαλείας στα δίκτυα, θα έπρεπε να ενσωματωθούν κάποιες τεχνικές που θα εξασφάλιζαν την ασφάλεια των προγραμμάτων και των δεδομένων τους στο δίκτυο.

Πραγματικά, η Java εφοδιάστηκε με έναν ισχυρό μηχανισμό ελέγχου. Όταν ένα applet, δηλαδή μία Java εφαρμογή, φθάνει στον προορισμό της, γίνεται τοπικά έλεγχος για να διαπιστωθεί αν πραγματικά είναι η εφαρμογή που ζητήθηκε κι αν έχει όλα τα απαιτούμενα διακριτικά γνωρίσματα. Οι τεχνικές ελέγχου γνησιότητας βασίζονται στη μέθοδο κρυπτογράφησης δημοσίου κλειδιού (public key encryption).

Όσον αφορά στο θέμα της ασφάλειας, αξίζει να σημειωθούν και τα χαρακτηριστικά που αναφέρθηκαν και σε προηγούμενη παράγραφο. Με την απαγόρευση της πρόσβασης άλλων αντικειμένων στους πόρους ενός αντικειμένου, ουσιαστικά εξαλείφεται ο κίνδυνος των ιών και των αλλοιωμένων δεδομένων (corrupted data).

Ανεξάρτητη Αρχιτεκτονικής (Architecture Neutral)

Η Java σχεδιάστηκε για να υποστηρίζει εφαρμογές πάνω από δίκτυο αλλά ως γνωστό ένα δίκτυο περιλαμβάνει ένα μεγάλο σύνολο διαφορετικών μηχανών και λειτουργικών συστημάτων. Για να εξασφαλιστεί το ότι η ίδια εφαρμογή θα τρέχει ομοιογενώς σε αυτό το περιβάλλον, ο μεταγλωττιστής παράγει έναν ανεξάρτητο

μηχανής αντικείμενο κώδικα (object code), ο οποίος με τη βοήθεια του Συστήματος Εκτέλεσης Java (Java Runtime System - JRE ή Java Virtual Machine - JVM) μπορεί να τρέξει σε πολλούς διαφορετικούς επεξεργαστές.

Το παραπάνω γεγονός όμως δεν είναι μόνο χρήσιμο στην περίπτωση των δικτύων και των δικτυακών εφαρμογών αλλά και στις τοπικές εφαρμογές. Μέχρι τώρα, με την μεγάλη ποικιλία επεξεργαστών και μηχανημάτων που κυκλοφορούν στην αγορά, οι κατασκευαστές πακέτων λογισμικού είναι υποχρεωμένοι από τους νόμους της αγοράς να παράγουν το ίδιο προϊόν για πολλές διαφορετικές πλατφόρμες, όπως για παράδειγμα για UNIX, IBM PC, Apple Macintosh. Όσο περισσότερο προχωρά η αγορά επεξεργαστών, τόσο πιο πολύ εντείνεται το πρόβλημα. Με τη Java, η φιλοσοφία είναι ότι η εφαρμογή γράφεται μία φορά σε οποιαδήποτε πλατφόρμα και έπειτα θα μπορεί να τρέχει οπουδήποτε (WORA - Write Once, Run Anywhere).

Ο μεταφραστής της Java παράγει εντολές bytecode, οι οποίες είναι εντολές που δεν έχουν να κάνουν με κάποια συγκεκριμένη μηχανή αλλά με την Εικονική Μηχανή Java (Java Virtual Machine - JVM). Στη συνέχεια καθώς εκτελείται ο κώδικας αυτός γίνεται και η μετατροπή σε κώδικα συγκεκριμένο για την κάθε μηχανή.

Μεταφέρσιμος Κώδικας (Portable)

Το γεγονός ότι ο κώδικας που παράγει ο μεταφραστής της Java είναι ανεξάρτητος αρχιτεκτονικής, δεν συνεπάγεται απαραίτητα πως θα είναι και μεταφέρσιμος (portable). Για παράδειγμα, στη C και στη C++ υπάρχουν ζητήματα που εξαρτώνται από τη συγκεκριμένη αρχιτεκτονική στην οποία τρέχει ο μεταφραστής. Για παράδειγμα το μέγεθος ενός ακεραίου αριθμού, που αντιπροσωπεύεται από τον τύπο int, διαφέρει από μηχανή σε μηχανή, ανάλογα με τον επεξεργαστή.

Στη Java ακόμα και οι βιβλιοθήκες είναι portable σε όλες τις υποστηριζόμενες πλατφόρμες. Έτσι, για παράδειγμα, υπάρχει μία κλάση αντικειμένων για τα γραφικά παράθυρα που είναι ίδια για όλες τις UNIX, Windows 95/98/2000/NT, Mac πλατφόρμες. Ακόμα και ο ίδιος ο μεταφραστής της Java είναι γραμμένος σε Java, αλλά το περιβάλλον εκτέλεσης έχει γραφεί σε ANSI C απολύτως συμβατή με όλες τις πλατφόρμες.

Μεταφράσιμη (Interpreted)

Ο bytecode κώδικας, δηλαδή το αποτέλεσμα της διαδικασίας της μεταγλώττισης, μεταφράζεται σε τοπικές εντολές μηχανής και δεν αποθηκεύεται πουθενά. Ο κώδικας αυτός μεταφράζεται στη συνέχεια (interpreted) με όλα τα γνωστά πλεονεκτήματα που έχει αυτό το γεγονός στην απόδοση του προγράμματος.

Η διασύνδεση (linking) που είναι η υπεύθυνη διαδικασία για τη σύνθεση των επιμέρους κομματιών (modules) του προγράμματος έχει μία σχετικά εύκολη δουλειά, με αποτέλεσμα να μην υπάρχει μεγάλος φόρτος στη φάση αυτή. Επιπρόσθετα, λόγω αυτής της μεταφράσιμης φύσης του κώδικα, η εκσφαλμάτωση

(debugging) είναι μία σχετικά εύκολη διαδικασία, ακόμα και στο ανομοιογενές περιβάλλον ενός δικτύου.

Υψηλή Απόδοση (High Performance)

Πέρα από την πολύ καλή απόδοση που επιτυγχάνεται με τη χρήση bytecodes, πολλές φορές είναι αναγκαία ακόμα υψηλότερη απόδοση. Για το λόγο αυτό, τα bytecodes μπορούν να μεταφράζονται σε εντολές μηχανής για τη συγκεκριμένη κεντρική μονάδα επεξεργασίας (Central Processing Unit - CPU), κατά τη μεταφορά και εκτέλεσή τους (on the fly).

Επιπλέον, επειδή η λογική του σχεδιασμού της παραγωγής εντολών μηχανής κατά αποδοτικό τρόπο ήταν πρωταρχική, η διαδικασία είναι εξαιρετικά απλή και αποδοτική. Μάλιστα, ο μεταφραστής κάνει αυτόματα δέσμευση καταχωρητών (register allocation) και βελτιστοποίηση κατά τη διάρκεια παραγωγής των bytecodes.

Σε μεταφράσιμο (interpreted) κώδικα, έχουμε περίπου 300.000 κλήσεις μεθόδων το δευτερόλεπτο σε ένα Sun SPARCStation 10. Έτσι η απόδοση των bytecodes που μετατρέπονται σε εντολές μηχανής είναι ασύγκριτη με αυτή των C/C++.

Πολυνηματικότητα (Multithreaded)

Πολυνηματικότητα (multithreading) καλείται η τεχνική δημιουργίας εφαρμογών που βασίζονται στα πολλαπλά «νήματα» (threads). Ένα νήμα είναι μία ακολουθία εντολών η οποία μπορεί να εκτελείται παράλληλα με άλλες παρόμοιες. Είναι δηλαδή ένα είδος παραλληλισμού του κώδικα, ο οποίος παρ' όλα αυτά μπορεί να τρέχει και σε έναν επεξεργαστή. Η δυσκολία του να γραφούν προγράμματα που να αντιμετωπίζουν καταστάσεις όπου πολλά πράγματα ταυτόχρονα πρέπει να συμβαίνουν, ανάγεται στη δυσκολία εξαγωγής ενός παράλληλου προγράμματος από το αντίστοιχο σειριακό, με τα αντίστοιχα προβλήματα συγχρονισμού, αδιεξόδων, αποτυχιών.

Η Java, παρ' όλα αυτά, έχει ένα ισχυρό σύνολο αρχών συγχρονισμού που βασίζεται στο *monitor and condition variable paradigm* το οποίο αναφέρθηκε για πρώτη φορά από τον [Hoare, C.A.R., 1974]. Με τη χρήση αυτών των τεχνικών μέσα στην ίδια τη γλώσσα, τα νήματα γίνονται ευκολότερα στη χρήση τους. Η όλη υλοποίηση βασίστηκε στο σύστημα Cedar/Mesa της Xerox.

Άλλα χαρακτηριστικά της πολυνηματικότητας είναι η καλύτερη αλληλεπίδραση πραγματικού χρόνου. Σε αυτόνομα περιβάλλοντα Java, η απόκριση προσεγγίζει πολύ τις απαιτήσεις απόκρισης πραγματικού χρόνου. Αν όμως αυτό το περιβάλλον βρίσκεται κάτω από ένα διαφορετικό λειτουργικό σύστημα (π.χ. UNIX, Windows, Macintosh) η απόδοση αυτή φυσικό είναι να παρουσιάζεται μειωμένη.

Δυναμικότητα (Dynamic)

Η Java είναι κατά πολλές έννοιες πιο δυναμική από ότι η C και η C++. Σχεδιάστηκε ώστε να προσαρμόζεται εύκολα σε ένα διαρκώς εξελισσόμενο περιβάλλον.

Για παράδειγμα ας υποθέσουμε ότι δύο εταιρίες παράγουν λογισμικό σε C++. Έστω, ότι η A εταιρία παράγει μία βιβλιοθήκη κλάσεων την οποία χρησιμοποιεί η εταιρία B για την παραγωγή του τελικού προϊόντος. Αν κάποια στιγμή η A αποφασίσει να βελτιώσει τη βιβλιοθήκη της, τότε η B θα πρέπει να πάρει να ανανεωμένη βιβλιοθήκη και να την χρησιμοποιήσει για να κάνει εκ νέου μετάφραση στο λογισμικό της και να το διανείμει πάλι στην αγορά. Με άλλα λόγια δεν είναι άμεσα χρησιμοποιήσιμη η νέα βιβλιοθήκη, πράγμα που έρχεται σε αντίθεση με την έννοια του πραγματικού αντικειμενοστραφούς προγραμματισμού.

Αυτό που κάνει η Java, είναι να διασυνδέει τα επιμέρους κομμάτια κατά τη διάρκεια της εκτέλεσης. Οι βιβλιοθήκες μπορούν να περιέχουν ακόμα και νέες εντελώς μεθόδους χωρίς καμία επίπτωση σε προγράμματα που έχουν ήδη διανεμηθεί.

Μία διεπαφή (interface) είναι ένα σύνολο μεθόδων (το αντίστοιχο των συναρτήσεων στη C) που ένα αντικείμενο πρέπει να περιέχει αλλά δεν το ενδιαφέρει η υλοποίησή τους. Με αυτόν τον τρόπο μία κλάση που υλοποιεί (implements) μία διεπαφή, είναι σίγουρο ότι υλοποιεί και τις αντίστοιχες μεθόδους, που υπάρχουν σε αυτό. Σε αντίθεση με τις διεπαφές, η κληρονομικότητα (inheritance) περνάει στις κλάσεις - παιδιά πέρα από το σύνολο των μεθόδων και τις υλοποιήσεις τους στην κλάση - πατέρα (superclass). Η Java υποστηρίζει πολλαπλές διεπαφές αλλά απλή κληρονομικότητα στις κλάσεις της. Κατά συνέπεια, μία κλάση μπορεί να κληρονομεί τις μεθόδους μόνο ενός πατέρα, σε αντίθεση με τη C++, η οποία επιτρέπει πολλαπλή κληρονομικότητα, αλλά μπορεί να υλοποιεί οσοδήποτε διεπαφές. Οι διεπαφές εξασφαλίζουν μεγαλύτερη ευελιξία, και επαναχρησιμοποίηση αφού διασυνδέουν αντικείμενα ανάλογα με το τι πρέπει να κάνουν ή να υλοποιούν, παρά με το πώς τα υλοποιούν.

Επιπρόσθετα, οι κλάσεις κατά τη φάση της εκτέλεσης πάντα έχουν την αντιπροσώπευσή τους (representation). Υπάρχει μία κλάση και ένα ή περισσότερα στιγμιότυπα (instances) της που περιέχουν πληροφορίες της εκτέλεσης. Αν προσπαθήσει κανείς να αποκτήσει ένα δείκτη σε αυτό το στιγμιότυπο είναι πάντα γνωστός ο τύπος του αντικειμένου στο οποίο δείχνει. Αντίθετα με τη C++, όπου ο προγραμματιστής δεν ξέρει αυτόν τον τύπο και κάνει αυθαίρετα προσαρμογή τύπου (type cast), χωρίς καμία εγγύηση ότι είναι και το σωστό, η Java κάνει πάντα έλεγχο τύπων κατά τη μεταγλώττιση και την εκτέλεση, έτσι ώστε να αποφευχθούν τέτοιες προβληματικές καταστάσεις. Η διαφορά μεταξύ της Java και της C++, όσον αφορά στο χαρακτηριστικό των τύπων, είναι ότι στη μεν πρώτη ο χρήστης εμπιστεύεται το μεταφραστή ότι δεν θα περάσει κανένα λάθος τύπων, στη δε δεύτερη, ο μεταφραστής εμπιστεύεται τον προγραμματιστή ότι δε θα του περάσει λάθος προσαρμογή τύπου.

Πρακτικές χρήσεις της Java

Η Java μπορεί να χρησιμοποιηθεί στα ακόλουθα πεδία εφαρμογών:

- Δικτυακές εφαρμογές οποιουδήποτε είδους, που τρέχουν σε οποιοδήποτε λειτουργικό (από e-mail και web servers και clients μέχρι Application Servers)
- Εφαρμογές για βάσεις δεδομένων
- Δυναμικές Web σελίδες (Java Server Pages)
- Εφαρμογές με γραφικά περιβάλλοντα (Graphical User Interfaces) που δεν έχουν τίποτα να ζηλέψουν από αντίστοιχες φτιαγμένες με άλλες γλώσσες προγραμματισμού.
- Υπηρεσίες κινητής τηλεφωνίας (WAP). Μπορείτε να κάνετε πολύ εύκολα μία εφαρμογή που θα προσπελαίνεται από το Internet μέσω του κινητού σας.
- Εφαρμογές για κάθε είδους μικροσυσκευή. Σε λίγο θα μπορείτε να γράφετε ένα calculator και να το αποθηκεύσετε στο κινητό σας.
- Εφαρμογές ασφάλειας (JavaRing, και Smart Cards που τρέχουν Java)
- Δυνατότητα εκτέλεσης και συνεργασίας με απομακρυσμένο κώδικα. Είναι δυνατόν το πρόγραμμά σας να χρησιμοποιεί μία κλάση η οποία είναι αποθηκευμένη σε ένα άλλο απομακρυσμένο μηχάνημα (RMI), ή να χρησιμοποιεί κλάσεις που είναι γραμμένες σε άλλη γλώσσα προγραμματισμού (CORBA).
- Αναγνώριση φωνής
- Διαχείριση XML
- Δισδιάστατα και Τρισδιάστατα γραφικά
- Εφαρμογές Τεχνητής Νοημοσύνης

Συμπέρασμα

Η γλώσσα προγραμματισμού Java εισάγει μία σειρά από ισχυρά εργαλεία που κάνουν τη δουλειά του προγραμματιστή πιο εύκολη και πιο ενδιαφέρουσα, ανοίγοντάς του νέες προοπτικές. Είναι μία αντικειμενοστραφής γλώσσα στην οποία δεν υπάρχουν οι δείκτες που τόσο ταλαιπωρούσαν μέχρι σήμερα τους προγραμματιστές. Η αυτόματη περισυλλογή «απορριμμάτων» (garbage collection), εξασφαλίζει τη σωστή διαχείριση της μνήμης χωρίς να απασχολείται ο χρήστης με ένα τέτοιο ζήτημα ρουτίνας. Η Java είναι επίσης μία γλώσσα ανεξάρτητη αρχιτεκτονικής των μηχανημάτων στα οποία τρέχει, πράγμα που την κάνει ιδανική για ένα ανομοιογενές δικτυακό περιβάλλον όπως είναι το Διαδίκτυο.

Κεφάλαιο 4: BibTeX μορφή

Το BibTeX[1] είναι ένα σχετικό με το LaTeX[18] εργαλείο (μορφή αρχείου και πρόγραμμα) για τον χειρισμό της βιβλιογραφίας που αναπτύχθηκε από τον Oren Patashnik και τον Leslie Lamport το 1985 για το σύστημα προετοιμασίας LaTeX εγγράφου . Η μορφή είναι εξ ολοκλήρου βασισμένη σε χαρακτήρες , έτσι μπορεί να χρησιμοποιηθεί από οποιαδήποτε πρόγραμμα. Θεωρείται η πιο κοινή μορφή για βιβλιογραφίες στο Internet.

Δομή της BibTeX καταχώρησης

Αρχίζουμε με ένα παράδειγμα:

```
@book{companion,  
author = "Goossens, Michel and Mittelbach, Franck and Samarin, Alexander",  
title = "The {{\LaTeX}} {C}ompanion",  
publisher = "Addison-Wesley",  
year = 1998  
}
```

Η γενική μορφή είναι η ακόλουθη:

```
@entry_type{internal_key,  
field_1 = "value_1",  
field_2 = "value_2",  
  
...  
field_n = "value_n"  
}
```

Μια καταχώρηση αποτελείται από:

- Έναν τύπο καταχώρησης, στο παράδειγμά μας αυτό είναι το άρθρο(book)
- Ένα κλειδί, στο παράδειγμά μας αυτό είναι το companion
- Διάφορα πεδία που αποτελούνται από
 - ο ένα όνομα, στο παράδειγμα μας author
 - ο μια τιμή, στο παράδειγμά μας Goossens, Michel and Mittelbach, Franck and Samarin, Alexander

Δυνατά πεδία

Ο πίνακας παρακάτω περιγράφει το ρόλο των πεδίων που χρησιμοποιούνται σε μια εγγραφή BibTex.

address	Γενικά η πόλη ή η πλήρης διεύθυνση του εκδότη.
Author	Για τα ονόματα του συγγραφέα. Η μορφή εισαγωγής είναι αρκετά ειδική, δεδομένου ότι το BibTeX πρέπει να είναι ικανό να διακρίνει μεταξύ των πρώτων(μικρό) και τελευταίων ονομάτων(επίθετο).
booktitle	Για τον τίτλο ενός βιβλίου ένα μέρος του οποίου αναφέρεται.
Chapter	Ο αριθμός του κεφαλαίου (ή οποιουδήποτε μέρους) του βιβλίου που αναφέρεται. Εάν όχι ένα κεφάλαιο, ο τύπος πεδίου που πιθανόν να χρησιμοποιηθεί για την ορθότητα /εκτίμηση του τύπου τμηματοποίησης.
crossref	Αυτός είναι αρκετά ιδιαίτερο. Χρησιμοποιείται για να παραπέμψει μέσα στη βιβλιογραφία. Παραδείγματος χάριν, εσείς πιθανόν να αναφέρετε ένα έγγραφο, και ένα μέρος από αυτό. Σε αυτήν την περίπτωση, το δεύτερος μπορεί να παραπέμψει στο πρώτο, ή να κληρονομήσει τουλάχιστον μερικά από τα πεδία του πρώτου.
edition	Ο αριθμός έκδοσης. Ή στην πραγματικότητα είναι αυτό σχετικά με την σειρά έκδοσης του, παραδείγματος χάριν edition = " First ".Αυτό πιθανόν να προκαλέσει προβλήματα κατά την προσπάθεια να εξαχθεί μια βιβλιογραφία σε άλλη γλώσσα.
editor	Το όνομα του συντάκτη/ εκδότη (ων) της καταχώρησης. Η μορφή είναι η ίδια όπως για τους συγγραφείς
howpublished	Χρησιμοποιείται μόνο σε σπάνιες περιπτώσεις όπου το έγγραφο που αναφέρεται δεν είναι ένας κλασσικός τύπος όπως ένα @book, ένα @article ή μια δημοσίευση @inproceedings.
institution	Για μια τεχνική αναφορά, το όνομα του ιδρύματος που το δημοσίευσε.
journal	Το όνομα του περιοδικού στο οποίο το αναφερόμενο άρθρο έχει δημοσιευθεί.
key	Χρησιμοποιείται για τον καθορισμό της ετικέτας, σε περίπτωση που δεν μπορεί να υπολογιστεί από το BibTeX.Αυτό δεν αποσπά την ετικέτα, αλλά καθορίζει την ετικέτα όταν το BibTeX χρειάζεται μια αλλά δεν μπορεί να την υπολογίσει
month	Λοιπόν... Ο μήνας κατά τη διάρκεια του οποίου το έγγραφο έχει δημοσιευθεί. Αυτό επίσης προκαλεί το πρόβλημα της μετάφρασης της βιβλιογραφίας: Είναι καλύτερα να έχει μια αριθμητική τιμή, ή μια συντόμευση, αντί του πλήρους ονόματος του μήνα. Η κατοχή του αριθμού θα επέτρεπε επίσης στο BibTeX να ταξινομήσει τις

	καταχωρήσεις περισσότερο ακριβή (ακόμα κι αν, από όσο είναι γνωστό κανένα στυλ βιβλιογραφίας δεν κάνει αυτό προς το παρόν).
note	Για οποιαδήποτε πρόσθετα στοιχεία θα θέλατε να προσθέσετε. Δεδομένου ότι οι κλασσικές μορφές γράφτηκαν το 1985, δεν έχουν έναν πεδίο url, και η σημείωση χρησιμοποιείται συχνά για αυτό τον σκοπό, μαζί με το πακέτο url.sty.
number	Ένας αριθμός... Όχι οποιοσδήποτε, αλλά ο αριθμός μιας αναφοράς. Για τους αριθμούς όγκου/ πλήθους, ένα ειδικό πεδίο volume υπάρχει.
organization	Το ίδρυμα που οργανώνει μιας διάσκεψη.
pages	Οι σχετικές σελίδες του εγγράφου. Χρήσιμο για τον αναγνώστη όταν αναφέρετε σε ένα τεράστιο βιβλίο .
publisher	Το ίδρυμα που δημοσίευσε το έγγραφο.
school	Για τις διατριβές, το όνομα του σχολείου στο οποίο η διατριβή έχει προετοιμαστεί.
series	Το όνομα μιας συλλογής σειρών ή των βιβλίων.
title	Ο τίτλος του αναφερόμενου εγγράφου. Υπάρχουν μερικοί κανόνες που παρατηρούνται κατά την εισαγωγή αυτού του πεδίου.
Type	Ο τύπος δημοσίευσης, εάν είναι απαραίτητο. Για διατριβή, παραδείγματος χάριν, προκειμένου να διακρίνει μεταξύ μιας master διατριβής και ενός PHD. Ή ο τύπος του αναφερόμενου τμήματος.
volume	Ο αριθμός τόμου σε μια σειρά ή μια συλλογή των βιβλίων.
year	Το έτος δημοσίευσης.

Δυνατοί τύποι καταχώρησης

Ο πίνακας παρακάτω περιγράφει τους διαφορετικούς τύπους καταχώρησης.

Τύπος καταχώρησης	Υποχρεωτικά πεδία	Προαιρετικά πεδία
@article : Ένα άρθρο που δημοσιεύεται σε ένα περιοδικό.	author, title, year, journal.	volume, number, pages, month, note.
@book : Ένα βιβλίο.	author ή editor, volume ή number, title, publisher, year.	series, address, edition, month, note.
@booklet : Ένα μικρό βιβλίο, το οποίο δεν έχει κανένα πεδίο publisher.	title.	author, howpublished, address, address, month, year, note.
@conference : Άρθρο που προέκυψε στα πρακτικά (proceedings) μιας διάσκεψης, μιας συνεδρίαση...	author, title, booktitle, year.	editor, volume ή number, series, pages, address, month, organization, publisher, note.
@inbook : Μέρος (γενικά ένα κεφάλαιο) από ένα βιβλίο.	author ή editor, title, chapter ή pages.	volume, number, series, type, address, edition, month, note.
@incollection : Μέρος ενός βιβλίου που έχει δικό του τίτλο.	author, title, booktitle, publisher, year.	editor, volume ή number, series, type, chapter, pages, address, edition, month, note.
@inproceedings	Ίδια με το @conference.	
@manual : Ένα μικρό εγχειρίδιο.	Title.	author, organization, year, address, edition, month, note.
@mastersthesis : Κύρια διατριβή, ή κάτι ισοδύναμο.	author, title, school, year.	type, address, month,note.
@misc : Όταν τίποτα άλλο δεν ταιριάζει...	Τουλάχιστον ένα από τα προαιρετικά πεδία.	author, title, howpublished, year, month, note.

@phdthesis : Διατριβή PHD, ή κάτι παρόμοιο.	author, title, school, year.	type, address, month, note.
@proceedings : Πρακτικά διασκέψεων.	title, year.	editor, volume ou number, series, address, month, organization, publisher, note.
@techreport :Τεχνική αναφορά, που δημοσιεύεται από ένα εργαστήριο, ένα ερευνητικό κέντρο...	author, title, institution, year.	type, address, number, month, note.
@unpublished : Ένα έγγραφο που δεν έχει δημοσιευθεί. Μοιάζει πολύ με το @misc, αλλά το author και το title απαιτούνται εδώ.	author, title, note.	month, year.

Κεφάλαιο 5: Παρόμοια εφαρμογή

Εισαγωγή

Bibster

(Βιβλιογραφικό σύστημα Peer-to-Peer βασισμένο στην Σημασιολογία)

Το Bibster[19] απευθύνεται στους ερευνητές που μοιράζονται βιβλιογραφικά μεταδεδομένα. Την περίοδο που επινοήθηκε, πολλοί ερευνητές στην επιστήμη της πληροφορικής κρατούσαν καταλόγους βιβλιογραφικών μεταδεδομένων σε μορφή BibTeX, τους οποίους έπρεπε με κόπο να τους διατηρούν, μιας και επρόκειτο για χειρωνακτική διαδικασία. Και επιπλέον δεν είχαν διαθέσιμη μια πρακτικά εύκολη επισκόπηση αυτών των βιβλιογραφικών τους δεδομένων.

Ερευνητές που διατηρούσαν εκατοντάδες KB βιβλιογραφικών πληροφοριών, στα πολυάριθμα BibTeX αρχεία τους, έδειχναν την προθυμία τους να διαμοιραστούν αυτούς τους πόρους, υπό τον όρο ότι δεν θα ήταν απαραίτητο να επενδύσουν εργασία για να πραγματοποιήσουν αυτόν τον στόχο.

Τα ακόλουθα χαρακτηριστικά κάνουν αυτήν την εφαρμογή μια ενδιαφέρουσα περίπτωση χρήσης για ένα P2P σύστημα που βασίζεται στην σημασιολογία (semantics-based Peer-to-Peer):

- Μια μη κατανεμημένη λύση δεν υπάρχει και δεν μπορεί να υπάρξει, εξαιτίας του μεγάλου αριθμού των μη επίσημων εργαστηρίων στα οποία οι ερευνητές αναφέρονται. Επιπρόσθετα οποιοσδήποτε τέτοιος μη κατανεμημένος πόρος θα κάλυπτε μια περιορισμένη επιστημονική κοινότητα.
- Η χρήση της τεχνολογίας του Σημασιολογικού Ιστού (Semantic Web) είναι καθοριστική σε αυτό το περιβάλλον. Μολονότι μια οντολογία κοινού πυρήνα βιβλιογραφικών πληροφοριών υπάρχει (title, author/editor, κλπ.), πολλή από αυτήν την πληροφορία είναι ευμετάβλητη και οι χρήστες καθορίζουν αυθαίρετες προσθήσεις, για παράδειγμα για να εισάγουν τα URL των δημοσιεύσεων, για να εισάγουν περιλήψεις, ιδιαίτερα σχόλια, κλπ.

Η βιβλιογραφική περιοχή του συστήματος Bibster βάσει σχεδίου έχει διάφορα χαρακτηριστικά :

- Οι ερευνητές αν θελήσουν μπορούν να ψάξουν για βιβλιογραφικές καταχωρήσεις χρησιμοποιώντας μια απλή λέξη κλειδί, αλλά και πιο προηγμένες, σημασιολογικές αναζητήσεις, π.χ. για δημοσιεύσεις ενός ειδικού τύπου, με τις συγκεκριμένες τιμές ιδιοτήτων, ή για ένα ιδιαίτερο θέμα ή σχετικά θέματα.
- Οι ερευνητές μπορεί αν θελήσουν να ρωτήσουν έναν απλό συγκεκριμένο ομότιμο μέλος (π.χ. ένας κεντρικός υπολογιστής με όλα τα βιβλιογραφικά μεταδεδομένα από μια συγκεκριμένη διάσκεψη), ένα συγκεκριμένο σύνολο ομότιμων μελών (π.χ. όλα τα ομότιμα μέλη που είναι γνωστό ότι έχουν τις

πληροφορίες για ένα δεδομένο θέμα), ή ολόκληρο δίκτυο των ομότιμων μελών (για να λάβει τη μέγιστη ανάκληση στην τιμή χαμηλής ακρίβειας)

- Οι ερευνητές αν θελήσουν μπορούν να ενσωματώσουν τα αποτελέσματα μιας ερώτησης σε μια τοπική βάση δεδομένων γνώσης για μελλοντική χρήση. Τέτοια δεδομένα μπορεί στη συνέχεια να χρησιμοποιηθούν για να δώσουν απάντηση στις ερωτήσεις από άλλα ομότιμα μέλη. Μπορούν επίσης να υλοποιήσουν την ενημέρωση των στοιχείων που είναι ήδη τοπικά αποθηκευμένα με τις πρόσθετες πληροφορίες για αυτά τα αποκτηθέντα στοιχεία από άλλα ομότιμα μέλη.
- Οι οντολογίες που χρησιμοποιούνται μέσα στο Bibster είναι “μικρής σπουδαιότητας (lightweight)”: απλές ταξινομήσεις των όρων χωρίς πολύ περαιτέρω τυποποιημένη έννοια.

Ρόλος Οντολογιών

Η χρήση των οντολογιών παίζει σημαντικό ρόλο σε όλα τα στάδια του Bibster: εισαγωγή στοιχείων, διατύπωση ερωτήσεων, δρομολόγηση ερωτήσεων, και επεξεργασία απαντήσεων.

1. Το σύστημα επιτρέπει στους χρήστες να εισάγουν τα βιβλιογραφικά μεταδεδομένα τους σε μια τοπική αποθήκη(βάση). Αυτά τα βιβλιογραφικά μεταδεδομένα γίνονται διαθέσιμα κάτω από δύο κοινές οντολογίες: η πρώτη οντολογία (SWRC) περιγράφει διαφορετικές γενικές πτυχές των βιβλιογραφικών μεταδεδομένων (και θα μπορούσε να ισχύει σε πολλές διαφορετικές ερευνητικές περιοχές), η δεύτερη οντολογία (ιεραρχία θέματος ACM) περιγράφει συγκεκριμένες κατηγορίες βιβλιογραφίας για την περιοχή πληροφορικής. Οι βιβλιογραφικές καταχωρήσεις που τίθενται στην διάθεση του Bibster από έναν χρήστη ταξινομούνται αυτόματα κάτω από αυτές τις δύο οντολογίες. Και οι οντολογίες και το συγκεκριμένο στιγμιότυπο βιβλιογραφικών δεδομένων αναπαρίστανται σε RDF.
2. Οι χρήστες μπορούν να στείλουν ερωτήσεις σε άλλα ομότιμα μέλη ψάχνοντας για βιβλιογραφικά μεταδεδομένα. Αυτές οι ερωτήσεις διατυπώνονται από την άποψη των δύο αυτών οντολογιών: οι ερωτήσεις μπορούν να έχουν σχέση με πεδία όπως τον συγγραφέα, τον τύπο δημοσίευσης κ.λπ. (χρησιμοποιώντας τους όρους από την SWRC οντολογία) ή οι ερωτήσεις μπορεί να αφορούν συγκεκριμένους όρους της πληροφορικής (χρησιμοποίηση της ACM Ιεραρχία θέματος). Αυτές οι ερωτήσεις του χρήστη μεταφράζονται σε ερώτηση RDF γλώσσας SeRQL που απαντιέται από διαφορετικά ομότιμα μέλη στο δίκτυο.
3. Αυτές οι ερωτήσεις πρέπει να δρομολογηθούν διαμέσου του ομότιμου δικτύου, και πάλι εδώ οι οντολογίες διαδραματίζουν έναν κρίσιμο ρόλο. Οι ερωτήσεις δρομολογούνται μέσω του δικτύου εξαρτώμενες από τα πρότυπα πείρας των ομότιμων μελών. Ένα τέτοιο πρότυπο πείρας περιγράφει πάνω

σε ποιες έννοιες από την οντολογία ACM ένα ομότιμο μέλος μπορεί να απαντήσει. Μια συνάρτηση ταιριάσματος καθορίζει πόσο πολύ το σημασιολογικό περιεχόμενο μιας ερώτησης ταιριάζει με το πρότυπο πείρας του κάθε ομότιμου μέλους. Η δρομολόγηση γίνεται έπειτα βάσει αυτής της σημασιολογικής ταξινόμησης.

4. Τέλος, λαμβάνονται οι απαντήσεις για μια ερώτηση. Λόγω της κατανομημένης φύσης και του ενδεχομένως μεγάλου μεγέθους του Peer-to-Peer δικτύου, αυτό το σύνολο απαντήσεων να είναι πολύ μεγάλο, και να περιέχει πολλές διπλές απαντήσεις. Λόγω της ημι-δομημένης φύσης των βιβλιογραφικών μεταδεδομένων, τέτοια διπλότυπα δεν είναι συχνά ακριβώς ίδια αντίγραφα.

Πάλι σε αυτό το βήμα, εκμεταλλεύεται τις οντολογίες, για να μετρήσει αυτή τη φορά την σημασιολογική ομοιότητα μεταξύ των διαφορετικών απαντήσεων, και για να διαγράψει τα προφανή αντίγραφα όπως καθορίζονται από την λειτουργία ομοιότητας.

Το Σύστημα Bibster

Το σύστημα Bibster έχει υλοποιηθεί ως περίπτωση της αρχιτεκτονικής του συστήματος SWAP. Το σχήμα 1 παρουσιάζει ένα υψηλού επιπέδου σχέδιο της αρχιτεκτονικής από έναν απλό κόμβο στο peer-to-peer σύστημα. Στην συνέχεια παρουσιάζονται εν συντομία τα επιμέρους συστατικά όπως αρχικοποιήθηκαν για το σύστημα Bibster.

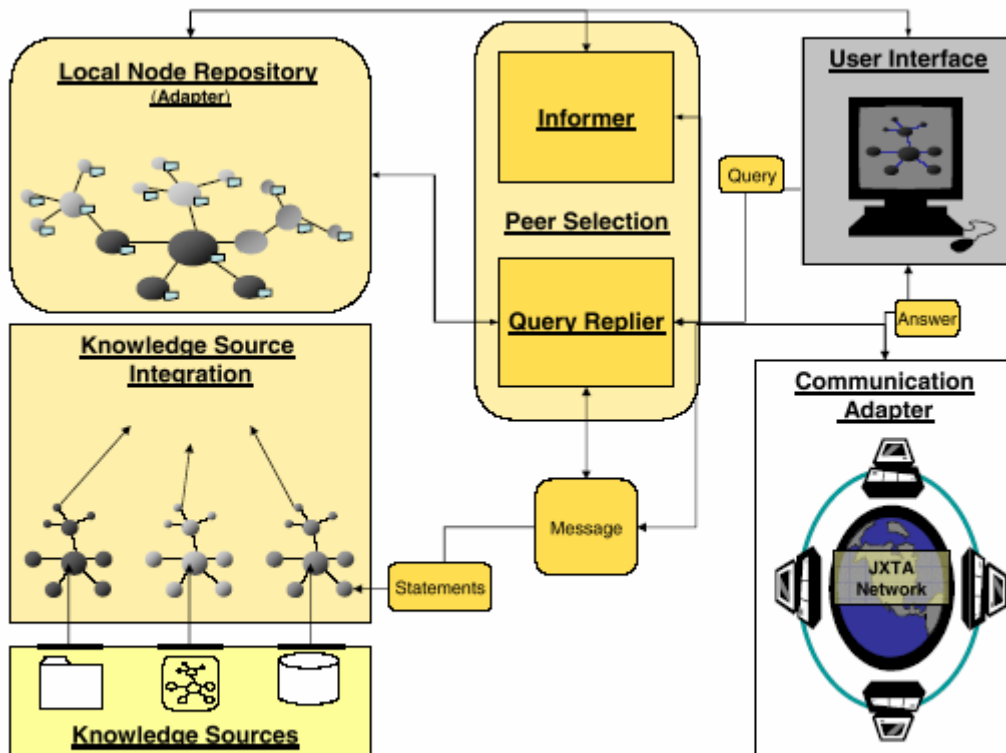


Fig. 1. SWAP system architecture

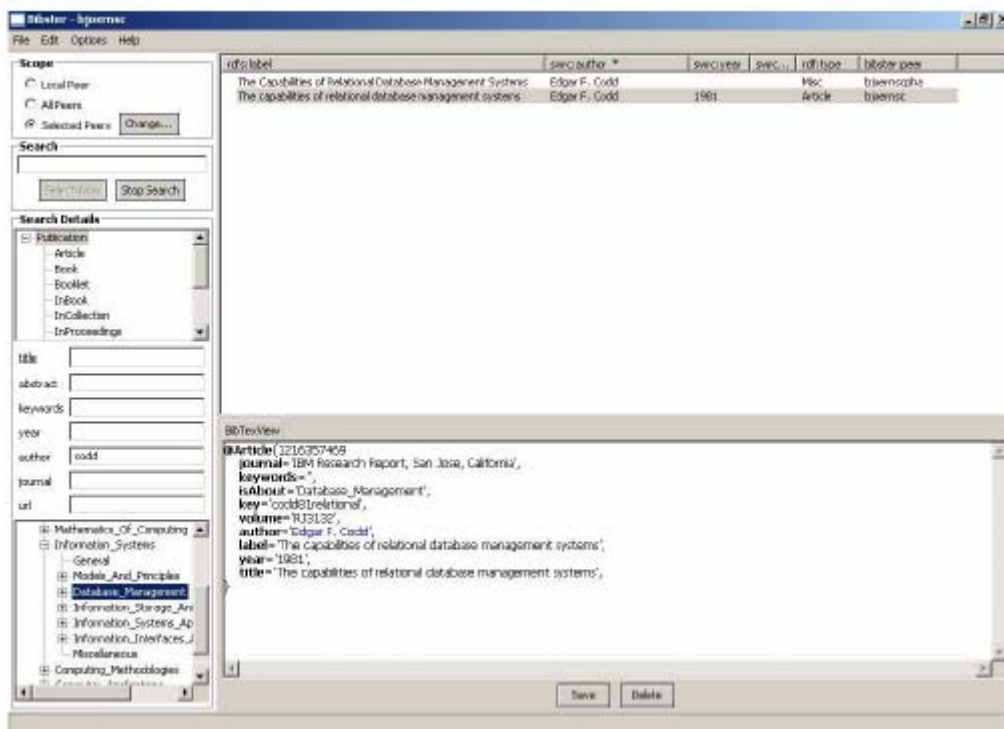


Fig. 2. User interface for the Bibster application

Προσαρμογέας επικοινωνίας (Communication Adapter). Αυτό το συστατικό είναι αρμόδιο για την δικτυακή επικοινωνία μεταξύ των ομότιμων μελών. Χρησιμοποιεί ως ένα επίπεδο μεταφοράς για τα άλλα μέρη του συστήματος, για την αποστολή

και την προώθηση των ερωτήσεων. Κρύβει όλες τις χαμηλού επιπέδου λεπτομέρειες επικοινωνίας από το υπόλοιπο του συστήματος. Στη συγκεκριμένη εφαρμογή του συστήματος Bibster χρησιμοποιείτε το JXTA ως πλατφόρμα επικοινωνίας.

Οι πηγές γνώσης(knowledge sources).Οι πηγές γνώσης στο σύστημα Bibster είναι πηγές από βιβλιογραφικά μεταδεδομένα, όπως τα BibTeX αρχεία που αποθηκεύεται τοπικά στο σύστημα αρχείων του χρήστη.

Ο ολοκληρωτής πηγής γνώσης(Knowledge Source Integrator). Ο ολοκληρωτής πηγής γνώσης είναι αρμόδιος για την εξαγωγή και την ολοκλήρωση των εσωτερικών και εξωτερικών πηγών γνώσης στην αποθήκη τοπικού κόμβου(Local Node Repository).

Αποθήκη τοπικού κόμβου(Local Node Repository). Προκειμένου να ρυθμιστούν τα πρότυπα των πληροφοριών και οι όψεις του όπως και οι πληροφορίες που αποκτιούνται από το δίκτυο, κάθε ομότιμο μέλος διατηρεί ένα εσωτερικό λειτουργικό μοντέλο που αποθηκεύεται στην αποθήκη τοπικού κόμβου. Αυτό το μοντέλο παρέχει την ακόλουθη λειτουργικότητα:

- Μεσολαβεί μεταξύ των όψεων και των αποθηκευμένων πληροφοριών
- Υποστηρίζει τη διατύπωση και την επεξεργασία ερώτησης
- Διευκρινίζει τη διεπαφή του ομότιμου μέλους στο δίκτυο
- Παρέχει την βάση για την ομότιμη ταξινόμηση και επιλογή

Στο σύστημα Bibster, η αποθήκη τοπικού κόμβου είναι βασισμένη στη αποθήκη RDF(S) Sesame. Η γλώσσα διατύπωσης ερωτήσεων SeRQL χρησιμοποιείται για να διατυπώσει σημασιολογικές ερωτήσεις προς στην αποθήκη τοπικού κόμβου.

Ο πληροφοριοδότης(Informer).Ο στόχος του πληροφοριοδότη είναι να διαφημίσει ως προ-δραστηριότητα (proactively) την διαθέσιμη γνώση ενός ομότιμου μέλους στο peer-to-peer δίκτυο και να ανακαλύψει τα ομότιμα μέλη με γνώση που μπορεί να είναι σχετική για την απάντηση στις ερωτήσεις του χρήστη. Αυτό πραγματοποιείται με την αποστολή διαφημίσεων για την πείρα ενός ομότιμου μέλους. Στο σύστημα Bibster, αυτές οι περιγραφές πείρας περιέχουν ένα σύνολο θεμάτων για τα οποία το ομότιμο μέλος είναι ειδικό. Τα ομότιμα μέλη μπορούν να δεχτούν αυτές τις διαφημίσεις, δημιουργώντας έτσι μια σημασιολογική σύνδεση με το άλλο ομότιμο μέλος. Αυτές οι σημασιολογικές συνδέσεις σχηματίζουν μια σημασιολογική τοπολογία, η οποία είναι η βάση για την ευφυή δρομολόγηση ερώτησης.

Διαχειριστής απαντήσεων ερώτησης(Query Replier).Ο διαχειριστής απαντήσεων ερώτησης είναι το τμήμα συντονισμού που ελέγχει την διαδικασία κατανομής ερωτήσεων. Λαμβάνει τις ερωτήσεις από την διεπαφή του χρήστη και τις διανέμει σύμφωνα με το περιεχόμενο της ερώτησης. Όταν το ομότιμο μέλος λαμβάνει μια ερώτηση από ένα άλλο ομότιμο μέλος, αυτό προσπαθεί να την απαντήσει ή να την προωθήσει. Η απόφαση σε ποια ομότιμα μέλη μια ερώτηση θα μπορούσε να σταλθεί παίρνεται βασισμένη στη γνώση για την πείρα των άλλων ομότιμων μελών.

Διεπαφή χρήστη(User Interface). Η διεπαφή χρήστη, όπως παρουσιάζεται στο σχήμα 2 επιτρέπει στο χρήστη να εισάγει, να δημιουργήσει και να επεξεργαστεί τα βιβλιογραφικά μεταδεδομένα καθώς επίσης και να διατυπώσει ερωτήσεις με τρόπο διαισθητικό. Εκτός από τις ερωτήσεις βασισμένες σε απλή λέξη για όλες τις ιδιότητες, ο χρήστης μπορεί να διατυπώσει προηγμένες σημασιολογικές ερωτήσεις προς την SWRC οντολογία και την ιεραρχία θέματος ACM.

Επιπλέον, το πεδίο αναφοράς της ερώτησης μπορεί να καθοριστεί: Οι ερωτήσεις μπορούν να αξιολογηθούν στον τοπικό ομότιμο μέλος, στα επιλεγμένα ομότιμα μέλη, ή σε παγκόσμιο επίπεδο. Τα αποτελέσματα της ερώτησης, τα οποία απεικονίζονται σε έναν κατάλογο που ομαδοποιείται βάση των διπλότυπων, μπορούν να ενσωματωθούν στην τοπική αποθήκη ή να εξαχθούν σε μορφές όπως BibTeX και HTML.

Σημασιολογική εξαγωγή βιβλιογραφικών μεταδεδομένων

Μεγάλα ποσά βιβλιογραφικών μεταδεδομένων αποθηκεύονται σε BibTeX αρχεία. Πολλοί ερευνητές έχουν συγκεντρώσει εκτενείς συλλογές BibTeX αρχείων για τις βιβλιογραφικές τους αναφορές. Εντούτοις αυτά τα αρχεία είναι ημι-δομημένα και έτσι οι απλές ιδιότητες τους μπορεί ή όχι να ερμηνεύονται σωστά. Ένα άλλο πρόβλημα είναι ότι υπάρχουν μη καλά ορισμένες διεπαφές για την ανταλλαγή τυποποιημένων BibTeX αρχείων.

Για την ανταλλαγή βιβλιογραφικών δεδομένων σε ένα peer-to-peer βασισμένο στη σημασιολογία δίκτυο, πρέπει αυτό να αντιπροσωπεύεται με έναν δομημένο και μεθοδικό τρόπο. Η χρήση των τυποποιημένων αναπαραστάσεων είναι ουσιώδης για τον διαμοιρασμό της γνώσης με τα άλλα ομότιμα μέλη.

Το BibToOnto είναι ένα συστατικό του Bibster για την εξαγωγή σαφούς γνώσης βιβλιογραφικών στοιχείων. Το εργαλείο αναπτύχθηκε ειδικά για το πρόγραμμα Bibster. Το καθαρό BibTeX θα μετασχηματιστεί σε μια οντολογία βασισμένη στην αναπαράσταση γνώσης. Αυτός ο μετασχηματισμός χρησιμοποιείται για να δώσει νόημα στις δομές πληροφοριών που πρόκειται να ανταλλαχθούν μεταξύ των ομότιμων μελών.

Η οντολογία στόχος είναι η σημασιολογική οντολογία ερευνητικής Κοινότητας Παγκόσμιου Ιστού (SWRC, Semantic Web Research Community Ontology), η οποία σχηματίζει μεταξύ των άλλων μια ερευνητική κοινότητα, τους ερευνητές της, τα θέματα, τις δημοσιεύσεις, τα εργαλεία, και τις ιδιότητες μεταξύ αυτών. Η οντολογία SWRC καθορίζει μια διαμοιρασμένη και κοινή θεωρία περιοχής που βοηθά τους χρήστες και τις μηχανές να επικοινωνούν απλά μεταξύ τους και υποστηρίζει την ανταλλαγή σημασιολογίας.

Το BibToOnto ταξινομεί αυτόματα τις βιβλιογραφικές καταχωρήσεις σύμφωνα με την ιεραρχία θέματος ACM, που χρησιμοποιεί μια προσέγγιση βασισμένη σε μια απλή λέξη κλειδί. Επιπλέον, είναι δυνατό να αναταξινομηθούν οι καταχωρήσεις χειρωνακτικά στην διεπαφή χρήστη του Bibster.

Η ιεραρχία θέματος ACM έχει γίνει ένα τυποποιημένο σχήμα για την περιγραφή και την κατηγοριοποίηση της βιβλιογραφίας της επιστήμης της πληροφορικής.

Καλύπτει 1287 θέματα του πεδίου της επιστήμης της πληροφορικής. Εκτός από τις σχέσεις υπο- και υπερθέματα (supertopic), παρέχει επίσης πληροφορίες για σχετικά θέματα.

Το ακόλουθο παράδειγμα παρουσιάζει έναν μετασχηματισμό μιας καταχώρησης BibTeX σε ένα αντικείμενο βασισμένο στην οντολογία SWRC. Το αποτέλεσμα αναπαριστάται ως ένα RDF γράφημα στο σχήμα 3.

```

Παράδειγμα 1. @ARTICLE{codd81relational,
author = {Edgar F. Codd},
title = {The capabilities of relational database management systems},
journal = {IBM Research Report, San Jose, California},
volume = {RJ3132},
year = {1981}
}
    
```

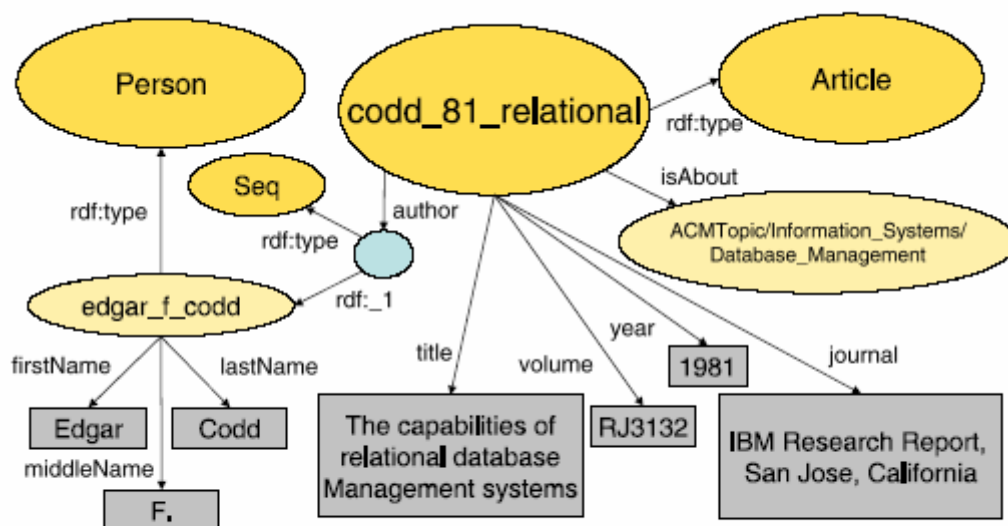


Fig. 3. SWRC Sample Metadata

Οι authors και οι editors αναπαριστώνται ως περιπτώσεις της swrc:κλάση Person. Μπορούν να προσδιορίζονται από ένα μοναδικό URI.

Η ίδια η δημοσίευση αρχικοποιείται ως swrc:Article που είναι μια υποκλάση της swrc:Publication. Η σειρά των authors εγγυάται από την χρήση των ακολουθιών RDF. Τα θέματα ACM που αντιστοιχούν στις δημοσιεύσεις αναπαρίστανται με τις swrc:isAbout ιδιότητες. Σε αυτό το παράδειγμα, το σχετικό θέμα είναι το Database Management.

Σημασιολογική ζήτηση πληροφοριών (Semantic Querying)

Κάθε ομότιμος κόμβος στο σύστημα Bibster έχει μια τοπική RDF αποθήκη, στην οποία και η τοπική γνώση και η γνώση που λαμβάνεται από άλλα ομότιμα μέλη αποθηκεύονται. Μια διεπαφή χρήστη επιτρέπει στους χρήστες να επεξεργαστούν,

να φυλλομετρήσουν και να εξετάσουν αυτήν την γνώση. Η de-facto τυποποιημένη γλώσσα διατύπωσης ερωτήσεων στο σύστημα είναι η SeRQL.

Η SeRQL (γλώσσα διατύπωσης ερωτήσεων Sesame RDF, προφέρεται "circle") είναι μια RDF(S) γλώσσα διατύπωσης ερωτήσεων που αναπτύχθηκε στα πλαίσια του προγράμματος SWAP για να παρουσιάσει τις πρακτικές απαιτήσεις από την κοινότητα χρήστη Sesame που δεν ικανοποιήθηκαν επαρκώς από άλλες γλώσσες διατύπωσης ερωτήσεων.

Επιλογή ομότιμου μέλους βασισμένη στην πείρα(Expertise Based Peer Selection)

Η επεκτασιμότητα ενός Peer-to-Peer δικτύου καθορίζεται ουσιαστικά από τον τρόπο του πώς οι ερωτήσεις διαδίδονται στο δίκτυο. Τα Peer-to-Peer δίκτυα που μεταδίδουν όλες τις ερωτήσεις σε όλα τα ομότιμα μέλη δεν επεκτείνουν την ευφυή δρομολόγηση ερώτησης και τις τοπολογίες δικτύου που απαιτούνται για να μπορούν να δρομολογηθούν οι ερωτήσεις σε ένα υποσύνολο ομότιμων μελών τα οποία είναι ικανά να απαντήσουν στις ερωτήσεις.

Σύγχρονα πρωτόκολλα δρομολόγησης όπως το Chord, το CAN και το Pastry επιτρέπουν την περίπλοκη δρομολόγηση βασισμένη στους κατανεμημένους δείκτες. Πρόσφατα, στο σημασιολογικό Πλαίσιο Παγκόσμιου Ιστού(Semantic Web context), τα Peer-to-Peer δίκτυα βασισμένα στο σχήμα , έχουν προκύψει ότι βασίζονται σε σύνθετες, με δυνατότητα επέκτασης , σημασιολογικές περιγραφές των πόρων αντί των καθορισμένων και περιορισμένων περιγραφών. Επιτρέπουν σύνθετες ερωτήσεις προς αυτά τα μεταδεδομένα αντί των ερωτήσεων βασισμένων σε απλή λέξη κλειδί. Το κόστος αναζήτησης με αυτόν τον τρόπο (από την άποψη των διαφορετικών κόμβων που αναζητούν και μεταδίδουν δεδομένα) για μια δεδομένη ερώτηση μειώνεται, αφού οι δείκτες σημασιολογικά σχετικών εγγράφων είναι πιθανό να τοποθετηθούν στην ίδια θέση στο δίκτυο.

Σημασιολογική ανίχνευση διπλότυπου(Semantic Duplicate Detection)

Κατά την αναζήτηση πληροφοριών στο δίκτυο Bibster αναμένετε να ληφθεί ένας μεγάλος αριθμός αποτελεσμάτων με έναν ενδεχομένως υψηλό αριθμό διπλότυπων, ακόμα κι αν η ίδια η ερώτηση είναι ήδη πολύ περιοριστική. Ο μεγάλος αριθμός αποτελεσμάτων οφείλεται στο γεγονός ότι δεν υπάρχει μια κεντρική αποθήκη αλλά ένα Peer-to-Peer δίκτυο. Επιπλέον, δεδομένου ότι τα μεταδεδομένα δημιουργούνται με έναν διανεμημένο τρόπο, η αναπαράσταση των μεταδεδομένων είναι πολύ διαφορετική και ενδεχομένως ακόμα και μη συνεπής. Για να κάνει εφικτή της λειτουργία ενός δυνατού, αποδοτικού και εύκολα χρησιμοποιήσιμου συστήματος φιλτράρει αυτά τα διπλότυπα. Συγκεκριμένα, δεν φέρνει το χρήστη αντιμέτωπο με έναν κατάλογο όλων των ξεχωριστών αποτελεσμάτων. Αλλά παρουσιάζει τα αποτελέσματα ερώτησης ομαδοποιημένα από τα σημασιολογικά διπλότυπα.

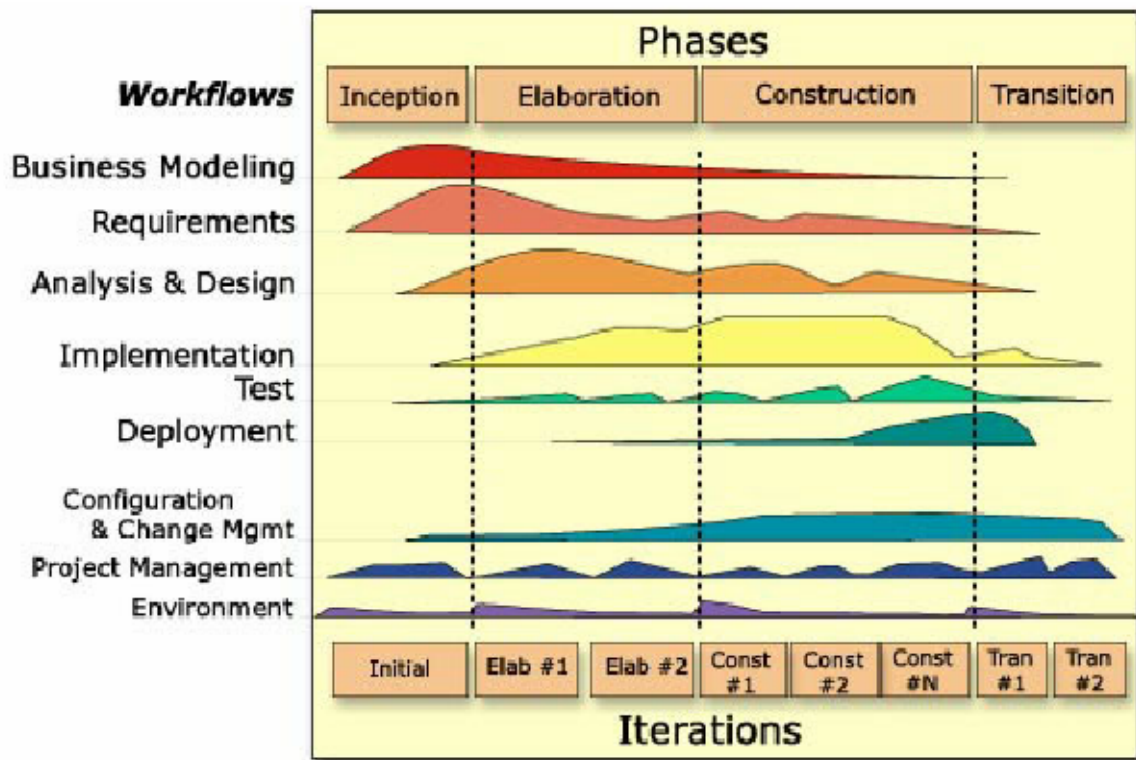
Κεφάλαιο 6: BibSource

Εισαγωγή

Το BibSource όπως αναφέρθηκε και στην εισαγωγή είναι ένα πρόγραμμα που υλοποιεί ένα P2P σύστημα ανταλλαγής βιβλιογραφικών πληροφοριών και συγκεκριμένα εγγραφών σε μορφή BibTeX.

Η ανάπτυξη του λογισμικού BibSource έγινε σύμφωνα με τη μεθοδολογία Rational Unified Process (RUP)[20]. Σύμφωνα με τη μεθοδολογία αυτή η ανάπτυξη του λογισμικού χωρίζεται σε τέσσερις φάσεις: την εναρκτήρια φάση, τη φάση επεξεργασίας τη φάση υλοποίησης και τη φάση μετάβασης - παράδοσης του λογισμικού. Σε κάθε φάση γίνονται πολλές επαναλήψεις. Σε κάθε επανάληψη εξετάζονται όλα τα στάδια ανάπτυξης του λογισμικού από την καταγραφή των απαιτήσεων, την ανάλυση τη σχεδίαση και την υλοποίηση.

Σχηματικά η διαδικασία RUP αναπαρίσταται στο παρακάτω διάγραμμα:

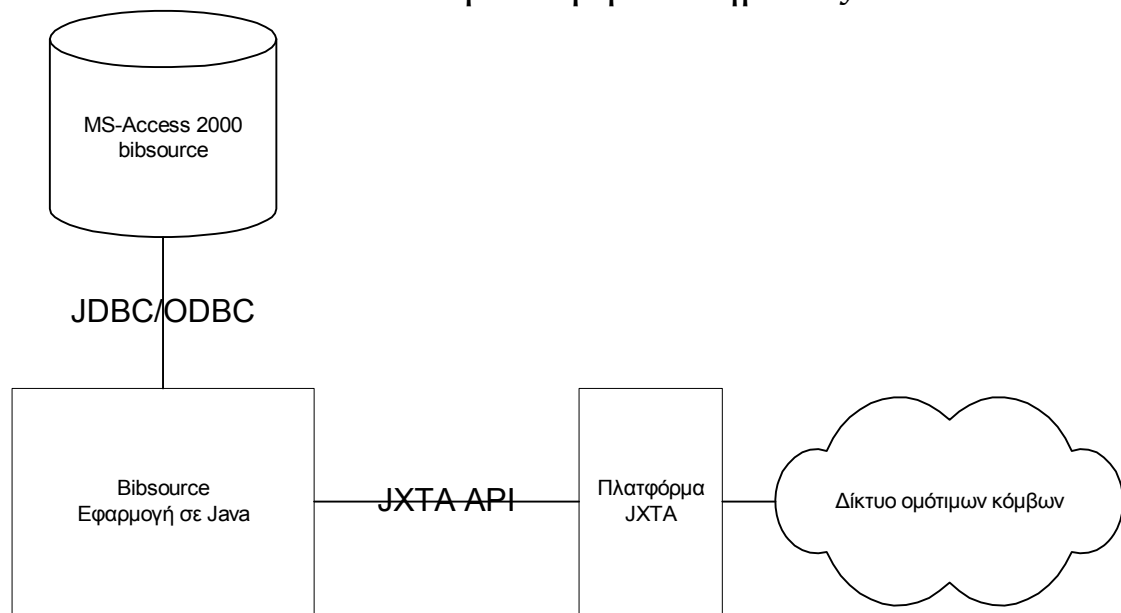


Σύμφωνα με τη διαδικασία αυτή παράγονται διάφορα σκευάσματα - παραδοτέα τα οποία αναφέρονται παρακάτω.

Περιγραφή περιβάλλοντος στο οποίο έγινε η ανάπτυξη

Η υλοποίηση της συγκεκριμένης εφαρμογής πραγματοποιήθηκε σε λειτουργικό σύστημα **Windows XP Professional**, στο προγραμματιστικό περιβάλλον του εργαλείου **NetBeans** με την γλώσσα **Java 1.4**.Ενώ για την πραγματοποίηση του P2P δικτύου χρησιμοποιήθηκε η **τεχνολογία JXTA**.Η υλοποίηση και η διαχείριση της βάσης δεδομένων που περιέχει τις καταχωρήσεις BibTeX έγινε **στην Microsoft Access 2002**.

Γενική άποψη συστήματος



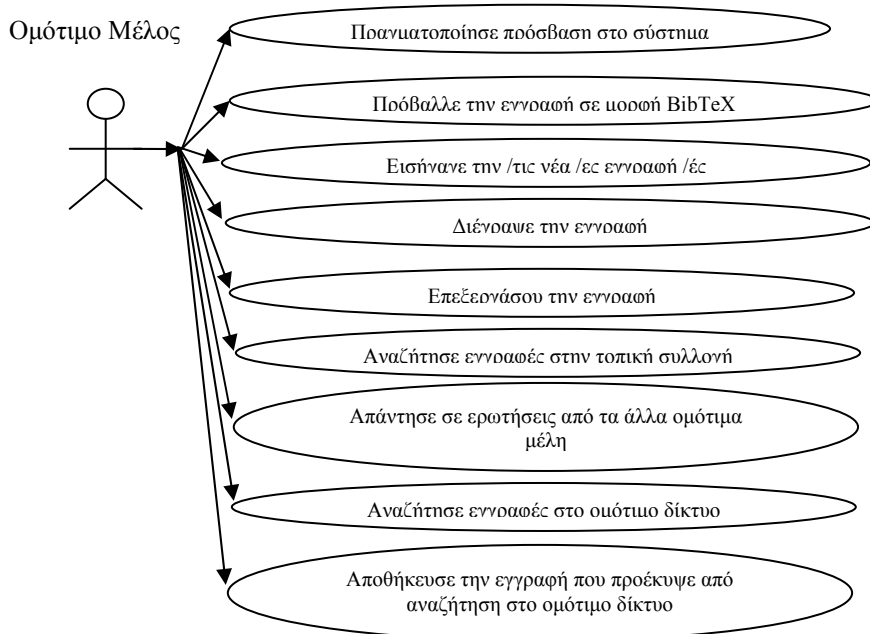
Καταγραφή απαιτήσεων

Η καταγραφή απαιτήσεων έγινε με το μέθοδο της καταγραφής περιπτώσεων χρήσης. Παράχθηκε ένα μοντέλο περιπτώσεων χρήσης (use case model) μαζί με μια αναλυτική περιγραφή των περιπτώσεων αυτών.

Γενική περιγραφή

Το BibSource είναι ένα λογισμικό που επιτρέπει τη διαχείριση βιβλιογραφικών εγγραφών τοπικά από χρήστες αλλά και την αναζήτηση παρόμοιων εγγραφών σε ένα peer-to-peer δίκτυο.

Διάγραμμα μοντέλου περιπτώσεων χρήσης



Επισκόπηση μοντέλου

Αρχικά το ομότιμο μέλος ενεργοποιεί την περίπτωση χρήσης "Πραγματοποίησε πρόσβαση στο σύστημα" για συνδεθεί με το σύστημα /ομότιμο δίκτυο. Μόλις επιτευχθεί η πρόσβαση ενεργοποιείται η περίπτωση χρήσης "Απάντησε σε ερωτήσεις από τα άλλα ομότιμα μέλη" για να μπορεί το ομότιμο μέλος να εκτελεί ερωτήσεις από άλλα ομότιμα μέλη. Στην διεπαφή χρήστη που θα του εμφανιστεί μπορεί να χρησιμοποιήσει την περίπτωση χρήσης "Πρόβαλλε την εγγραφή σε BibTeX" ή την "Εισήγαγε την/ τις νέα /ες εγγραφή /ές " ή την "Διέγραψε την εγγραφή" ή την "Επεξεργάσου την εγγραφή" για να προβάλλει μια εγγραφή σε μορφή BibTeX ή να εισάγει εγγραφές στην συλλογή του ή να διαγράψει ή να

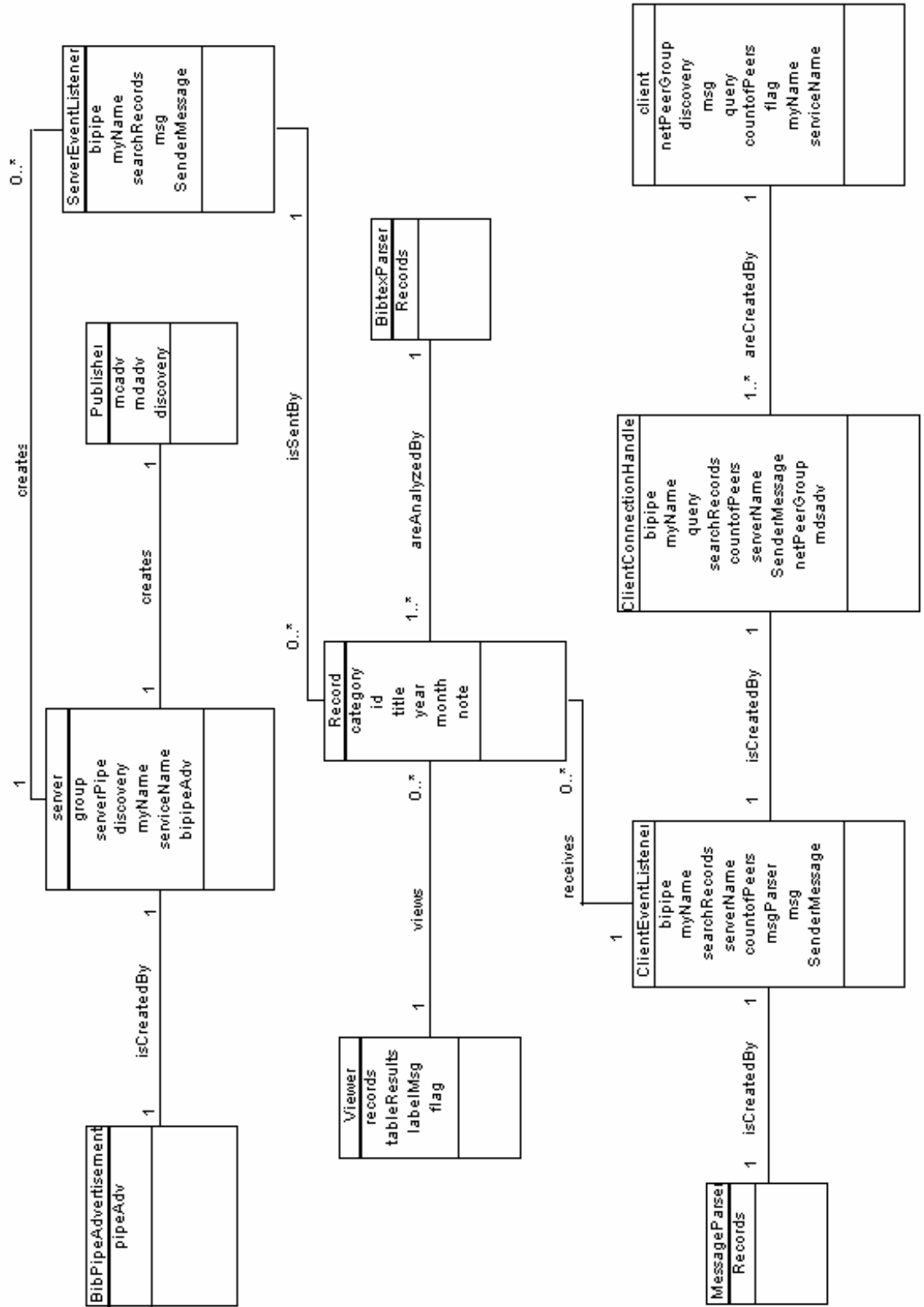
επεξεργαστεί κάποια συγκεκριμένη εγγραφή αντίστοιχα. Μπορεί να ενεργοποιήσει την περίπτωση χρήσης “Αναζήτησε εγγραφές στην τοπική συλλογή” ή την “Αναζήτησε εγγραφές στο ομότιμο δίκτυο” για να αναζητήσει εγγραφές από την τοπική συλλογή του ομότιμου μέλους ή από το ομότιμο δίκτυο αντίστοιχα. Στην δεύτερη περίπτωση μπορεί να χρησιμοποιήσει την “Αποθήκευσε την εγγραφή που προέκυψε από αναζήτηση στο ομότιμο δίκτυο ” για να αποθηκεύσει μια εγγραφή που είναι αποτέλεσμα της αναζήτησης αυτής.

Χαρακτήρες

Ο κύριος χαρακτήρας / ρόλος της εφαρμογής μας είναι το ομότιμο μέλος(peer) που αλληλεπιδρά με το σύστημα και με τον οποίο σχετίζονται και όλες οι περιπτώσεις χρήσεις που αναλύονται παρακάτω εκτενώς.

Στην επόμενη σελίδα απεικονίζεται το γενικό διάγραμμα των βασικών κλάσεων(Domain Model)

Domain Model



Αναλυτική περιγραφή περιπτώσεων χρήσης

1. “Πραγματοποίησε πρόσβαση στο σύστημα”

Σύντομη περιγραφή

Η συγκεκριμένη περίπτωση χρήσης χρησιμοποιείται από κάποιο ομότιμο μέλος του συστήματος για να επιτύχει παράλληλη πρόσβαση τόσο στο σύστημα όσο και στο ομότιμο δίκτυο.

Αρχική περιγραφή βήμα-βήμα

Ο χρήστης /ομότιμο μέλος θα πρέπει να έχει επιλέξει την εφαρμογή έτσι ώστε να αρχίσει να εκτελείται.

1.Αν είναι η πρώτη φορά που εκτελεί την εφαρμογή, εμφάνιση της κατάλληλης φόρμας

έτσι ώστε να αρχικοποιηθεί και διαμορφωθεί κατάλληλα η πλατφόρμα JXTA από το ομότιμο μέλος.

2.Εμφάνιση της αρχικής οθόνης για την εισαγωγή του ονόματος χρήστη(username) το οποίο θα χαρακτηρίζει το ομότιμο μέλος στο ομότιμο δίκτυο

3.Έλεγχος για κενή τιμή στο username

4.Αν δεν είναι κενό ,είσοδος στο σύστημα .

5.Εμφάνιση της κατάλληλης φόρμας με τις εγγραφές της συλλογής του μέλους.

Εναλλακτική ροή

1.1. Εμφάνιση της αρχικής οθόνης για την εισαγωγή του ονόματος χρήστη(username) το οποίο θα χαρακτηρίζει το ομότιμο μέλος στο ομότιμο δίκτυο

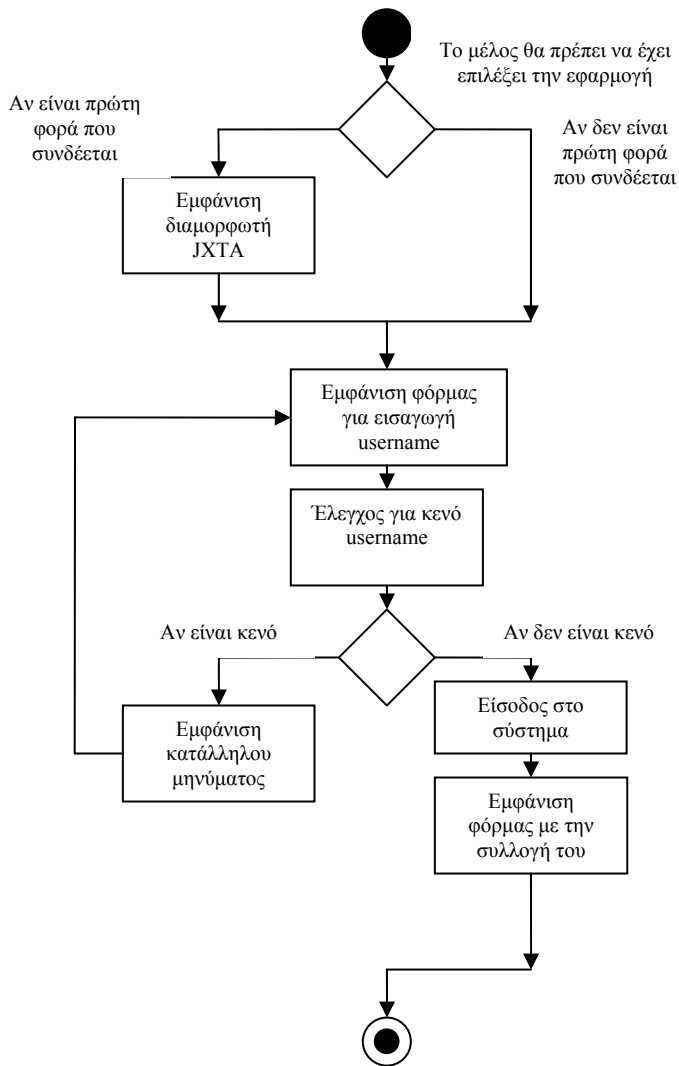
1.2. Έλεγχος για κενή τιμή στο username

1.3.Αν δεν είναι κενό, είσοδος στο σύστημα .

1.4.Εμφάνιση της κατάλληλης φόρμας με τις εγγραφές της συλλογής του μέλους

4&1.3.Εμφάνιση κατάλληλου μηνύματος

Διάγραμμα δραστηριότητας σε UML



Πρόσθετες επεξηγήσεις για την περίπτωση χρήσης πραγματοποιήσε πρόσβαση στο σύστημα

Πληροφορίες για την διαμόρφωση της πλατφόρμας JXTA

Κατά την διαμόρφωση της πλατφόρμας JXTA εμφανίζεται μια φόρμα με τρία Tabsheet στα οποία εισάγονται ειδικές πληροφορίες για το δικτυακό περιβάλλον του ομότιμου μέλους .Συγκεκριμένα:

- Στο tab Basic εισάγετε το όνομα του ομότιμου μέλους και ένας προσωπικός κωδικός.
- Στο tab Advanced διευκρινίζεται αν το ομότιμος μέλος είναι ομότιμος μέλος ραντεβού ή αναμετάδοσης καθώς και οι ρυθμίσεις του TCP και του HTTP πρωτοκόλλου. Οι εξερχόμενες συνδέσεις TCP (Enable outgoing connections) πρέπει να επιτρέπονται για τις περισσότερες καταστάσεις. Εάν δεν είστε πίσω από ένα firewall ή ένα NAT (Network Address Translation) που επιτρέπει σε μια συσκευή όπως ένας δρομολογητής να δρα ως ένας αντιπρόσωπος(agent) μεταξύ του διαδικτύου και του τοπικού δικτύου, οι εισερχόμενες συνδέσεις TCP πρέπει επίσης να ενεργοποιηθούν (Enable incoming connections), και δεν χρειάζεται να χρησιμοποιήσετε ένα ομότιμο μέλος αναμετάδοσης(relay peer). Εάν είστε πίσω από μια αντιπυρική ζώνη(firewall) ή ένα NAT ,οι εισερχόμενες συνδέσεις πρέπει να τεθούν εκτός λειτουργίας και ένας αναμεταδότης απαιτείται να οριστεί προκειμένου να επικοινωνήσει.
- Στο tab Rendezvous /Relays εισάγεται μια λίστα από τα ήδη γνωστά ομότιμα μέλη ραντεβού. Αν βρίσκεστε πίσω από ένα firewall ή NAT επιλέξτε Use a Relay.

2. “Πρόβαλλε την εγγραφή σε BibTeX”

Σύντομη περιγραφή

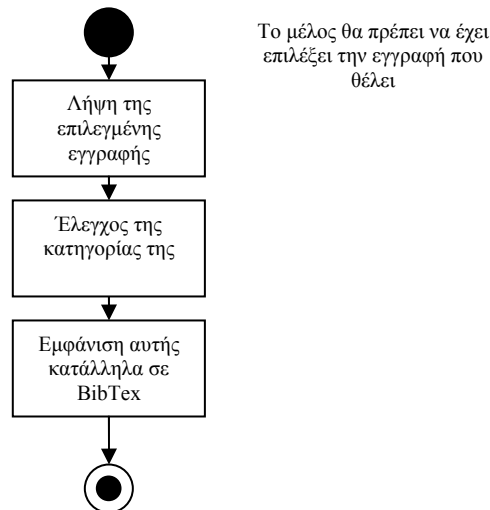
Η συγκεκριμένη περίπτωση χρήσης χρησιμοποιείται από κάποιο μέλος του συστήματος για να δει την BibTeX μορφή της εγγραφής που επιθυμεί.

Αρχική περιγραφή βήμα-βήμα

Ο χρήστης / μέλος θα πρέπει να έχει συνδεθεί με το σύστημα και να είναι στην κύρια φόρμα. Στον πίνακα με τις καταχωρήσεις να έχει επιλέξει αυτή που επιθυμεί.

- 1.Λήψη της επιλεγμένης εγγραφής
- 2.Έλεγχος της κατηγορίας /τόπου της
- 3.Εμφάνιση αυτής με τα κατάλληλα πεδία σε μορφή BibTeX

Διάγραμμα δραστηριότητας σε UML



3. “Εισήγαγε την /τις νέα /ες εγγραφή /ές ”

Σύντομη περιγραφή

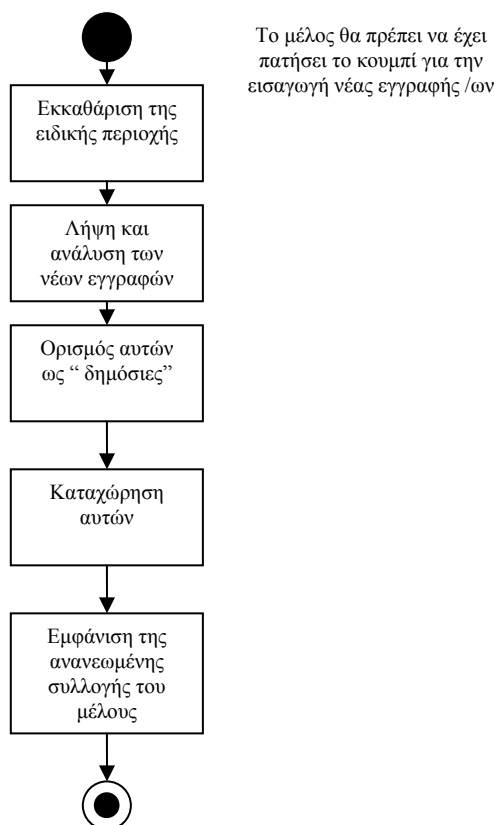
Η συγκεκριμένη περίπτωση χρήσης χρησιμοποιείται από κάποιο μέλος του συστήματος για να εισαγάγει νέα /ες εγγραφή /ές στην συλλογή του.

Αρχική περιγραφή βήμα-βήμα

Ο χρήστης /μέλος θα πρέπει να έχει συνδεθεί με το σύστημα και να είναι στην κύρια φόρμα και μάλιστα στο tab mycollection, και να έχει πατήσει το κουμπί για νέα εγγραφή /ές.

- 1.Εκκαθάριση της περιοχής που εμφανίζονται οι εγγραφές σε μορφή BibTeX
- 2.Λήψη και ανάλυση των εγγραφών σε μορφή BibTeX που έχει εισάγει ο χρήστης στην περιοχή προβολής των εγγραφών.
- 3.Ορισμός των νέων εγγραφών εξ ορισμού ως “δημόσιες”(published)
- 4.Καταχώρηση αυτών ανάλογα με τον τύπο καταχώρησης στον οποίο ανήκουν.
- 5.Εμφάνιση της ανανεωμένης συλλογής με τις νέες εγγραφές

Διάγραμμα δραστηριότητας σε UML



4. “Διέγραψε την εγγραφή”

Σύντομη περιγραφή

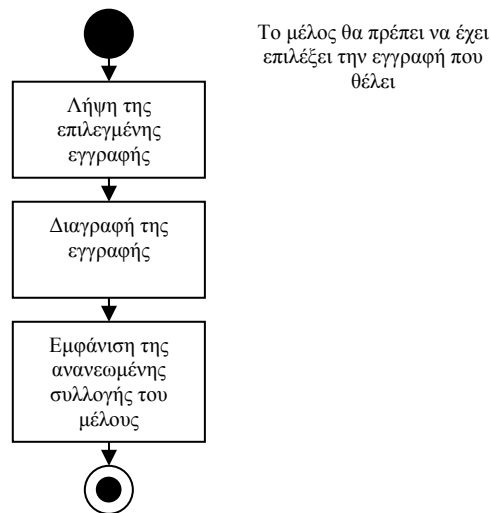
Η συγκεκριμένη περίπτωση χρήσης χρησιμοποιείται από κάποιο μέλος του συστήματος για να διαγράψει την εγγραφή που επιθυμεί.

Αρχική περιγραφή βήμα-βήμα

Ο χρήστης / μέλος θα πρέπει να έχει συνδεθεί με το σύστημα και να είναι στην κύρια φόρμα. Στον πίνακα με τις καταχωρήσεις να έχει επιλέξει αυτή που επιθυμεί.

- 1.Λήψη της επιλεγμένης εγγραφής
- 2.Διαγραφή εγγραφής
- 3.Εμφάνιση της ανανεωμένης συλλογής του ομότιμου μέλους.

Διάγραμμα δραστηριότητας σε UML



5. “Επεξεργάσου την εγγραφή”

Σύντομη περιγραφή

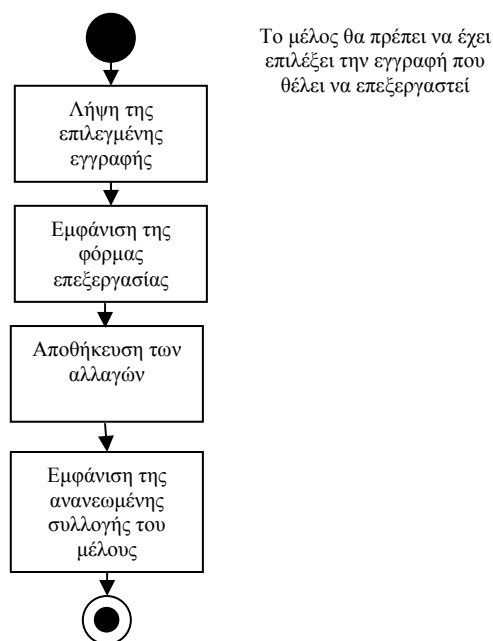
Η συγκεκριμένη περίπτωση χρήσης χρησιμοποιείται από κάποιο μέλος του συστήματος για να επεξεργαστεί τις τιμές των πεδίων μιας συγκεκριμένης εγγραφής.

Αρχική περιγραφή βήμα-βήμα

Ο χρήστης /μέλος θα πρέπει να έχει συνδεθεί με το σύστημα και να είναι στην κύρια φόρμα Στον πίνακα με τις καταχωρήσεις επιπλέον θα πρέπει να έχει επιλέξει την εγγραφή που θέλει να επεξεργαστεί.

- 1.Λήψη της επιλεγμένης εγγραφής
- 2.Εμφάνιση της φόρμας επεξεργασίας με τα δεδομένα της ανάλογα με τον τύπο καταχώρησης στον οποίο ανήκει η εγγραφή.
- 3.Αποθήκευση των αλλαγών.
- 3.Εμφάνιση της ανανεωμένης συλλογής του ομότιμου μέλους.

Διάγραμμα δραστηριότητας σε UML



6. “Αναζήτησε εγγραφές στην τοπική συλλογή”

Σύντομη περιγραφή

Η συγκεκριμένη περίπτωση χρήσης χρησιμοποιείται από κάποιο ομότιμο μέλος του συστήματος για να αναζητήσει και να βρει τις επιθυμητές εγγραφές από την συλλογή του.

Αρχική περιγραφή βήμα-βήμα

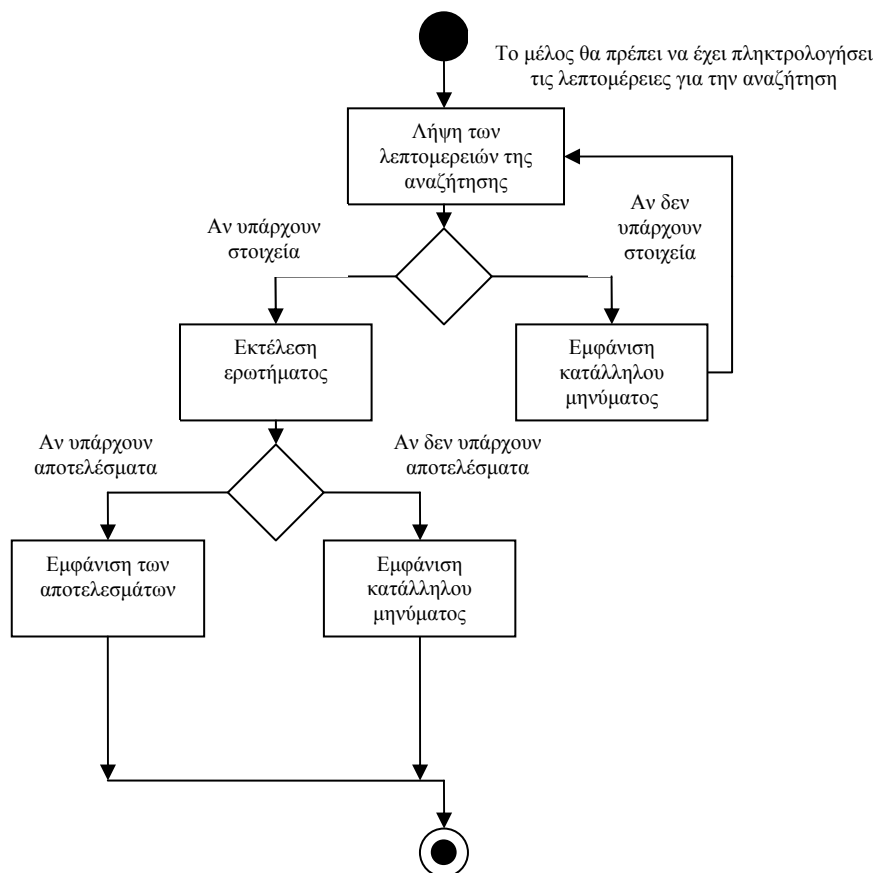
Ο χρήστης / μέλος θα πρέπει να έχει συνδεθεί με το σύστημα και να είναι στην κύρια φόρμα. Όπου και θα πρέπει πέρα από το να έχει επιλέξει αναζήτηση στην συλλογή του, να έχει πληκτρολογήσει τις πληροφορίες στα κατάλληλα πεδία που θα λειτουργήσουν ως συνθήκες στην αναζήτηση για μεγαλύτερη ακρίβεια

- 1.Λήψη των στοιχείων που συνθέτουν τις λεπτομέρειες της αναζήτησης.
- 2.Αν υπάρχουν εκτέλεση του ερωτήματος
- 3.Αν υπάρχουν αποτελέσματα εμφάνιση αυτών

Εναλλακτική ροή

2. Αν δεν υπάρχουν στοιχεία εμφάνιση κατάλληλου μηνύματος
3. Αν δεν υπάρχουν αποτελέσματα εμφάνιση κατάλληλου μηνύματος

Διάγραμμα δραστηριότητας σε UML



7. “Απάντησε σε ερωτήσεις από τα άλλα ομότιμα μέλη”

Σύντομη περιγραφή

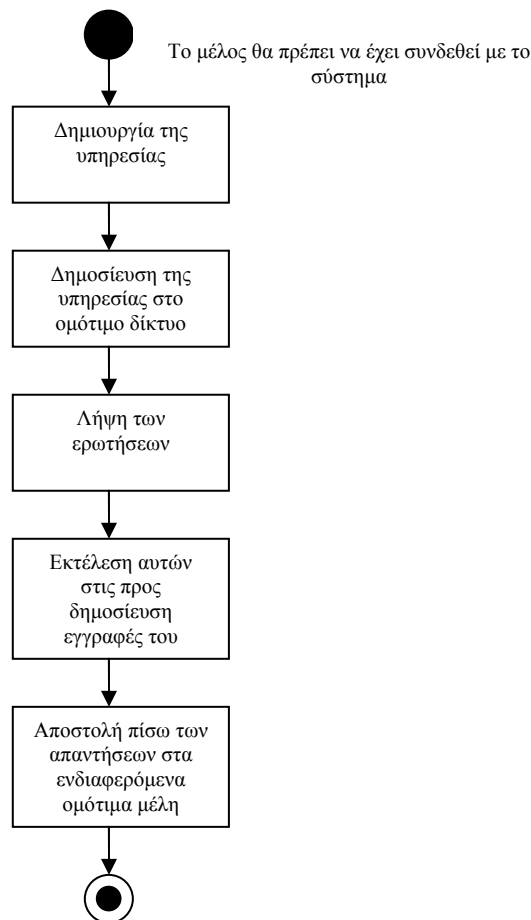
Η συγκεκριμένη περίπτωση χρήσης ενεργοποιείται από την εφαρμογή όταν ο χρήστης συνδέεται στο σύστημα και είναι υπεύθυνη για την λήψη και απάντηση των ερωτήσεων που δέχεται το ομότιμο μέλος από άλλα ομότιμα μέλη.

Αρχική περιγραφή βήμα-βήμα

Ο χρήστης / μέλος θα πρέπει να έχει συνδεθεί με το σύστημα .

1. Δημιουργία υπηρεσίας η οποία και περιέχει την διασωλήνωση από την οποία και θα επικοινωνήσουν τα ομότιμα μέλη με αυτό το μέλος.
2. Δημοσίευση της υπηρεσίας στο ομότιμο δίκτυο μέσω κάποιας ειδικής διαφήμισης.
3. Λήψη των ερωτήσεων
4. Εκτέλεση αυτών στις εγγραφές τις συλλογής του που έχουν χαρακτηριστεί ως “δημόσιες”
5. Αποστολή των απαντήσεων πίσω στα αιτηθέντα ομότιμα μέλη.

Διάγραμμα δραστηριότητας σε UML



8. “Αναζήτησε εγγραφές στο ομότιμο δίκτυο”

Σύντομη περιγραφή

Η συγκεκριμένη περίπτωση χρήσης χρησιμοποιείται από κάποιο ομότιμο μέλος του συστήματος για να αναζητήσει και να βρει τις επιθυμητές εγγραφές από ομότιμο δίκτυο.

Αρχική περιγραφή βήμα-βήμα

Ο χρήστης / μέλος θα πρέπει να έχει συνδεθεί με το σύστημα και να είναι στην κύρια φόρμα, όπου και θα πρέπει πέρα από το να έχει επιλέξει αναζήτηση στο ομότιμο δίκτυο ,να έχει πληκτρολογήσει τις πληροφορίες στα κατάλληλα πεδία που θα λειτουργήσουν ως συνθήκες στην αναζήτηση για μεγαλύτερη ακρίβεια

1.Λήψη των στοιχείων που συνθέτουν τις λεπτομέρειες της αναζήτησης.

2.Αν υπάρχουν διαγραφή των διαφημίσεων/ υπηρεσιών που έχει ανακαλύψει μέχρι τώρα.

3.Ανακάλυψη εκ νέου όλων των διαφημίσεων/ υπηρεσιών

4.Αποστολή στις ανακαλυμμένες υπηρεσίες-ομότιμα μέλη την ερώτηση

5.Λήψη των απαντήσεων από αυτά τα ομότιμα μέλη.

6.Αν υπάρχουν αποτελέσματα εμφάνιση αυτών.

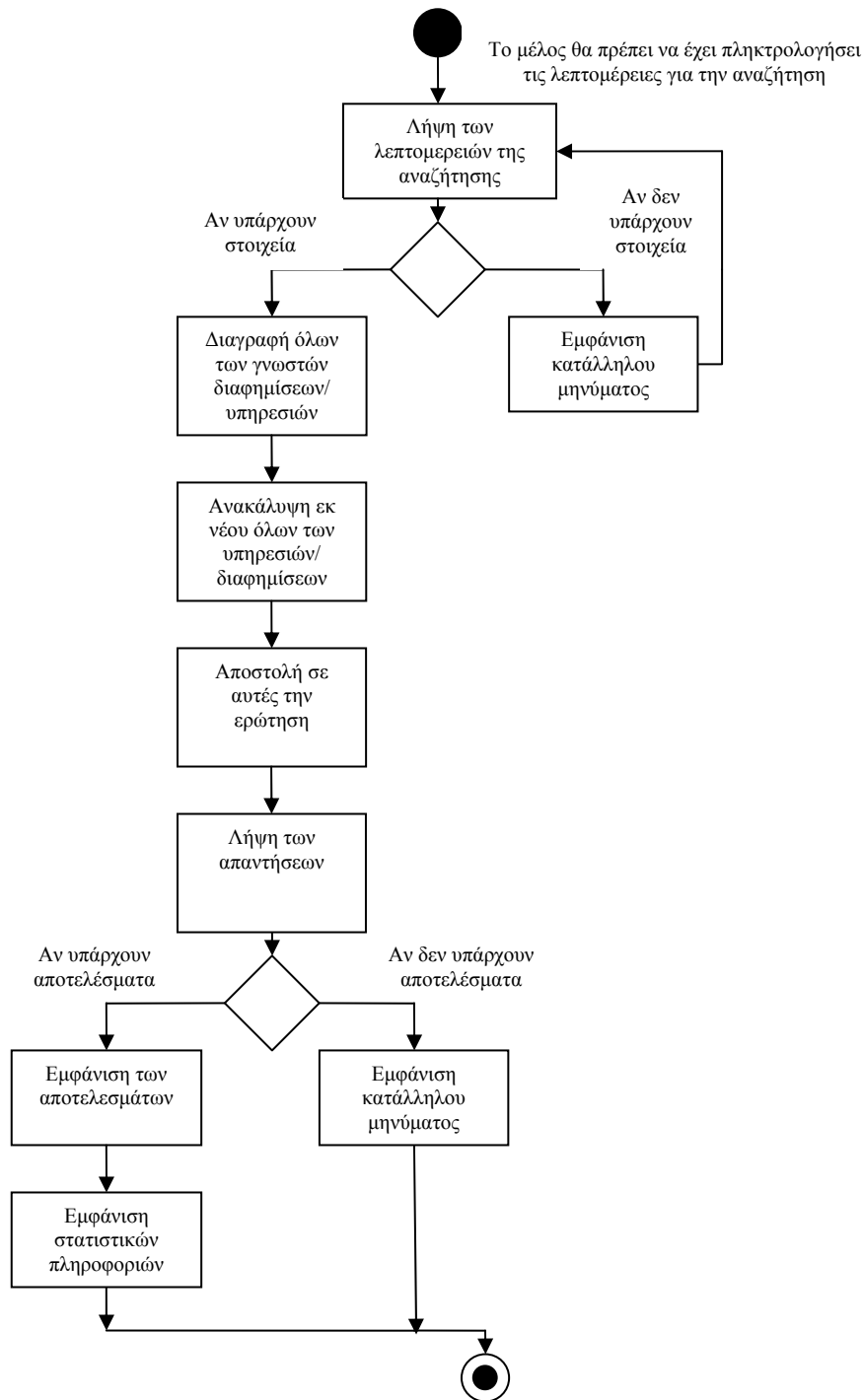
7.Εμφάνιση στατιστικών πληροφοριών σχετικά με τις απαντήσεις που λήφθηκαν.

Εναλλακτική ροή

2. Αν δεν υπάρχουν στοιχεία εμφάνιση κατάλληλου μηνύματος

6. Αν δεν υπάρχουν αποτελέσματα εμφάνιση κατάλληλου μηνύματος

Διάγραμμα δραστηριότητας σε UML



9. “Αποθήκευσε την εγγραφή που προέκυψε από αναζήτηση στο ομότιμο δίκτυο ”

Σύντομη περιγραφή

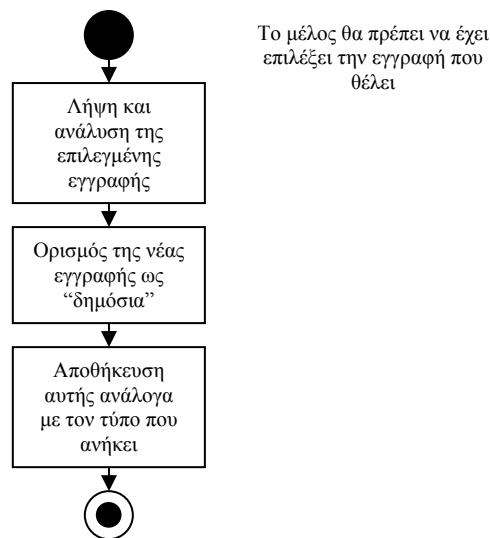
Η συγκεκριμένη περίπτωση χρήσης χρησιμοποιείται από κάποιο μέλος του συστήματος για να αποθηκεύσει κάποια εγγραφή στην τοπική του συλλογή που προέκυψε από μια αναζήτηση στο ομότιμο δίκτυο.

Αρχική περιγραφή βήμα-βήμα

Ο χρήστης / μέλος θα πρέπει να έχει πραγματοποιήσει μια αναζήτηση στο ομότιμο δίκτυο και να έχουν βρεθεί εγγραφές που εκπληρώνουν τις συνθήκες αναζήτησης . Στον πίνακα με τις καταχωρήσεις επιπλέον θα πρέπει να έχει επιλέξει την εγγραφή που επιθυμεί να αποθηκεύσει.

- 1.Λήψη και ανάλυση της επιλεγμένης εγγραφής
- 2.Ορισμός της νέας εγγραφής ως “δημόσια”
- 3.Αποθήκευση της εγγραφής στην τοπική συλλογή του ομότιμου μέλους αναλόγως με τον τύπο καταχώρησης στον οποίο ανήκει

Διάγραμμα δραστηριότητας σε UML



Ορισμός εγγραφής σε BNF

Ακολουθώς ορίζετε σε BNF μια εγγραφή σε BibTeX καθώς και όλοι οι περιορισμοί στην μορφή των δεδομένων για να είναι αποδεκτή από την εφαρμογή ως εξής:

Αλφάβητο

<letter> ::= a | b | ... | z | A | B | ... | Z

<digit> ::= 0 | 1 | ... | 9

<symbol> ::= . | * | ! | # | \$ | % | (|) | / | + | -

<type> ::= article | book | booklet | inbook | incollection | inproceedings | manual | mast
ersthesis | misc | phdthesis | proceedings | techreport | unpublished

<field> ::= address | author | booktitle | chapter | | crossref | edition | editor | howpublis
hed |

institution | journal | month | note | numberorganization | pages | publisher | school | s
eries | title | type | volume | year

Κανονική μορφή μιας εγγραφής BibTeX

@type{(letter | digit | symbol)+, \n

(field=((\ t¹(" ")⁺) | (" ")⁺){ | " } (letter | digit | symbol)(

letter | digit | symbol | { | } | " | \n)⁺(| ")(\n | \n)⁺

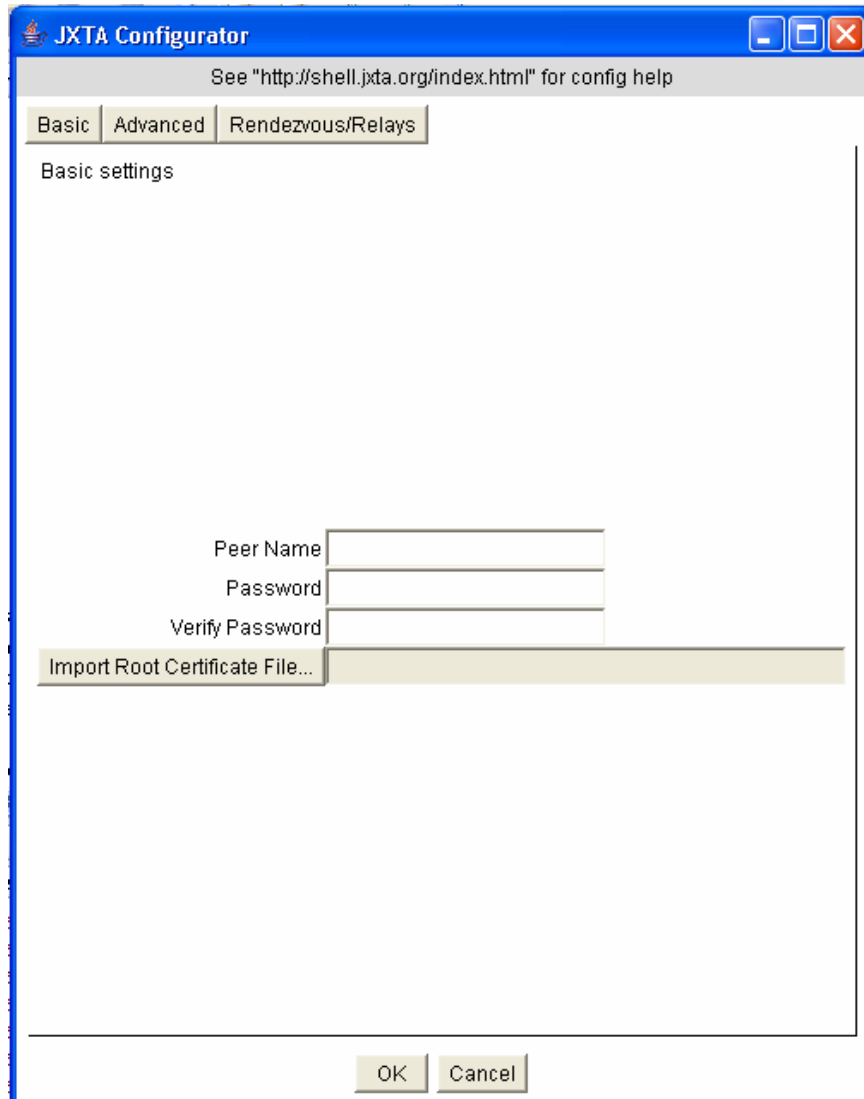
}

Αρχικά ο χρήστης θα πρέπει να δώσει τον χαρακτήρα @ ως έναρξη της νέας εγγραφής που ακολουθείται από τον τύπο της εγγραφής και ένα {. Στην συνέχεια είναι το κλειδί της εγγραφής που αποτελείται από ένα η περισσότερα γράμματα , αριθμούς ή και σύμβολα και που ακολουθείται από ένα κόμμα και μια αλλαγή γραμμής(\n) . Έπειτα αρχίζει μια ανάθεση μιας τιμής σε ένα πεδίο που ξεκινά από το όνομα του πεδίου, το = , ένα \t και ένα η περισσότερα κενά ή από ένα η περισσότερα κενά. Ακολουθεί ένα { ή ένα " , ένα γράμμα ή ψηφίο ή σύμβολο και ένα η περισσότερα γράμματα ή ψηφία ή σύμβολα ή { } ή " ή αλλαγές γραμμής που κλείνουν από ένα } ή ένα " . Τέλος έχουμε ένα κόμμα και μια αλλαγή γραμμής ή μια αλλαγή γραμμής . Τέτοιες αναθέσεις τιμών μπορούν να είναι μια ή περισσότερες. Η εγγραφή κλείνει με το χαρακτήρα }.

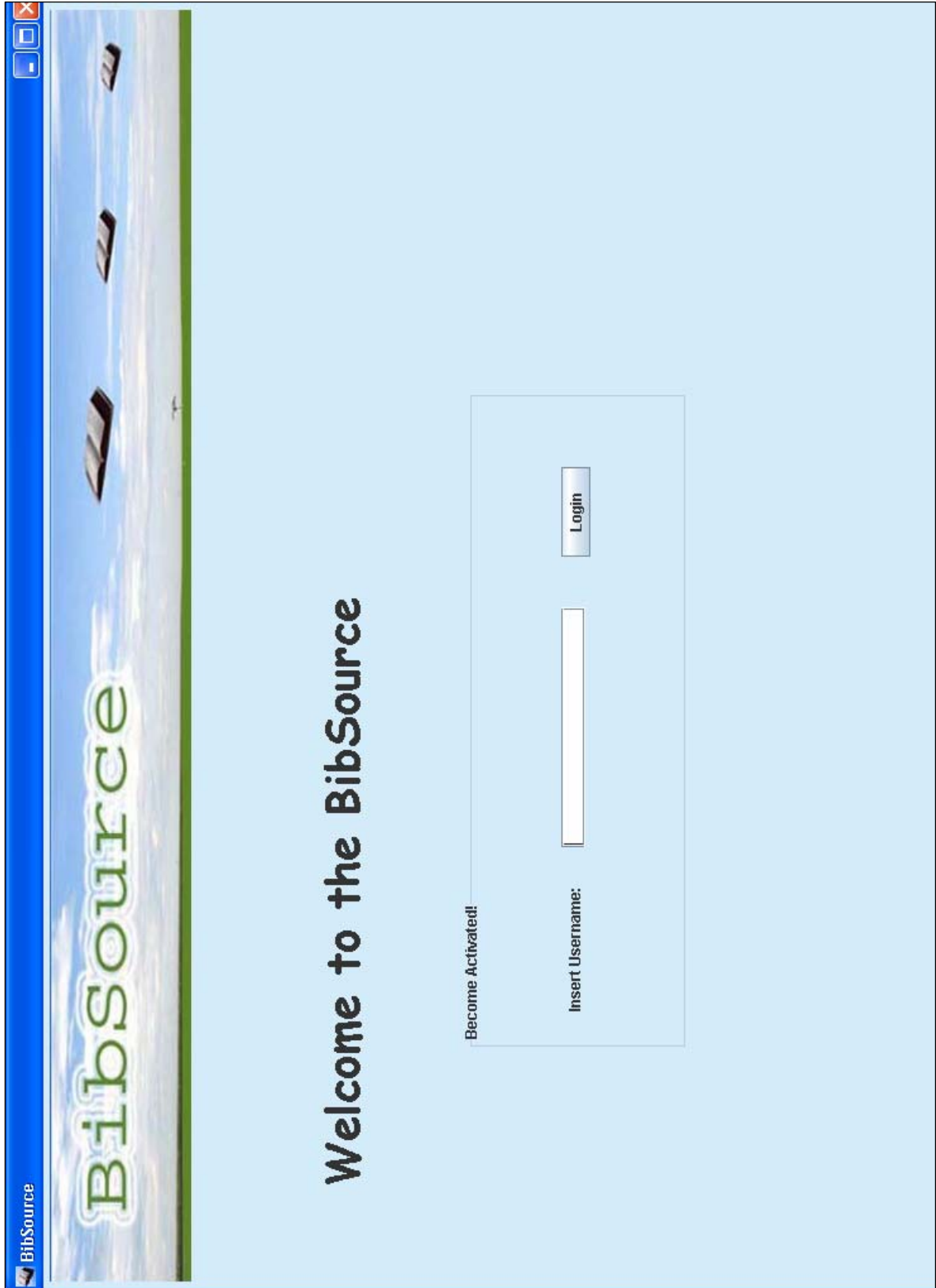
Δείγματα διεπαφών

Στην συνέχεια παρουσιάζονται οι βασικές διεπαφές της εφαρμογής μας

Διαμόρφωση πλατφόρμας JXTA



Εισαγωγική φόρμα



Κόρια φόρμα

The screenshot displays the BibSource web application interface. At the top left, there is a navigation bar with the BibSource logo and a 'Log Out' link. Below the navigation bar, the main content area is divided into several sections:

- Search Section:** Includes a search input field, 'Search' and 'Stop' buttons, and radio buttons for 'Local Collection' (selected) and 'All Peers'.
- Search Details Section:** Contains input fields for 'Title', 'Author', 'Key', and 'Year', along with a 'Category' dropdown menu set to 'All'.
- Results Section:** A table displaying search results with columns for 'Published', 'Key', 'Title', 'Author', 'Year', and 'Category'. The table contains 10 rows of data.
- Actions Section:** Located at the bottom right, it features three buttons: 'New', 'Edit', and 'Delete'.

Published	Key	Title	Author	Year	Category
<input type="checkbox"/>	ROFC:1	\LaTeX Example Manual	Russell L. Oer...	1988	manual
<input checked="" type="checkbox"/>	RSGD:1	Continuous-Time Control Model Vali...	Roy Smith and...	1996	Article
<input checked="" type="checkbox"/>	RSGD:1	Continuous-Time Control Model Vali...	Roy Smith and...	1996	Article
<input checked="" type="checkbox"/>	JS:1	Title of Article	Joseph W. Si...	1983	Incollection
<input checked="" type="checkbox"/>	AB:1	Title of Book	Author Book	1988	book
<input checked="" type="checkbox"/>	MWJCDL:1	Title of Book	Matthew Wrigh...	1988	inbook
<input checked="" type="checkbox"/>	AB:1	Title of Book	Author Book	1988	book
<input type="checkbox"/>	HC:1	Title of Booklet	Han Chen	1989	booklet
<input type="checkbox"/>	PR:1	Title of Conference Proceedings		1987	proceedings

Φόρμα επεξεργασίας εγγραφής

The screenshot displays the BibSource P2P application interface. At the top, there is a navigation bar with 'BibSource' and 'Log Out' buttons. Below this is a search bar with 'Peer:Peer1' and 'Search'/'Stop' buttons. A 'Target' section has radio buttons for 'Local Collection' (selected) and 'All Peers'. A 'Search Details' section includes input fields for 'Title', 'Author', 'Key', and 'Year', and a 'Category' dropdown menu set to 'All'. The main area shows a table of search results with columns for 'Published', 'Key', 'Title', 'Author', 'Year', and 'Category'. The first row is selected, showing 'ROPC:1', '{LaTeX} Example Manual', 'Russell L. Oer...', and '1988'. An 'Edit Bibliographical Entry' dialog box is open, displaying a table of properties and values for the selected entry. The dialog also includes 'Type: MANUAL', a 'Published' checkbox, and 'Save', 'Cancel', 'New', 'Edit', and 'Delete' buttons. Below the dialog, a code editor shows the BibTeX entry for the selected record.

Property	Value
Key	ROPC:1
Title	{LaTeX} Example Manual
Author	Russell L. Oertel and Pierce Cant...
Organization	Department of Electrical Engineer...
Address	College Station, Texas
Edition	
Month	September
Year	1988
Note	

```

@manual(ROPC:1,
  title =
    {LaTeX} Example Manual),
  author =
    (Russell L. Oertel and Pierce Cant...),
  organization =
    (Department of Electrical Engineer...),
  address =
    (College Station, Texas),
  month =
    (September),
  year =
    (1988)
)
  
```


Ανάλυση του συστήματος

Κλάσεις

Στο συγκεκριμένο σημείο γίνεται μια λεπτομερής αναφορά στις κλάσεις που προέκυψαν κατά την ανάλυση και σχεδίαση του συστήματος και χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής. Σύμφωνα με τη μεθοδολογία RUP από τις περιπτώσεις χρήσης και την αρχική καταγραφή των απαιτήσεων γίνεται προσπάθεια να αποφασιστούν οι βασικές κλάσεις που μετέχουν σε κάθε περίπτωση χρήσης. Κάθε μια από τις βασικές κλάσεις μπορεί να ανήκει σε μια από τρεις κατηγορίες. Είτε να είναι κλάση οντότητας οπότε ασχολείται με την αποθήκευση διάφορων στοιχείων που μπορεί να είναι χρήσιμα στο σύστημα, είτε κλάση ελέγχου οπότε ασχολείται με τον έλεγχο και το συντονισμό άλλων κλάσεων είτε κλάση ορίου οπότε ασχολείται είτε με την παρουσίαση δεδομένων στους χρήστες είτε με τη διασύνδεση με άλλα συστήματα.

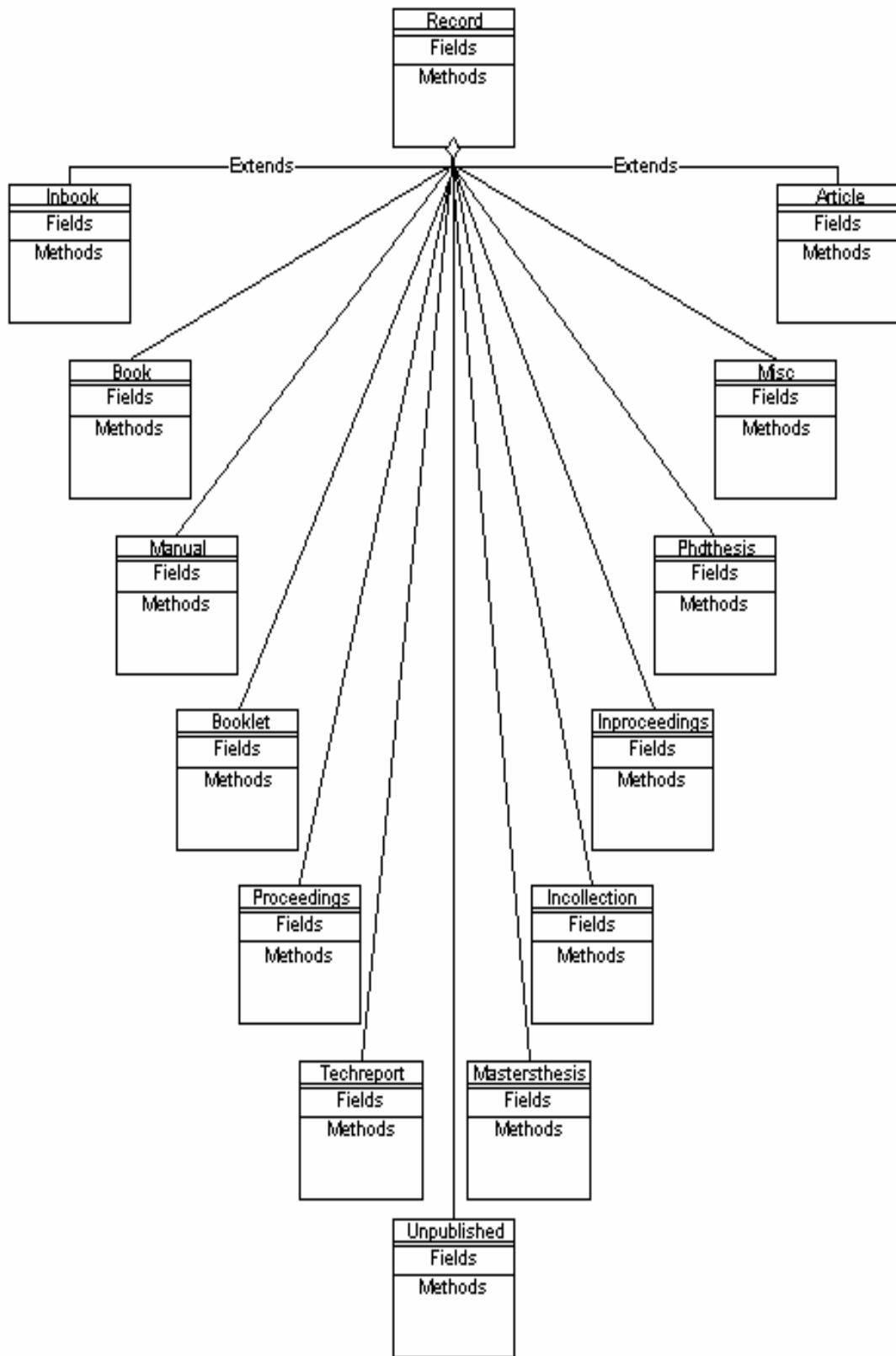
Προέκυψαν λοιπόν κατά την ανάλυση τα ακόλουθα:

Κλάσεις οντοτήτων

Η βασική κλάση οντότητας είναι η Record η οποία έχει σχέση γενίκευσης με κάθε μια από τις άλλες κλάσεις οντότητας που είναι στην ουσία οι άλλοι τύποι βιβλιογραφικής καταχώρησης.

Η κλάση Record περιέχει μόνο τα πεδία και τις συναρτήσεις που είναι κοινά μεταξύ όλων των υποκλάσεων.

Διάγραμμα κλάσεων οντοτήτων



Αναλυτική περιγραφή

Record
category:String id:int key:String title:String year:String month:String note:String status:String
String getΟνομα_πεδίου() void setΟνομα_πεδίου(String value) int setBaseValue(String field,String value) int setBaseValue(int field,String value)

Article
id:int author:String journal:String volume:String number:String pages:String
Vector getNames() String printBib() void insertValues(JTable jTableValues) void insertFields(JTable jTableValues) int setValue(String field,String value) int setValue(int field,String value)

Book
id:int author:String editor:String publisher:String volume:String number:String series:String address:String edition:String
Vector getNames() String printBib() void insertValues(JTable jTableValues) void insertFields(JTable jTableValues) int setValue(String field,String value) int setValue(int field,String value)

Booklet
id:int author:String howpublished:String address:String
Vector getNames() String printBib() void insertValues(JTable jTableValues) void insertFields(JTable jTableValues) int setValue(String field,String value) int setValue(int field,string value)

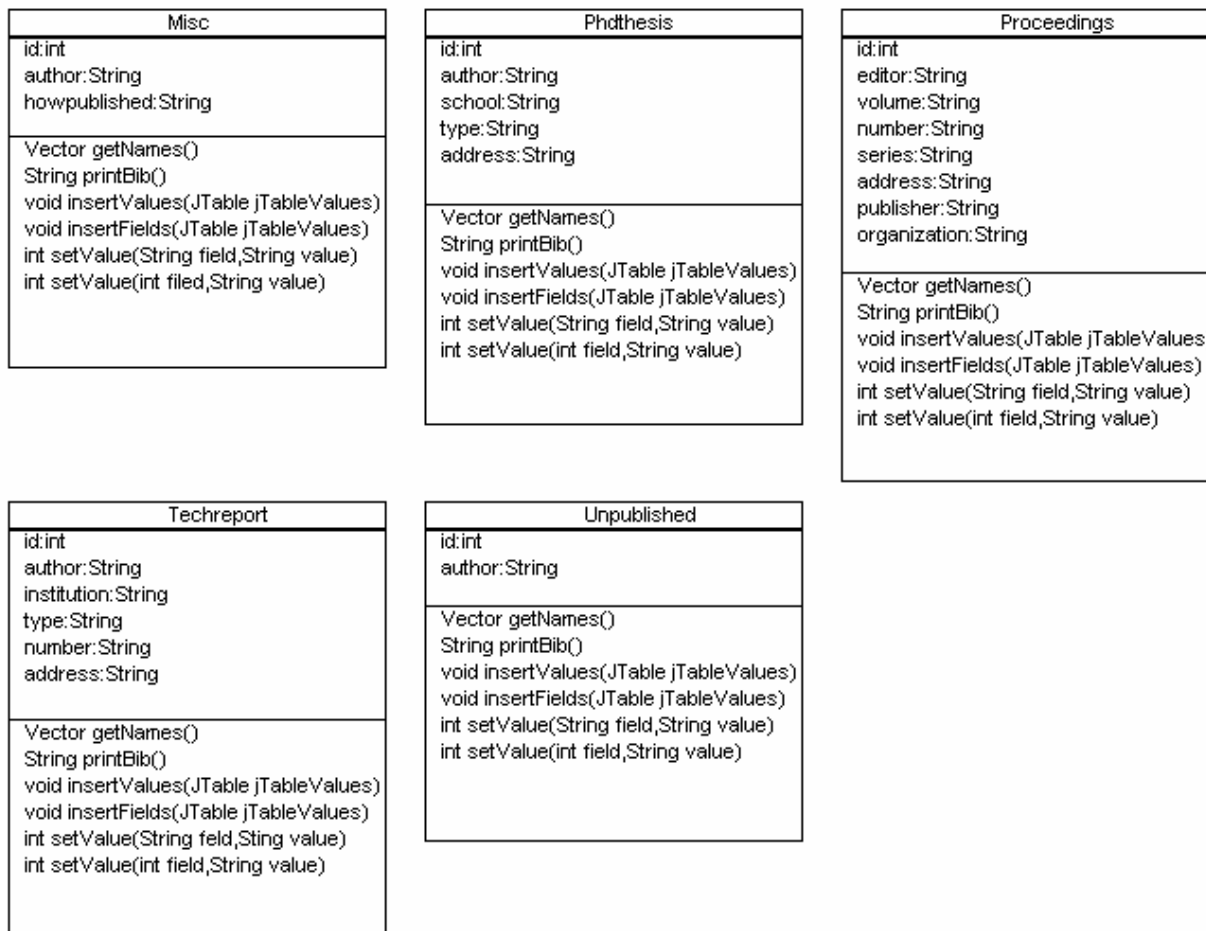
Inbook
id:int author:String editor:String chapter:String pages:String publisher:String volume:String number:String series:String type:String address:String edition:String
Vector getNames() String printBib() void insertValues(JTable jTableValues) void insertFields(JTable jTableValues) int setValue(String field,String value) int setValue(int field,String value)

Incollection
id:int author:String booktitle:String publisher:String editor:String volume:String number:String series:String pages:String address:String edition:String
Vector getNames() String printBib() void insertValues(JTable jTableValues) void insertFields(JTable jTableValues) int setValue(String field,String value) int setValue(int field,String value)

Inproceedings
id:int author:String booktitle:String editor:String volume:String number:String series:String pages:String address:String edition:String organization:String publisher:String
Vector getNames() String printBib() void insertValues(JTable jTableValues) void insertFields(JTable jTableValues) int setValue(String field,String value) int setValue(int field,String value)

Manual
id:int author:String organization:String address:String edition:String
Vector getNames() String printBib() void insertValues(JTable jTableValues) void insertFields(JTable jTableValues) int setValue(String field,String value) int setValue(int field,String value)

Mastersthesis
id:int author:String school:String type:String address:String
Vector getNames() String printBib() void insertValues(JTable jTableValues) void insertFields(JTable jTableValues) int setValue(String field,String value) int setValue(int field,String value)



Επεξήγηση μεθόδων

Όνομα Μεθόδου	Περιγραφή
String getΌνομα πεδίου()	Επιστρέφει την τιμή του συγκεκριμένου πεδίου
void setΌνομα πεδίου(String value)	Θέτει ως τιμή του συγκεκριμένου πεδίου την τιμή που δίνεται ως όρισμα
int setBaseValue(String field,String value)	Παίρνει ως όρισμα το όνομα ενός πεδίου και την τιμή που θα ανατεθεί στο συγκεκριμένο πεδίο. Έτσι ανάλογα με το πεδίο που δίνεται γίνεται και η ανάθεση στο συγκεκριμένο πεδίο. Η συγκεκριμένη συνάρτηση αφορά μόνο τα βασικά πεδία της κύριας κλάσης Record. Αν το field αναφέρεται στα βασικά πεδία της Record τότε

	επιστρέφει 0 σε διαφορετική περίπτωση επιστρέφει -1.
<code>int setBaseValue(int field,String value)</code>	Γίνεται το ίδιο ακριβώς με την προηγούμενη συνάρτηση μόνο που δεν περνά σαν όρισμα το όνομα του πεδίου αλλά ένας αριθμός που αντιστοιχεί σε κάποιο πεδίο.
<code>int setValue(String field,String value)</code>	Καλεί την <code>setBaseValue</code> της <code>Record</code> αν αυτή επιστρέψει -1 τότε κάνει ανάλογα με το πεδίο που δίνεται ως όρισμα την ανάθεση της <code>value</code> στο συγκεκριμένο πεδίο. Σε διαφορετική περίπτωση δεν κάνει τίποτα (επιστρέφει 0)
<code>int setValue(int field,String value)</code>	Γίνεται το ίδιο ακριβώς με την προηγούμενη συνάρτηση μόνο που δεν περνά σαν όρισμα το όνομα του πεδίου αλλά ένας αριθμός που αντιστοιχεί σε κάποιο πεδίο.
<code>Vector getNames()</code>	Επιστρέφει έναν <code>Vector</code> με όλα τα ονόματα πεδίων που αντιστοιχούν στον συγκεκριμένο τύπο εγγραφής.
<code>String printBib()</code>	Επιστρέφει ένα αλφαριθμητικό που είναι η αναπαράσταση του συγκεκριμένου τύπου εγγραφής σε μορφή <code>BibTeX</code> .
<code>void insertValues(JTable jTableValues)</code>	Εισάγει στον πίνακα που τίθεται σαν όρισμα τις τιμές των πεδίων της συγκεκριμένης εγγραφής.

Κλάσεις ελέγχου

Οι κλάσεις ελέγχου που χρησιμοποιήθηκαν στο πρόγραμμα είναι οι εξής:

BibPipeAdvertisement

BibPipeAdvertisement
PipeAdvertisement pipeAdv
PipeAdvertisement getAdvertisement()

Όνομα Μεθόδου	Περιγραφή
PipeAdvertisement getAdvertisement()	Συνάρτηση που επιστρέφει την διαφήμιση διασωλήνωσης που δημιουργήθηκε.

BibtexParser

BibtexParser
Vector Records
Boolean parseRecord(String strBibText)
String getRecord(int nIndex)
Vector getRecords()

Όνομα Μεθόδου	Περιγραφή
Boolean parseRecord(String strBibText)	Συνάρτηση η οποία παίρνει ένα αλφαριθμητικό που είναι στην ουσία μια ή περισσότερες εγγραφές σε BibTeX το οποίο αναλύεται για να εξαχθούν μια μια όλες τις εγγραφές και να εισαχθούν σε έναν Vector. Αν η διαδικασία είναι επιτυχής επιστρέφει true.
String getRecord(int nIndex)	Επιστρέφει την εγγραφή που έχει θέση μέσα στον Vector ίση με τον αριθμό που δίνεται σαν όρισμα.

 Vector getRecords()

 Επιστρέφει τον Vector με όλες τις εγγραφές που προέκυψαν κατά την ανάλυση.

ClientConnectionHandler

ClientConnectionHandler
JxtaBiDiPipe bipipe
String myName
String query
Results searchRecords
CountSt countofPeers
String serverName
String SenderMessage
PeerGroup netPeerGroup
ModuleSpecAdvertisement mdsadv
void run()

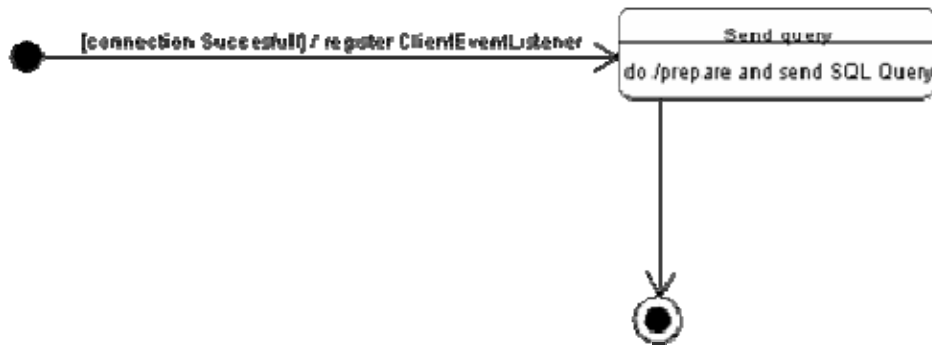
Όνομα Μεθόδου

void run()

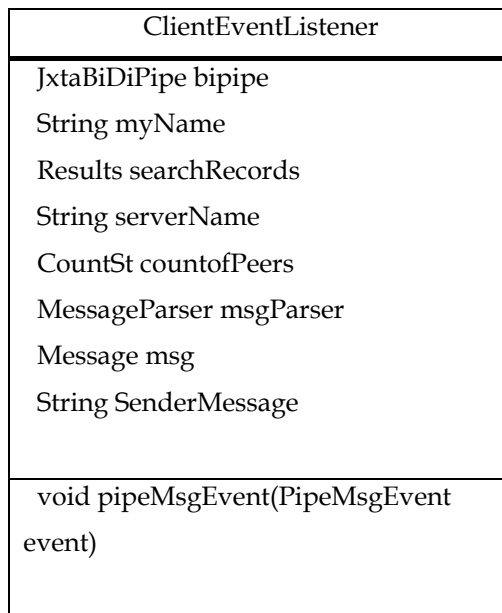
Περιγραφή

Επειδή η συγκεκριμένη κλάση είναι ένα νήμα για τον λόγο αυτό υπάρχει και η συγκεκριμένη συνάρτηση. Συγκεκριμένα παίρνει την διαφήμιση διασωλήνωσης από την διαφήμιση προδιαγραφής μονάδας(ModuleSpecAdvertisement) και δημιουργεί μια διασωλήνωση εξαγωγής σημείου τέλους (output pipe endpoint) για να συνδεθεί και να στείλει την ερώτηση .

Επειδή η συγκεκριμένη κλάση έχει μια συγκεκριμένη διάρκεια ζωής στο σύστημα χρησιμοποιούμε το **διάγραμμα καταστάσεων** για να περιγράψουμε την συμπεριφορά της.

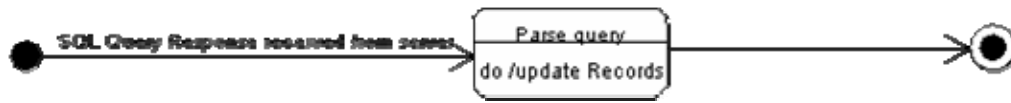


ClientEventListener

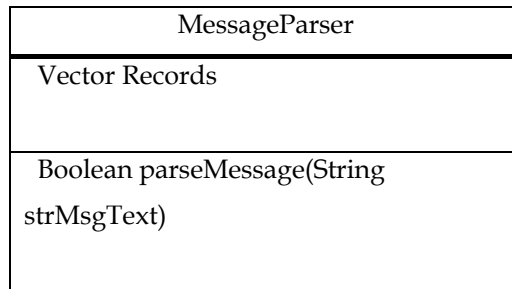


Όνομα Μεθόδου	Περιγραφή
void pipeMsgEvent(PipeMsgEvent event)	Η συγκεκριμένη κλάση είναι μια κλάση που υλοποιεί το interface PipeMsgListener και για τον λόγο αυτό υπάρχει η συνάρτηση αυτή έτσι ώστε να διαχειρίζεται τα λαμβανόμενα PipeMsgEvent events .Συγκεκριμένα λαμβάνει τις απαντήσεις της ερώτησης που έστειλε το ομότιμο μέλος στο ομότιμο δίκτυο και τις προσθέτει στο κατάλληλο Vector.

Επειδή η συγκεκριμένη κλάση έχει μια συγκεκριμένη διάρκεια ζωής στο σύστημα χρησιμοποιούμε το **διάγραμμα καταστάσεων** για να περιγράψουμε την συμπεριφορά της.

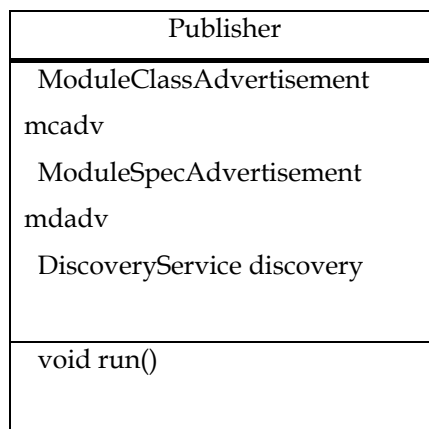


MessageParser



Όνομα Μεθόδου	Περιγραφή
Boolean parseMessage(String strMsgText)	Συνάρτηση η οποία παίρνει ένα αλφαριθμητικό ,πρόκειται για ένα μήνυμα απάντησης και είναι στην ουσία μια ή περισσότερες εγγραφές το οποίο αναλύεται για να εξαχθούν μια μια όλες τις εγγραφές και να εισαχθούν σε έναν Vector.Αν η διαδικασία είναι επιτυχής επιστρέφει true.
Vector getRecords()	Επιστρέφει τον Vector με όλες τις εγγραφές που προέκυψαν κατά την ανάλυση.

Publisher



Όνομα Μεθόδου	Περιγραφή
void run()	Επειδή η συγκεκριμένη κλάση είναι ένα νήμα για τον λόγο αυτό υπάρχει και η συγκεκριμένη συνάρτηση. Συγκεκριμένα δημοσιεύει την διαφήμιση της κλάσης της μονάδας (Module class advertisement) και την διαφήμιση της προδιαγραφής της μονάδας (Module Spec advertisement) συνεχώς σε τακτά χρονικά διαστήματα μέσα από έναν ατέρμονα βρόγχο.

ServerEventListener

ServerEventListener
JxtaBiDiPipe bipipe
String myName
Vector searchRecords
Message msg
String SenderMessage
void pipeMsgEvent(PipeMsgEvent event)

Όνομα Μεθόδου	Περιγραφή
void pipeMsgEvent(PipeMsgEvent event)	Η συγκεκριμένη κλάση είναι μια κλάση που υλοποιεί το interface PipeMsgListener και για τον λόγο αυτό υπάρχει η συνάρτηση αυτή έτσι ώστε να διαχειρίζεται τα λαμβανόμενα PipeMsgEvent events .Συγκεκριμένα λαμβάνει τις ερωτήσεις τις εκτελεί και στέλνει τις απαντήσεις στο ενδιαφερόμενο ομότιμο μέλος.

Επειδή η συγκεκριμένη κλάση έχει μια συγκεκριμένη διάρκεια ζωής στο σύστημα χρησιμοποιούμε το **διάγραμμα καταστάσεων** για να περιγράψουμε την συμπεριφορά της.



Viewer

Viewer
Results records JTable tableResults JLabel labelMsg int flag
void run() void finish()

Όνομα Μεθόδου	Περιγραφή
void run()	Επειδή η συγκεκριμένη κλάση είναι ένα νήμα για τον λόγο αυτό υπάρχει και η συγκεκριμένη συνάρτηση. Συγκεκριμένα επανεμφανίζει τα στοιχεία από ένα Vector που είναι στην ουσία οι απαντήσεις από μια ερώτηση στο ομότιμο δίκτυο σε τακτά χρονικά διαστήματα.
void finish()	Συνάρτηση που καλείται για να δηλώσει τον τερματισμό του συγκεκριμένου νήματος.

client

client
PeerGroup netPeerGroup DiscoveryService discovery Message msg String query CountSt countofPeers Results searchRecords int flag String myName String serviceName
void run() void startJxta() void clearLocalCache() Enumeration lookForAdvertisements() void startClient() void finish()

Όνομα Μεθόδου	Περιγραφή
void run()	Επειδή η συγκεκριμένη κλάση είναι ένα νήμα για τον λόγο αυτό υπάρχει και η συγκεκριμένη συνάρτηση. Συγκεκριμένα καλεί την startJxta() και την startClient() συνάρτηση.
void startJxta()	Λαμβάνει την υπηρεσία ανακάλυψης.
void clearLocalCache()	Σβήνει τις όλες έως τώρα ανακαλυμμένες διαφημίσεις
Enumeration lookForAdvertisements()	Ανακαλύπτει όλες τις διαφημίσεις τόσο τις τοπικές όσο και τις μακρινές και επιστρέφει ένα enumeration με όλες αυτές τις διαφημίσεις.
void startClient()	Η συνάρτηση αυτή αναζητά τις διαφημίσεις προδιαγραφής μονάδας που συσχετίζονται με κάποια συγκεκριμένη υπηρεσία και εξάγει τις πληροφορίες διασωλήνωσης για να

συνδεθεί με αυτήν την υπηρεσία. Για κάθε υπηρεσία δημιουργεί ένα ClientConnectionHandler (thread) για να την διαχειριστεί.(δηλαδή την σύνδεση με αυτό το ομότιμο μέλος).

void finish()

Συνάρτηση που καλείται για να δηλώσει τον τερματισμό του συγκεκριμένου νήματος.

server

server
PeerGroup group JxtaServerPipe serverPipe DiscoveryService discovery String myName String serviceName BibPipeAdvertisement bibpipeAdv
void run() void startJxta() void startServer()

Όνομα Μεθόδου

Περιγραφή

void run()

Επειδή η συγκεκριμένη κλάση είναι ένα νήμα για τον λόγο αυτό υπάρχει και η συγκεκριμένη συνάρτηση. Συγκεκριμένα καλεί την **startJxta()** και την **startServer()** συνάρτηση.

void startJxta()

Λαμβάνει την υπηρεσία ανακάλυψης.

void startServer()

Πρώτα η κλάση server δημιουργεί ,ρυθμίζει και δημοσιεύει μια διαφήμιση κλάσης μονάδας(Module class advertisement) και μια διαφήμιση προδιαγραφής μονάδας (Module Spec advertisement) που περιέχει την διαφήμιση διασωλήνωσης(PipeAdvertisement).Έπειτα ξεκινά το

νήμα Publisher για να δημοσιεύει αυτές τις διαφημίσεις συνεχώς σε τακτά χρονικά διαστήματα . Όταν μια αίτηση για σύνδεση ληφθεί ο server ξεκινά ένα νήμα `ServerEventListener` για να την διαχειριστεί.

Κλάσεις ορίου-διεπαφής

Γενικά υπάρχουν τρεις βασικές κλάσεις ορίου .

Firstform

Η Firstform είναι η main class της εφαρμογής.

Firstform
JButton jButtonLogin
JLabel jLabelMessage
JLabel jLabelPicture
JLabel jLabelUsername
JTextField jTextFieldUsername
JPanel jPanelBasic
JPanel jPanelForUsername
JPanel jPanelPicture
void main(String args[])
void jButtonLoginPerformed(java.awt.event.ActionEvent evt)

Όνομα Μεθόδου	Περιγραφή
void main(String args[])	Είναι το σημείο από το οποίο ξεκινά η εκτέλεση της εφαρμογής. Σε αυτήν την συνάρτηση δημιουργείται και ξεκινά η εξ ορισμού jxta NetPeerGroup.
void jButtonLoginPerformed (java.awt.event.ActionEvent evt)	Συνάρτηση που καλείται όταν ο χρήστης πατήσει το κουμπί jButtonLogin για να επιτύχει πρόσβαση στο σύστημα. Αν είναι κενό εμφανίζει μήνυμα διαφορετικά πραγματοποιεί πρόσβαση στο σύστημα.

Mainform

Mainform	
ButtonGroup buttonGroupTarget;	JPanel jPanelDetails
JButton jButtonCancel	JPanel jPanelGenSearch
JButton jButtonDelete;	JPanel jPanelMessage
JButton jButtonEdit;	JPanel jPanelMyCollection
JButton jButtonNew;	JPanel jPanelResults
JButton jButtonSave;	JPanel jPanelSearch
JButton jButtonSearch;	JPanel jPanelTarget
JButton jButtonStop;	JProgressBar jProgressBar1
JComboBox jCBoxCategory	JRadioButton jrButtonLocal
JDialog jDialogAbout	JRadioButton jrButtonRemote
JLabel jLabelAbout	JScrollPane jScrollPaneMyColection
JLabel jLabelAuthor	JScrollPane jScrollPaneResults
JLabel jLabelBibTeXView	JScrollPane
JLabel jLabelBibTextResults	jScrollPaneViewCollection
JLabel jLabelCategory	JScrollPane jScrollPaneViewResults
JLabel jLabelFooter	JTextArea jTAreaAbout
JLabel jLabelKey	JTextArea jTAreaViewCollection
JLabel jLabelLogout	JTextArea jTAreaViewResults
JLabel jLabelMember	JTextField jTextFieldAuthor
JLabel jLabelMsgViewer	JTextField jTextFieldGeneral
JLabel jLabelPicture	JTextField jTextFieldKey
JLabel jLabelPictureAbout	JTextField jTextFieldTitle
JLabel jLabelTitle	JTextField jTextFieldYear
JLabel jLabelYear	JTabbedPane jTabbedPaneMenu
JPanel jPanelAbout	JTable jTableMyCollection
JPanel jPanelBasic	JTable jTableResults
	JTextArea jTextAreaMessage
void AboutClicked(java.awt.event.MouseEvent evt) void CancelSave(java.awt.event.MouseEvent evt) void Logout(java.awt.event.MouseEvent evt) void Close(java.awt.event.WindowEvent evt) void EditRecord(java.awt.event.ActionEvent evt) void SaveFromSearch(java.awt.event.ActionEvent evt) void stopSearch(java.awt.event.ActionEvent evt) void jTableResultsMouseClicked(java.awt.event.MouseEvent evt)	

```

void Search(java.awt.event.ActionEvent evt)
void NewRecord(java.awt.event.MouseEvent evt)
void deleteRecord(java.awt.event.MouseEvent evt)
void jTableMyCollectionMouseClicked(java.awt.event.MouseEvent evt)
void myCollection(java.awt.event.ComponentEvent evt)

```

Όνομα Μεθόδου	Περιγραφή
void AboutClicked (java.awt.event.MouseEvent evt)	Συνάρτηση η οποία εμφανίζει ένα ειδικό παράθυρο Dialog για πληροφορίες σχετικά με την εφαρμογή.
void CancelSave (java.awt.event.MouseEvent evt)	Μέθοδος για να ακυρωθεί η προσπάθεια εισαγωγής νέα /ων εγγραφής /ών.
void Logout(java.awt.event.MouseEvent evt)	Βγάζει το ομότιμο μέλος από το σύστημα.
void Close(java.awt.event.WindowEvent evt)	Συνάρτηση που καλείται όταν κλείνει το παράθυρο για να τερματίσει την jxta NetPeerGroup.
void EditRecord (java.awt.event.ActionEvent evt)	Καλείται για να εμφανίσει μια ειδική φόρμα για την επεξεργασία μιας συγκεκριμένης εγγραφής.
void SaveFromSearch (java.awt.event.ActionEvent evt)	Μέθοδος για την αποθήκευση μιας εγγραφής που είναι αποτέλεσμα μιας αναζήτησης στο ομότιμο δίκτυο.
void stopSearch (java.awt.event.ActionEvent evt)	Συνάρτηση για τον τερματισμό της αναζήτησης στο ομότιμο δίκτυο.
void jTableResultsMouseClicked (java.awt.event.MouseEvent evt)	Όταν ο χρήστης επιλέξει μια εγγραφή στον πίνακα με τα αποτελέσματα εμφανίζει την μορφή BibTeX της συγκεκριμένης εγγραφής στην ειδική περιοχή κειμένου.

<code>void Search(java.awt.event.ActionEvent evt)</code>	Υλοποιεί την αναζήτηση ανάλογα με τον στόχο που έχει επιλεγθεί τοπική συλλογή ή ομότιμο δίκτυο.
<code>void NewRecord (java.awt.event.MouseEvent evt)</code>	Συνάρτηση που παίρνει το κείμενο από την ειδική περιοχή κειμένου το αναλύει δημιουργεί τις νέες εγγραφές και τις αποθηκεύει στην τοπική συλλογή.
<code>void deleteRecord (java.awt.event.MouseEvent evt)</code>	Καλείται για να διαγράψει την επιλεγμένη εγγραφή από την τοπική συλλογή
<code>void jTableMyCollectionMouseClicked (java.awt.event.MouseEvent evt)</code>	Όταν ο χρήστης επιλέξει μια εγγραφή στον πίνακα με τις εγγραφές της τοπικής συλλογής εμφανίζει την μορφή BibTeX της συγκεκριμένης εγγραφής στην ειδική περιοχή κειμένου
<code>void myCollection (java.awt.event.ComponentEvent evt)</code>	Μέθοδος που εκτελείται κάθε φορά που εμφανίζεται η σελίδα tab με τις εγγραφές του εκάστοτε ομότιμου μέλους έτσι ώστε αυτή να γεμίσει με τις εγγραφές της τοπικής συλλογής.

Editform

Editform
JButton jButtonCancel
JButton jButtonSave
JCheckBox jCheckPublished
JLabel jLabelHeader
JLabel jLabelPublished
JLabel jLabelType
JPanel jPanelInner
JPanel jPanel1
JPanel jPanelHeader
JPanel jPanelValues
JScrollPane jScrollPaneValues
JTable jTableValues
void SaveChanges(java.awt.event.ActionEvent evt)
void CancelEdit(java.awt.event.ActionEvent evt)

Όνομα Μεθόδου	Περιγραφή
void SaveChanges (java.awt.event.ActionEvent evt)	Μέθοδος που καλείται για να αποθηκεύσει τις αλλαγές που έχουν γίνει στις τιμές της υπο επεξεργασία εγγραφής.
void CancelEdit (java.awt.event.ActionEvent evt)	Καλείται για να ακυρώσει την επεξεργασία της συγκεκριμένης εγγραφής.

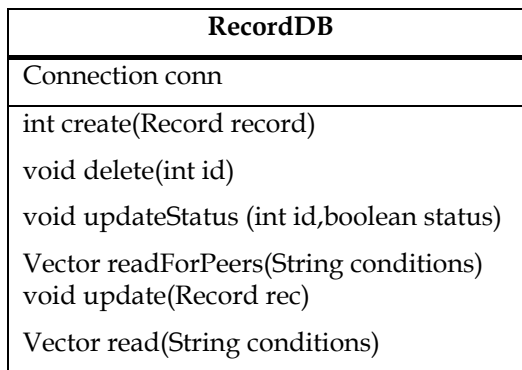
Σχεδίαση

Κατά τη διάρκεια της σχεδίασης μελετήθηκαν περισσότερο κάποιες κλάσεις και αποφασίστηκε ότι θα ήταν χρήσιμη η σχεδίαση μερικών ακόμα κλάσεων

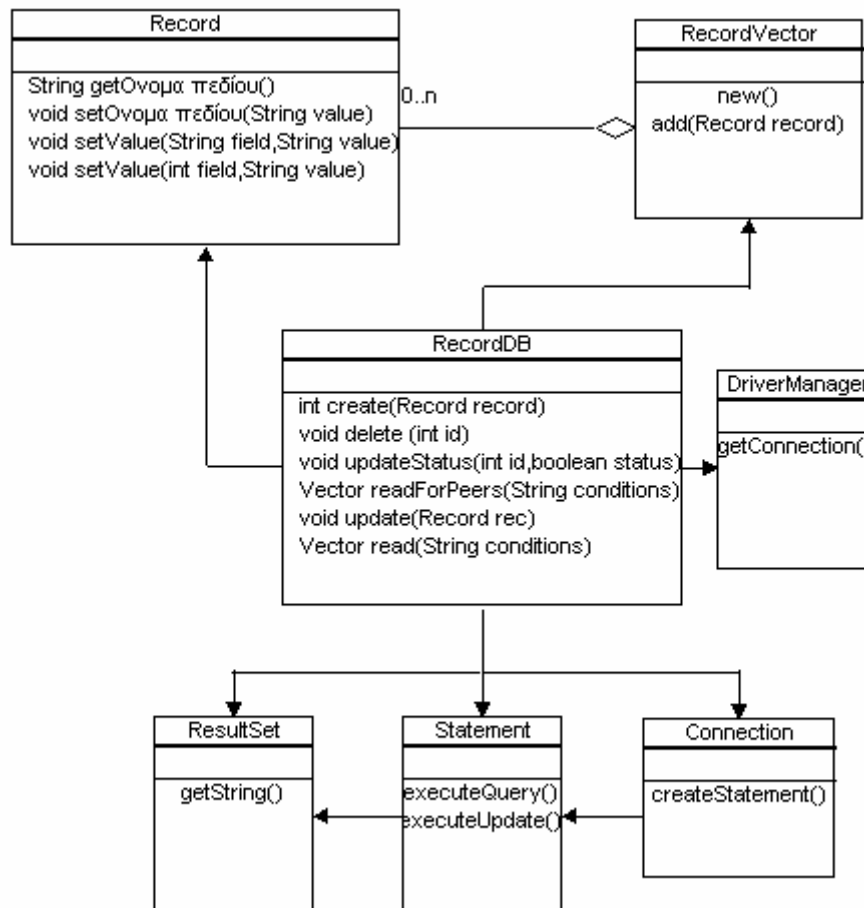
Κλάση για την επικοινωνία με την βάση

Επειδή το σύστημα που σχεδιάστηκε ήταν αντικειμενοστραφές ενώ η βάση που χρησιμοποιήθηκε ήταν σχεσιακή ήταν απαραίτητη η σχεδίαση μιας νέας κλάσης η οποία θα αναλάβει την μετατροπή των πινάκων σε αντικείμενα και το αντίστροφο. Αυτή είναι η RecordDB.

Η RecordDB συνδέεται με τον JDBC μέσω ODBC driver.



Ακολουθεί ένα διάγραμμα συνεργασίας που περιγράφει πως ακριβώς λειτουργεί η RecordDB.



Αποφασίστηκε ότι στην πρώτη έκδοση του προγράμματος η βάση δεδομένων θα είναι απλή με μόνο ένα πίνακα ο οποίος φαίνεται παρακάτω.

Record	
🔑 recordid:Αυτόματη Αρίθμηση key:κείμενο[50] author: κείμενο [150] title: κείμενο [150] journal: κείμενο [50] volume: κείμενο [50] number: κείμενο [50] pages: κείμενο [50] month: κείμενο [50] note: κείμενο [50] publisher: κείμενο [50] series: κείμενο [50] address: κείμενο [50]	howpublished: κείμενο [50] edition: κείμενο [50] booktitle: κείμενο [150] chapter: κείμενο [50] crossref: κείμενο [50] institution: κείμενο [50] editor: κείμενο [50] organization: κείμενο [50] school: κείμενο [50] type: κείμενο [50] category: κείμενο [50] status:Ναι/Όχι

Επεξήγηση συναρτήσεων/ μεθόδων

Όνομα Μεθόδου	Περιγραφή
int create(Record record)	Προσθέτει μια νέα εγγραφή στην τοπική συλλογή και επιστρέφει τον κωδικό της εγγραφής που μόλις προστέθηκε.
void delete(int id)	Διαγράφει μια εγγραφή με κωδικό id.
void updateStatus (int id,boolean status)	Αλλάζει την κατάσταση μιας εγγραφής για το αν θα είναι δημόσια ή όχι.
Vector readForPeers(String conditions)	Εκτελεί ένα ερώτημα στις εγγραφές που είναι δημόσιες με τις συνθήκες που τίθενται ως όρισμα και επιστρέφει το Vector με τις εγγραφές -αποτέλεσμα αναζήτησης
void update(Record rec)	Ενημερώνει την συγκεκριμένη εγγραφή.
Vector read(String conditions)	Εκτελεί ένα ερώτημα σε όλες τις εγγραφές με τις συνθήκες που τίθενται ως όρισμα και επιστρέφει το Vector με τις εγγραφές -αποτέλεσμα αναζήτησης.

Άλλες βοηθητικές κλάσεις

Τέλος δημιουργήσαμε και άλλες δύο κλάσεις για πρακτικούς λόγους όπως για το λόγο του να περνάμε τιμές μεταξύ διαφορετικών νημάτων και να μην έχουμε πρόβλημα με την παραλληλία μεταξύ αυτών. Έτσι έχουμε:

Result

Result
String peerName
Record record
String getPeerName() void setPeerName(String peerName) Record getRecord() void setRecord(Record record)

Η συγκεκριμένη κλάση στην ουσία είναι μια δομή.

Όνομα Μεθόδου	Περιγραφή
String getPeerName()	Επιστρέφει το όνομα του ομότιμου μέλους του συγκεκριμένου Result
void setPeerName(String peerName)	Δίνει την συγκεκριμένη τιμή στο peerName
Record getRecord()	Επιστρέφει την εγγραφή του συγκεκριμένου Result
void setRecord(Record record)	Θέτει ως Record το Record που τίθεται ως όρισμα.

Results

Results
Vector<Result> result
synchronized Vector getResults()
synchronized void addElement(Result resultItem)
synchronized Iterator iterator()
synchronized int size()
synchronized void removeAllElements()
synchronized Result elementAt(int nIndex)

Πρόκειται για μια κλάση που διαχειρίζεται έναν Vector με Result στοιχεία. Οι μέθοδοι της έχουν την χαρακτηριστική λέξη `synchronized` στον ορισμό τους για να αποφευχθούν οποιαδήποτε προβλήματα παραλληλίας όταν ένα αντικείμενο αυτής περνά σαν όρισμα από το ένα νήμα στο άλλο.

Όνομα Μεθόδου	Περιγραφή
synchronized Vector getResults()	Επιστρέφει τον Vector με τα στοιχεία.
synchronized void addElement(Result resultItem)	Προσθέτει ένα νέο στοιχείο στον Vector
synchronized Iterator iterator()	Επιστρέφει έναν Iterator για τον Vector
synchronized int size()	Επιστρέφει τον μέγεθος του Vector
synchronized void removeAllElements()	Διαγράφει όλα τα στοιχεία του Vector
synchronized Result elementAt(int nIndex)	Επιστρέφει το στοιχείο Result που έχει θέση στον Vector ίση με αυτή που δίνεται ως όρισμα.

CountSt

CountSt
int count
synchronized int getCount() synchronized void setCount(int num)

Όνομα Μεθόδου	Περιγραφή
synchronized int getCount()	Επιστρέφει την τιμή του count
synchronized void setCount(int num)	Θέτει την τιμή του count ίση με αυτή που τίθεται ως όρισμα

RecordFactory

RecordFactory
Record getInstance(String category)

Κλάση που χρησιμοποιείται για την διαχείριση της σχέσης κληρονομικότητας που υπάρχει μεταξύ της βασικής κλάσης Record και των υποκλάσεων αυτής.

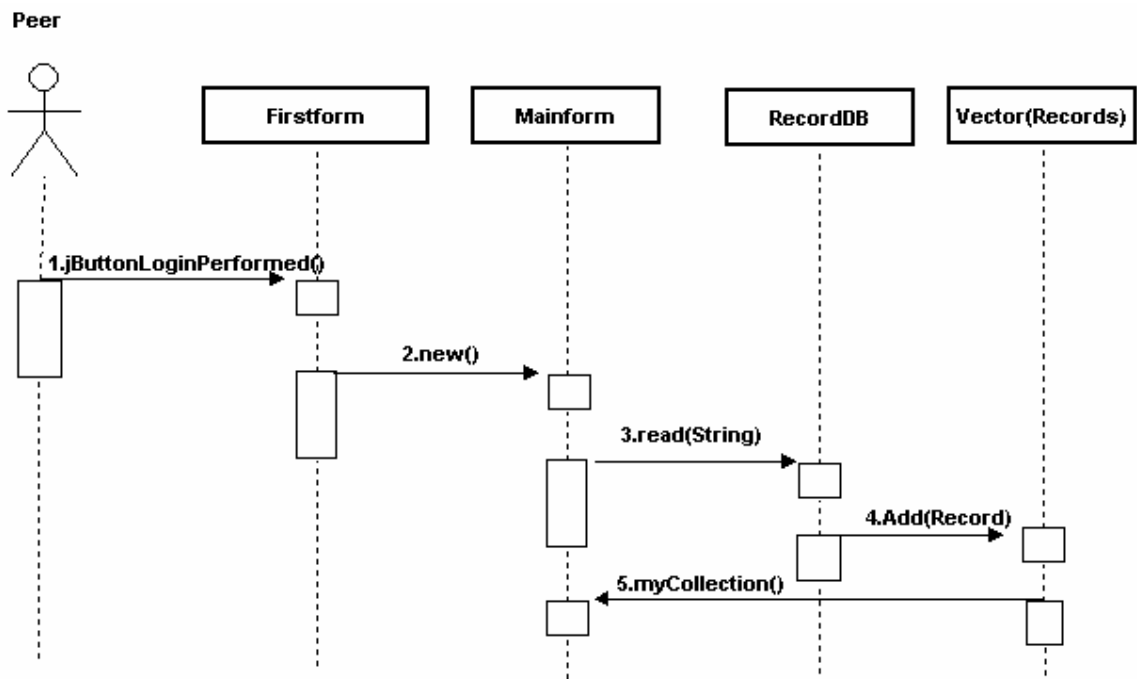
Όνομα Μεθόδου	Περιγραφή
Record getInstance(String category)	Επιστρέφει ένα νέο αντικείμενο Record κατηγορίας ίσης με αυτή που δίνεται σαν όρισμα

Περιγραφή σχεδίασης περιπτώσεων χρήσης

1.Πραγματοποίησε πρόσβαση στο σύστημα

- Αυτήν την περίπτωση χρήσης την χρησιμοποιεί οποιαδήποτε ομότιμο μέλος που επιθυμεί να συνδεθεί με το σύστημα.
- Αν δεν έχει ποτέ προσπαθήσει να συνδεθεί στο σύστημα εμφανίζεται η φόρμα διαμόρφωσης του JXTA όπου το ομότιμο μέλος συμπληρώνει στοιχεία σχετικά με τον ρόλο του στο ομότιμο δίκτυο κ.α
- Αν έχει διαμορφώσει το JXTA τότε εμφανίζεται η Firstform και ο χρήστης εισάγει το username στο JTextFieldUsername με το οποίο θέλει να χαρακτηρίζεται στο ομότιμο δίκτυο.
- Πατάει το jButtonLogin και καλείται η jButtonLoginPerformed μέθοδος.
- Αν είναι κενό το username τότε εμφανίζεται κάποιο μήνυμα
- Αν όχι τότε εμφανίζεται η Mainform και καλείται η myCollection μέθοδος που εμφανίζει όλες τις εγγραφές του ομότιμου μέλους στο jTableMyCollection.

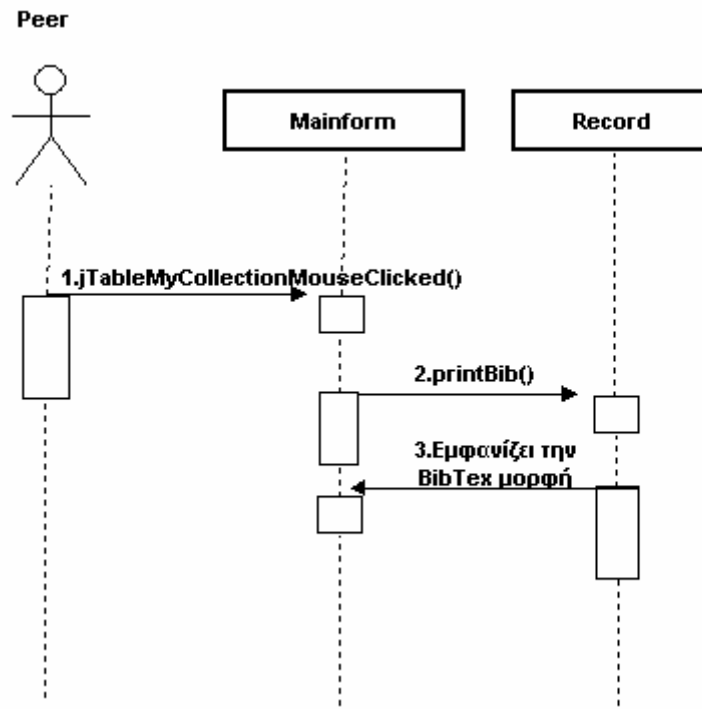
Διάγραμμα Ακολουθίας



2.Πρόβαλλε την εγγραφή σε BibTeX

- Περίπτωση χρήσης που χρησιμοποιείται από οποιαδήποτε συνδεδεμένο ομότιμο μέλος
- Όταν το τελευταίο επιλέξει την εγγραφή από το jTableMyCollection που θέλει να δει σε BibTeX μορφή καλείται η jTableMyCollectionMouseClicked μέθοδος .
- Η μέθοδος αυτή παίρνει την επιλεγμένη εγγραφή και ανάλογα με τον τύπο καταχώρησης που ανήκει καλείται και η αντίστοιχη μέθοδος printBib που επιστρέφει το κείμενο με την μορφή BibTeX της εγγραφής με την κατάλληλη δομή .
- Τέλος εμφανίζεται /προστίθεται το κείμενο αυτό στην περιοχή κειμένου jTAreaViewCollection .

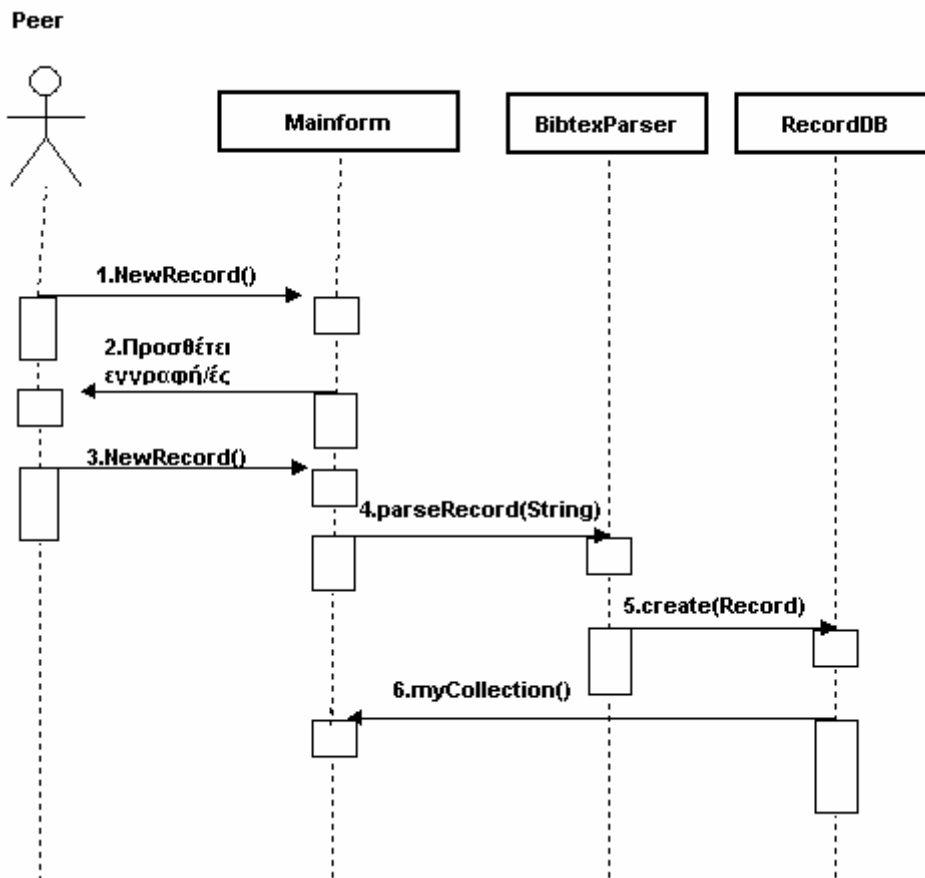
Διάγραμμα Ακολουθίας



3.Εισήγαγε την /τις νέα /ες εγγραφή /ες

- Περίπτωση χρήσης που χρησιμοποιείται από οποιαδήποτε συνδεδεμένο ομότιμο μέλος
- Ο χρήστης πατά το κουμπί jButtonNew και καλείται η μέθοδος NewRecord.
- Η μέθοδος αυτή καθαρίζει το jTAreaViewCollection και αλλάζει την ετικέτα του jButtonNew από New σε Save.
- Ο χρήστης προσθέτει τις εγγραφές (στην ουσία ένα κείμενο με εγγραφές σε BibTeX και πατά ξανά το jButtonNew και καλείται η μέθοδος NewRecord ξανά .
- Αυτήν την φορά η μέθοδος λαμβάνει το κείμενο το αναλύει , συνθέτει τις νέες εγγραφές με την κλήση της μεθόδου parseRecord της κλάσης BibtexParser και τις αποθηκεύει στην τοπική μνήμη με την μέθοδο create της κλάσης RecordDB.
- Τέλος επανεμφανίζεται το jTableMyCollection(το jPanelMyCollection στην ουσία, κλήση της myCollection()) με τις όλες τις εγγραφές νέες και παλιές.

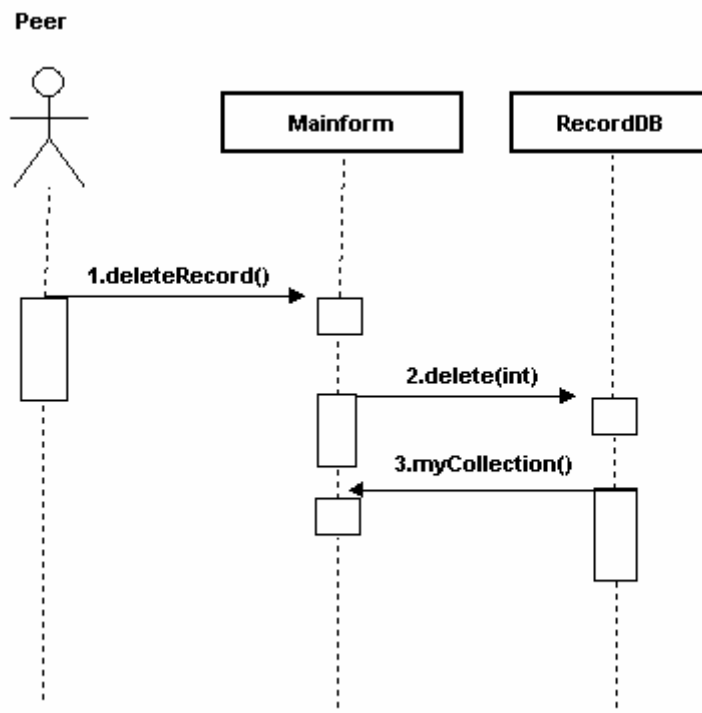
Διάγραμμα Ακολουθίας



4.Διέγραψε την εγγραφή

- Περίπτωση χρήσης που χρησιμοποιείται από οποιαδήποτε συνδεδεμένο ομότιμο μέλος
- Το τελευταίο επιλέγει την εγγραφή από το jTableMyCollection που θέλει να διαγράψει πατάει το jButtonDelete και καλείται η συνάρτηση deleteRecord.
- Η τελευταία λαμβάνει την επιλεγμένη εγγραφή από το jTableMyCollection και καλεί την μέθοδο delete της κλάσης RecordDB για να την διαγράψει.
- Τέλος επανεμφανίζεται το jTableMyCollection (το jPanelMyCollection στην ουσία, κλήση της myCollection()) έτσι ώστε να είναι συνεπής με τις τελευταίες αλλαγές που έγιναν.

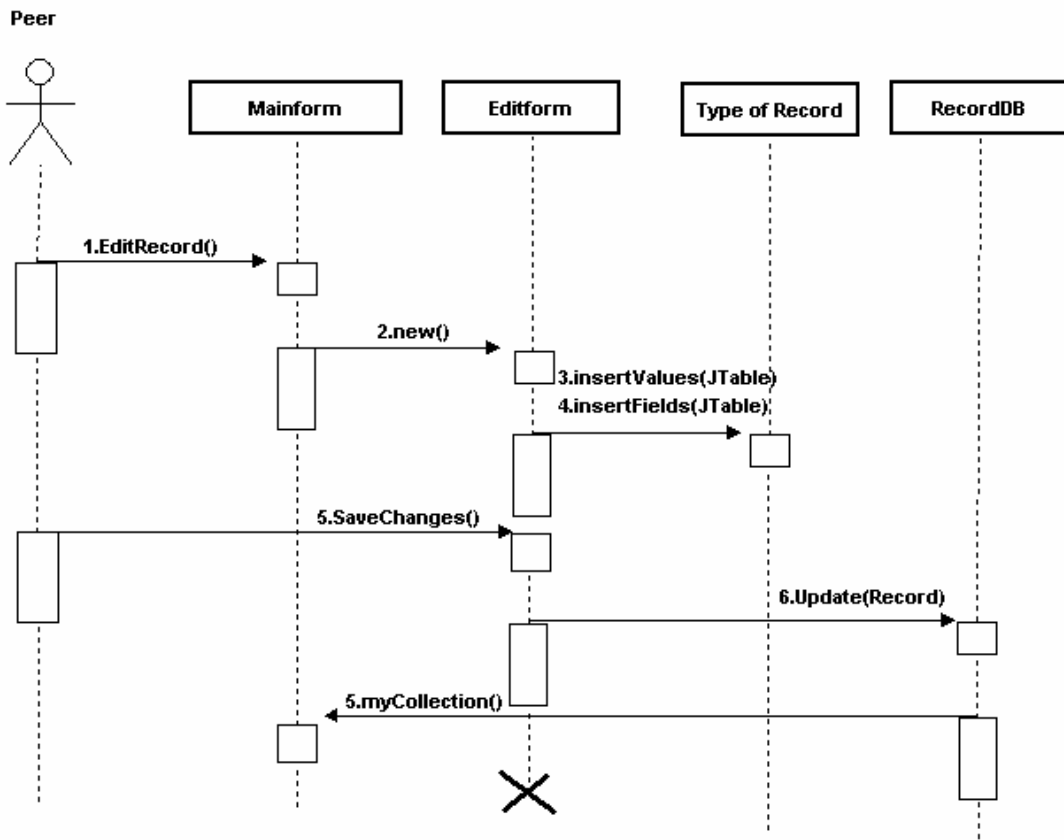
Διάγραμμα Ακολουθίας



5.Επεξεργάσου την εγγραφή

- Περίπτωση χρήσης που χρησιμοποιείται από οποιαδήποτε συνδεδεμένο ομότιμο μέλος
- Ο χρήστης /ομότιμο μέλος επιλέγει την εγγραφή από το jTableMyCollection που θέλει να επεξεργαστεί , πατάει το jButtonEdit και καλείται η συνάρτηση EditRecord
- Η συνάρτηση αυτή παίρνει την επιλεγμένη εγγραφή , δημιουργεί και εμφανίζει μια νέα Editform.Στον constructor αυτής καλείται η insertValues για να εισάγει στο jTableValues τις τιμές τις εγγραφής και η insertFields για να εισάγει στο jTableValues τα κατάλληλα πεδία ανάλογα με τον τύπο της καταχώρησης στον οποίο ανήκει η συγκεκριμένη εγγραφή.
- Το ομότιμο μέλος κάνει τις αλλαγές ,πατάει το jButtonSave και καλείται η μέθοδος SaveChanges του Editform.
- Η μέθοδος αυτή παίρνει όλες τις τιμές της εγγραφής ,καλεί την συνάρτηση update της κλάσης RecordDB για να την ενημερώσει και κλείνει το Editform.
- Τέλος επανεμφανίζεται το jTableMyCollection (το jPanelMyCollection στην ουσία, κλήση της myCollection()) έτσι ώστε να είναι συνεπής με τις τελευταίες αλλαγές που έγιναν.

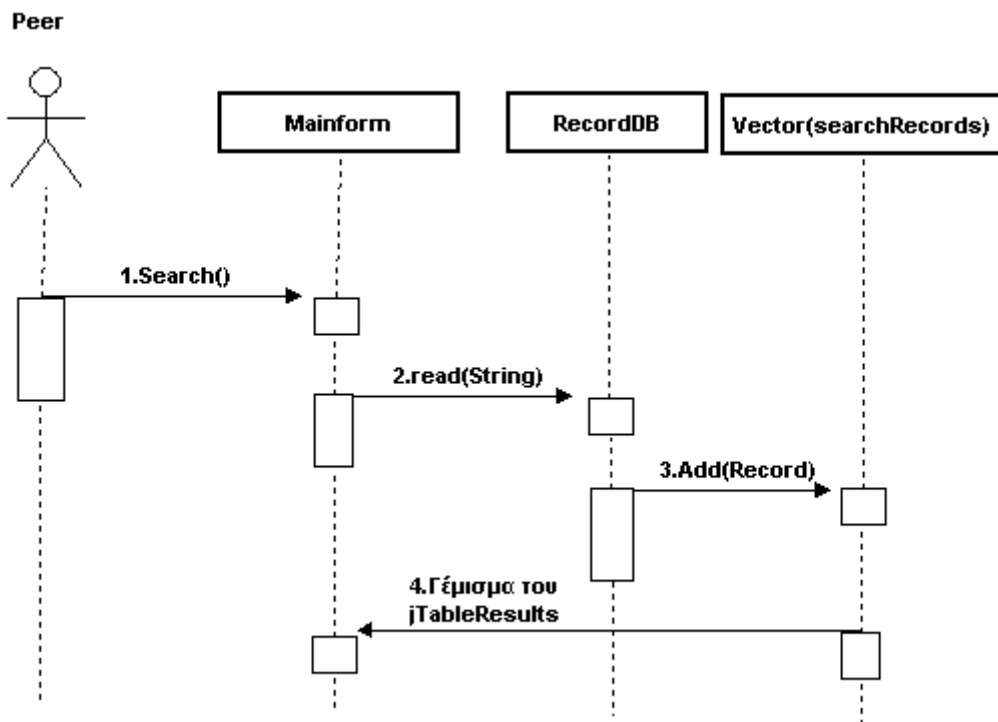
Διάγραμμα Ακολουθίας



6.Αναζήτηση εγγραφές στην τοπική συλλογή.

- Περίπτωση χρήσης που χρησιμοποιείται από οποιαδήποτε συνδεδεμένο ομότιμο μέλος
- Ο χρήστης εισάγει /επιλέγει τις τιμές που θέλει στα jTextFieldAuthor, jTextFieldGeneral, jTextFieldKey, jTextFieldTitle, jTextFieldYear, jComboBoxCategory και επιλέγει το jButtonLocal από το JPanelTarget.
- Πατάει το jButtonSearch και καλείται η μέθοδος Search η οποία παίρνει όλες τις τιμές που έχουν εισαχθεί ως περιορισμοί στην αναζήτηση.
- Αν δεν έχει εισαχθεί τίποτα εμφανίζεται κατάλληλο μήνυμα
- Διαφορετικά καλεί την συνάρτηση read της κλάσης RecordDB για να κάνει την αναζήτηση.
- Αν δεν υπάρχουν αποτελέσματα εμφανίζει κατάλληλο μήνυμα
- Διαφορετικά γεμίζει τον πίνακα jTableResults με τις εγγραφές-αποτέλεσμα της αναζήτησης.

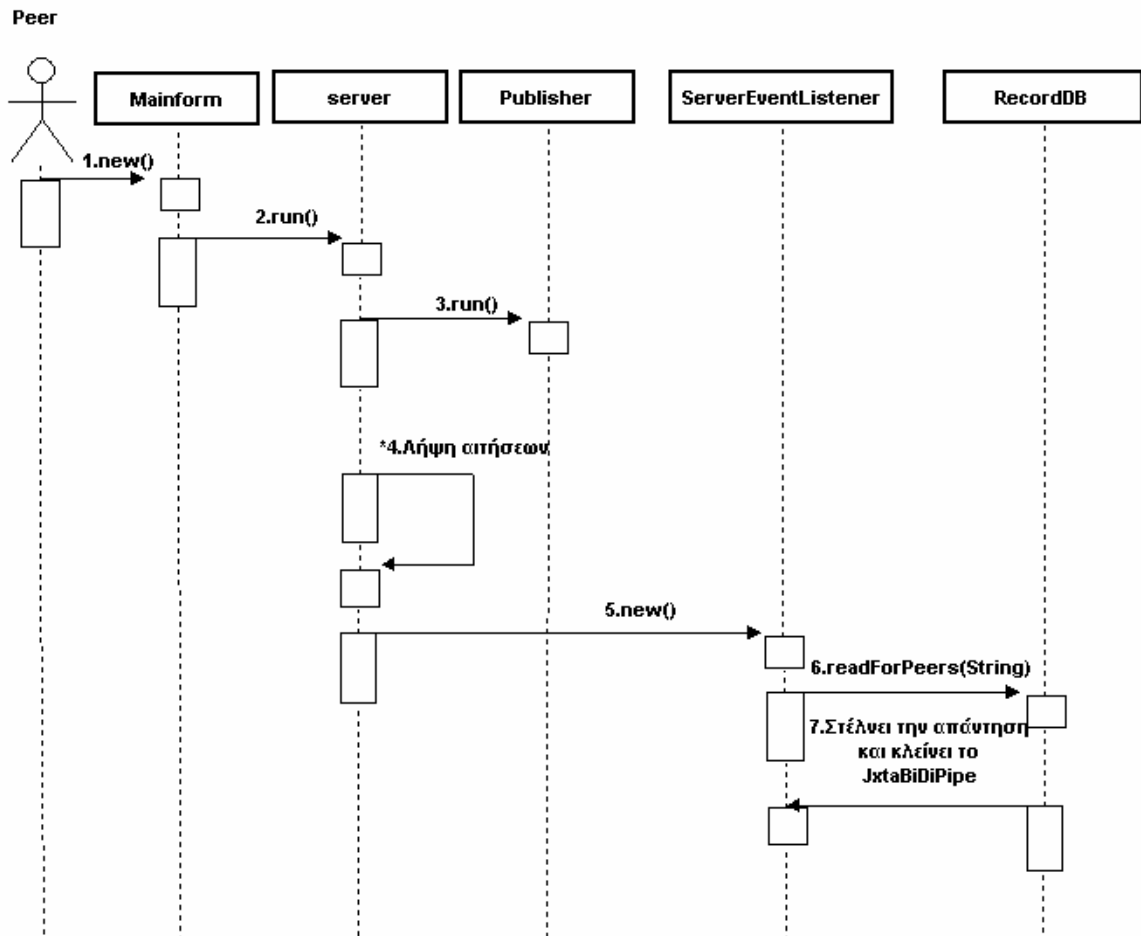
Διάγραμμα Ακολουθίας



<h2>7.Απάντηση σε ερωτήσεις από τα άλλα ομότιμα μέλη</h2>

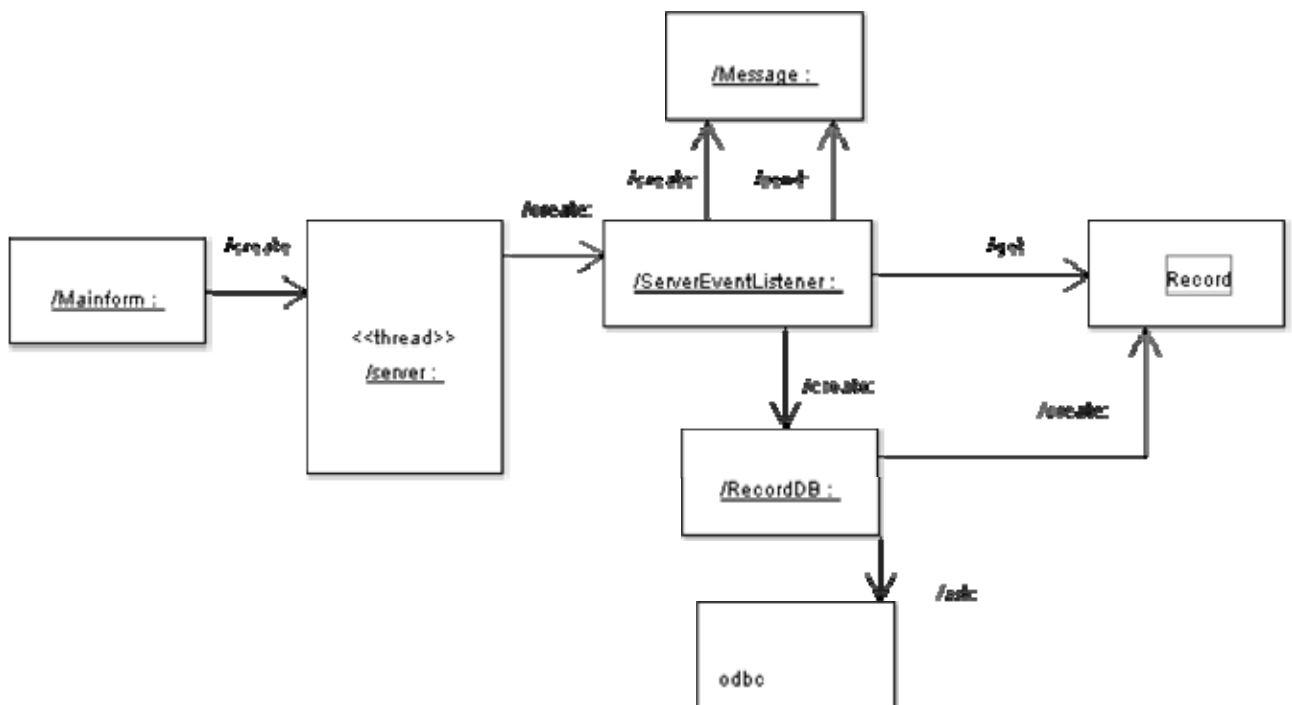
- Περίπτωση χρήσης που ενεργοποιείται όταν οποιαδήποτε ομότιμο μέλος συνδέεται στο σύστημα.
- Μόλις εμφανιστεί η MainForm τότε δημιουργείται ένα αντικείμενο κλάσης server.Το τελευταίο εκτελείται με την μέθοδο start αφού πρόκειται για ένα νήμα και η μέθοδος run της κλάσης server καλείται .
- Αυτή η μέθοδος καλεί την συνάρτηση startJxta για να πάρει την υπηρεσία ανακάλυψης και την startServer για να δημιουργήσει, ρυθμίσει και να δημοσιεύσει την διαφήμιση κλάσης μονάδας και την διαφήμιση προδιαγραφής μονάδας(η οποία περιέχει και την διαφήμιση της διασωλήνωσης) η οποία πρέπει να ανακαλυφθεί από κάποιο άλλο ομότιμο μέλος για να μπορέσει να επικοινωνήσει με το συγκεκριμένο ομότιμο μέλος.
- Επιπλέον η μέθοδος startServer δημιουργεί και ξεκινά ένα νήμα κλάσης Publisher για να δημοσιεύει σε τακτά χρονικά διαστήματα τις δύο διαφημίσεις που αναφέραμε παραπάνω όσο το ομότιμο μέλος είναι συνδεδεμένο.
- Αλλά και η ίδια η startServer έχει έναν ατέρμονα βρόγχο έτσι ώστε να λαμβάνει τις αιτήσεις για σύνδεση με αυτό το ομότιμο μέλος. Όταν ληφθεί μια δημιουργείται ένα αντικείμενο κλάσης ServerEventListener για να διαχειριστεί αυτήν την αίτηση. Ο ServerEventListener κάθε φορά που λαμβάνει ένα μήνυμα στην ουσία την ερώτηση καλεί την συνάρτηση readForPeers της κλάσης RecordDB ,δημιουργεί ένα ειδικό μήνυμα που περιέχει όλες αυτές και το στέλνει πίσω στον ενδιαφερόμενο. Η μορφή του μηνύματος είναι η εξής:
Κάθε εγγραφή ξεκινά και κλείνει με το σύμβολο ^ και τα πεδία της εγγραφής διαχωρίζονται με το σύμβολο &.
- Τέλος κλείνει το JxtaBiDiPipe που συνδέει τα δύο ομότιμα μέλη.

Διάγραμμα Ακολουθίας



Για τον λόγο ότι η συγκεκριμένη περίπτωση χρήσης είναι σημαντική ακολουθεί το διάγραμμα συνεργασίας αυτής.

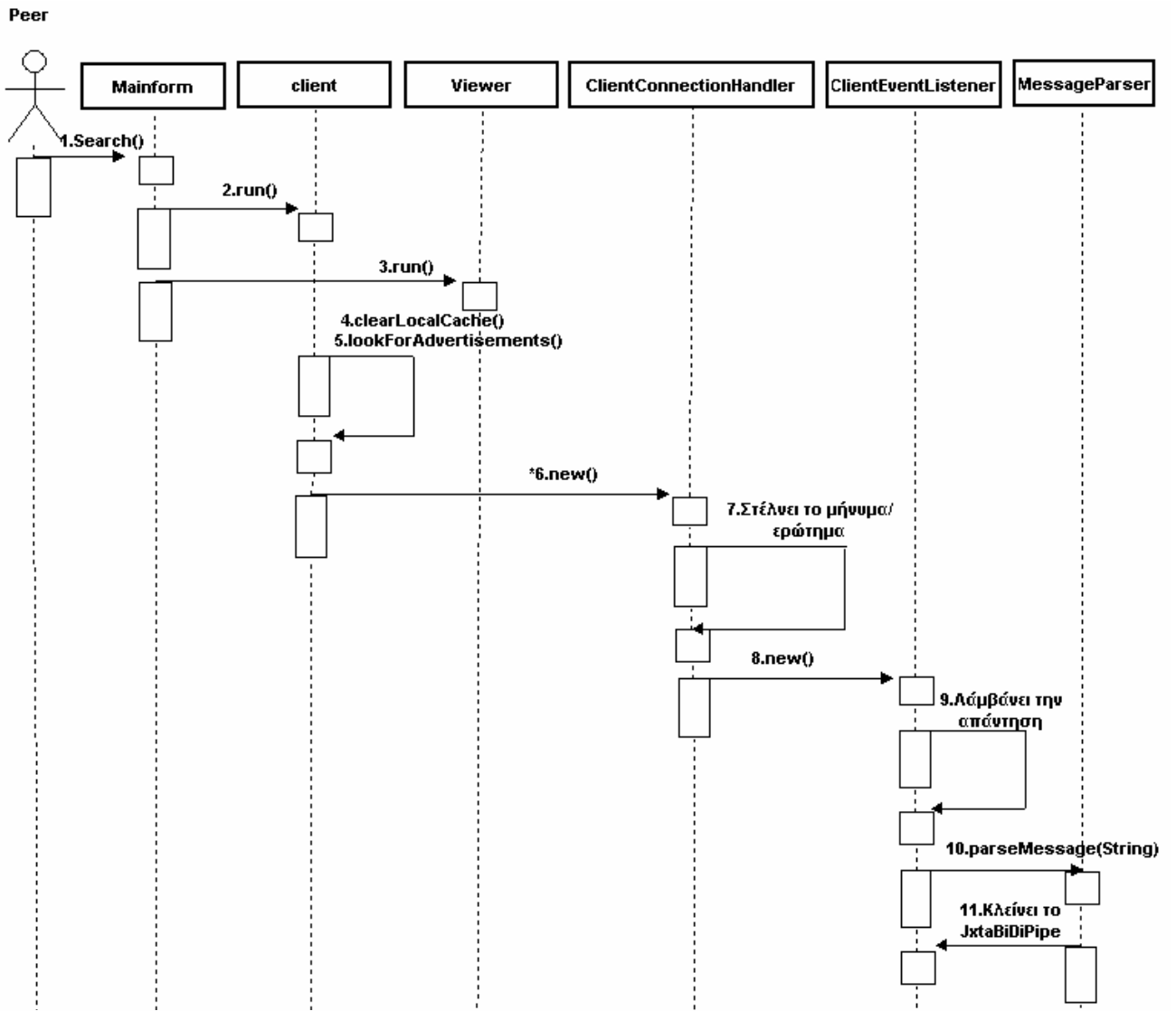
Διάγραμμα Συνεργασίας



8.Αναζήτηση εγγραφές στο ομότιμο δίκτυο

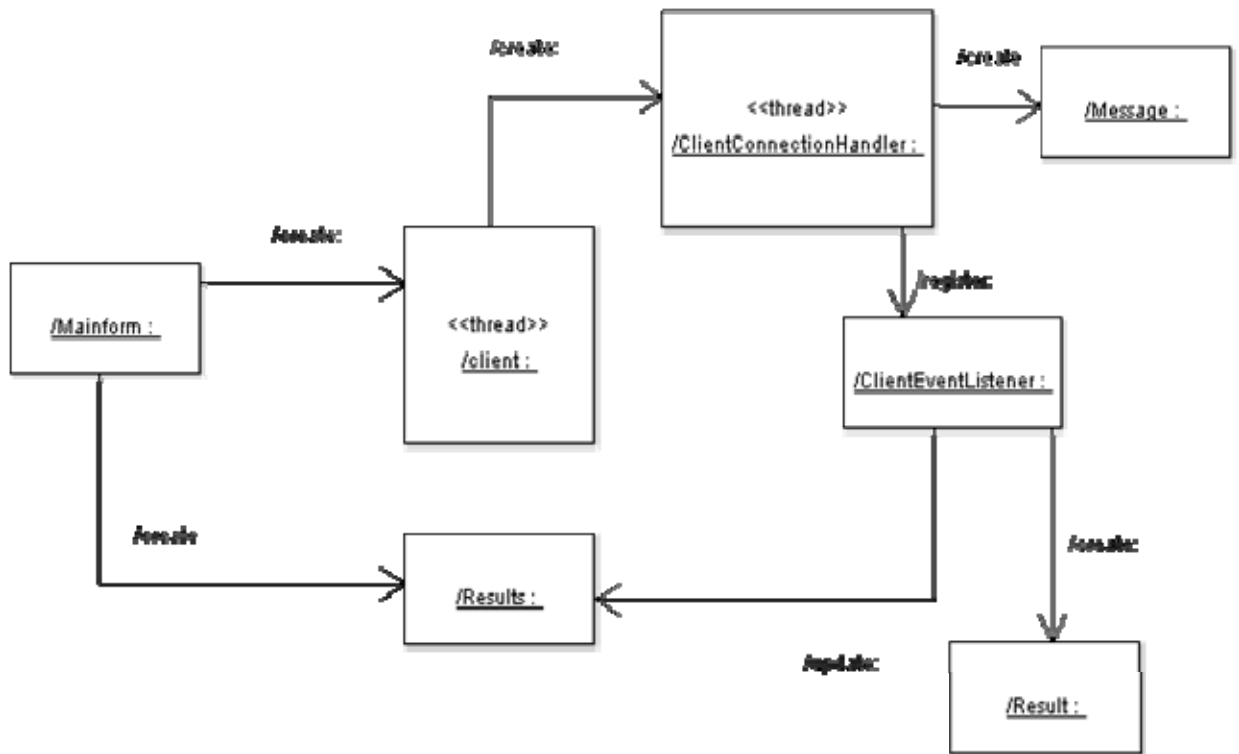
- Περίπτωση χρήσης που χρησιμοποιείται από οποιαδήποτε συνδεδεμένο ομότιμο μέλος
- Ο χρήστης εισάγει /επιλέγει τις τιμές που θέλει στα `jTFieldAuthor`, `jTFieldGeneral`, `jTFieldKey`, `jTFieldTitle`, `jTFieldYear`, `jCBoxCategory` και επιλέγει το `JRButtonRemote` από το `JPanelTarget`.
- Πατάει το `JButtonSearch` και καλείται η μέθοδος `Search` η οποία παίρνει όλες τις τιμές που έχουν εισαχθεί ως περιορισμοί στην αναζήτηση.
- Αν δεν έχει εισαχθεί τίποτα εμφανίζεται κατάλληλο μήνυμα
- Διαφορετικά δημιουργείται ένα αντικείμενο κλάσης `client` .Το αντικείμενο αυτό εκτελείται με την μέθοδο `start` αφού πρόκειται για ένα νήμα και η μέθοδος `run` της κλάσης `client` καλείται .
- Ταυτόχρονα δημιουργεί και ξεκινά ένα άλλο νήμα `Viewer` το οποίο ανανεώνει όσο διαρκεί η αναζήτηση στο ομότιμο δίκτυο τα περιεχόμενα του `JTableResults`.
- Το νήμα `client` καλεί την μέθοδο `startJxta` για να πάρει την υπηρεσία ανακάλυψης και την `startClient` η οποία αρχικά καλεί την μέθοδο `clearLocalCache` για να σβήσει όλες τις διαφημίσεις που είναι γνωστές και την μέθοδο `lookForAdvertisements` για να ανακαλύψει όλες τις διαφημίσεις εκ νέου.
- Έπειτα για κάθε διαφήμιση υπηρεσίας που λαμβάνει, δημιουργεί και ξεκινά ένα `ClientConnectionHandler` το οποίο παίρνει την διαφήμιση διασώληνωσης από την υπηρεσία , δημιουργεί ένα σημείο τέλους διασώληνωσης εξαγωγής και στέλνει ως μήνυμα τους περιορισμούς που ορίστηκαν ως περιορισμοί στην αναζήτηση από το ομότιμο μέλος. Ταυτόχρονα ορίζει έναν `ClientEventListener` για να λαμβάνει τα μηνύματα από την συγκεκριμένη σύνδεση .Ο `ClientEventListener` όταν λάβει ένα μήνυμα το αναλύει και εξάγει τις εγγραφές που προέκυψαν από την αναζήτηση στον απομακρυσμένο ομότιμο μέλος με την κλήση της μεθόδου `parseMessage` της κλάσης `MessageParser`.
- Και τέλος κλείνει το `JxtaBiDiPipe` που συνδέει τα δύο ομότιμα μέλη.

Διάγραμμα Ακολουθίας



Για τον λόγο ότι η συγκεκριμένη περίπτωση χρήσης είναι σημαντική ακολουθεί το διάγραμμα συνεργασίας αυτής.

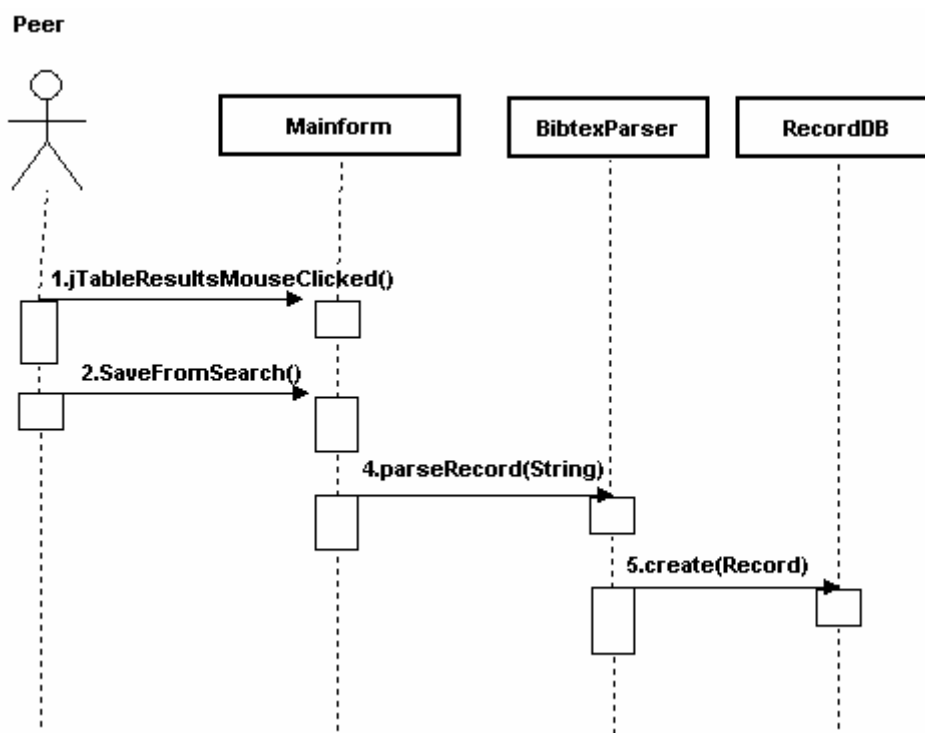
Διάγραμμα Συνεργασίας



9.Αποθήκευσε την εγγραφή που προέκυψε από αναζήτηση στο ομότιμο δίκτυο

- Περίπτωση χρήσης που χρησιμοποιείται από οποιαδήποτε συνδεδεμένο ομότιμο μέλος
- Το μέλος πρέπει να έχει εκτελέσει ένα ερώτημα αναζήτησης στο ομότιμο δίκτυο και να έχουν προκύψει μηδέν ή περισσότερες εγγραφές.
- Αν ισχύει το παραπάνω το ομότιμο μέλος επιλέγει την εγγραφή από τον jTableResults που επιθυμεί να αποθηκεύσει και ενεργοποιείται η μέθοδος jTableResultsMouseClicked που εμφανίζει την επιλεγμένη εγγραφή σε μορφή BibTeX στην περιοχή κειμένου jTAreaViewResults .Στην συνέχεια ο χρήστης πατάει το κουμπί jButtonSave και καλείται η μέθοδος SaveFromSearch.
- Η μέθοδος αυτή λαμβάνει το κείμενο που εμφανίζεται στην περιοχή κειμένου jTAreaViewResults το αναλύει και συνθέτει την νέα εγγραφή με την κλήση της μεθόδου parseRecord της κλάσης BibtexParser και την αποθηκεύει στην τοπική μνήμη με την μέθοδο create της κλάσης RecordDB.

Διάγραμμα Ακολουθίας



Προτάσεις για μελλοντική επέκταση της εφαρμογής

Στο μέλλον θα μπορούσαν να προστεθούν τα παρακάτω για την επέκταση /βελτίωση της εφαρμογής

- Να δίνεται η δυνατότητα εγγραφής των δεδομένων σε XML. Η μη επίσημα καθορισμένη γραμματική του BibTeX δεν επιτρέπει μια αποδοτική διαχείριση δεδομένων. Σχηματίζοντας τις XML βιβλιογραφίες με την XML μπορούμε να επιτύχουμε την δημιουργία πιο ισχυρών βιβλιογραφικών βάσεων δεδομένων και το χειρισμό τους χρησιμοποιώντας γενικού σκοπού XML εργαλεία .
- Η λύση που προτείνεται από την συγκεκριμένη εργασία αφορά περιβάλλον για λίγα ομότιμα μέλη. Για την επέκταση αυτής θα μπορούσε να υλοποιηθεί ένα πρωτόκολλο εύρεσης(JXTA Search)
- Να μπορούν να αποδοθούν βαθμοί σε κάθε ομότιμο. Ομότιμοι κόμβοι που συνεχώς εμφανίζουν στη βάση τους εγγραφές που πολλοί άνθρωποι βρήκαν σημαντικές, να έχουν υψηλότερους βαθμούς. Επιπλέον θα μπορούσε η έρευνα να επικεντρώνεται για μεγαλύτερες πιθανότητες επιτυχίας πρώτα σε τέτοιους ομότιμους κόμβους
- Να μπορούν να αποδοθούν βαθμοί δημοτικότητας σε κάθε εγγραφή ώστε να παράγονται στατιστικά στοιχεία για τις πιο δημοφιλείς εγγραφές
- Να μπορούν να δημιουργούνται ομάδες από ομότιμους κόμβους με ειδικότητα σε συγκεκριμένα θέματα (για παράδειγμα ομάδες από ομότιμους που έχουν βιβλιογραφικά δεδομένα πάνω στην Τεχνολογία Λογισμικού)

ΣΧΟΛΙΑ ΦΟΙΤΗΤΡΙΑΣ

- Η χρησιμοποίηση της μεθοδολογίας Rational Unified Process (RUP) έπαιξε σημαντικό ρόλο στην επιτυχής ολοκλήρωση της εφαρμογής .Λόγω του ότι η ανάπτυξη έγινε επαναλαμβανόμενα, ο κίνδυνος στο οδηγηθούμε σε κάποιο λανθασμένο αποτέλεσμα μετριάστηκε από νωρίς. Επιπλέον έγινε αποτελεσματική διαχείριση των απαιτήσεων, και μειώθηκε η πολυπλοκότητα με την οπτική μοντελοποίηση του προβλήματος.
- Προβλήματα κατά την ανάπτυξη
Κατά την υλοποίηση της εφαρμογής με την πλατφόρμα επικοινωνίας JXTA διαπιστώθηκε ότι κατά την δοκιμή του προγράμματος η συγκεκριμένη πλατφόρμα είχε κάποιο bug σε ότι αφορά τα παλιά advertisements. Δηλαδή τα advertisements που ανακαλύφθηκαν από μια προηγούμενη αναζήτηση .Τα οποία περισσότερο καθυστερούσαν το πρόγραμμα παρά το βοηθούσαν με αποτέλεσμα κάποιες φορές κάποιο ομότιμο μέλος να καθυστερεί να εντοπίσει κάποιο άλλο ομότιμο. Έγινε προσπάθεια επίλυσης του προβλήματος αυτού διαγράφοντας τα παλιά advertisements με την έναρξη κάθε νέας αναζήτηση αλλά πάλι δεν είχαμε αποτέλεσμα. Αυτό το bug είναι απόρροια της δοκιμής της εφαρμογής σε μη πραγματικό περιβάλλον. Σε αντίθετη περίπτωση δεν θα υπήρχε κανένα πρόβλημα.
- Κατά την υλοποίηση της πτυχιακής μου εργασίας μου δόθηκε η ευκαιρία να ολοκληρώσω ένα έργο εφαρμόζοντας όλα τα στάδια του κύκλο ανάπτυξης μιας εφαρμογής ακολουθώντας την μεθοδολογία RUP. Αποκόμισα πολλές και σημαντικές γνώσεις τόσο σε θεωρητικό όσο και σε πρακτικό επίπεδο(ανάπτυξη εφαρμογής). Είχα μια πρώτη επαφή με έννοιες που αφορούσαν κυρίως την πλατφόρμα επικοινωνίας JXTA ,τα P2P δίκτυα, την γλώσσα προγραμματισμού Java καθώς και την βιβλιογραφική μορφή BibTeX. Έμαθα να υλοποιώ εφαρμογές σε πολυνηματικό περιβάλλον, αλλά και να εκμεταλλεύομαι στο έπακρο τις προγραμματιστικές δυνατότητες που παρέχει η Java για την επίτευξη κάποιου στόχου.

Πηγές

Αναφορές

- [1] Oren Patashnik, "BibTEXing", February 1988
- [2] S. Milojicic, Dejan and Kalogeraki, Vana and Lukose, Rajan and Nagaraja, Kiran and Pruyne, Jim and Richard, Bruno and Rollins, Sami and Xu, Zhichen," Peer-to-Peer Computing", HP Laboratories, Palo Alto, March 2002, <http://www.hpl.hp.com/techreports/2002/HPL-2002-57.pdf>
- [3] <http://www.akamai.com/en/html/technology/overview.html>
- [4] <http://www.fast-track.com/about.php>
- [5] http://www.imesh.org/toolkit/work/tech_review/
- [6] J. Kubiatoiwicz, et al. OceanStore: An Architecture for Global-Scale Persistent Storage. ASPLOS, December 2000. <http://citeseer.ist.psu.edu/kubiatoiwicz00oceanstore.html>
- [7] Antony I. T. Rowstron and Peter Druschel, "Pastry: Scalable, distributed object location and routing for large-scale peer-to peer systems," in Middleware, 2001. <http://citeseer.ist.psu.edu/rowstron01pastry.html>
- [8] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for Internet applications. Technical Report TR-819, MIT, March 2001. <http://citeseer.ist.psu.edu/stoica01chord.html>
- [9] Ratnasamy, S., Francis, P., Handley, M., Karp, R., Shenker, S.: A scalable content addressable network. In: Proc. of ACM SIGCOMM '01. (2001)
- [10] Sun Microsystems, "Project JXTA: Java™ Programmers Guide", 2005, http://www.jxta.org/docs/JxtaProgGuide_v2.3.pdf
- [11] R. Gummadi, S. Gribble, S. Ratnasamy, S. Shenker, and I. Stoica. The impact of dht routing geometry on resilience and proximity. In Proc. ACM SIGCOMM'03, Karlsruhe, Germany, 2003. <http://citeseer.ist.psu.edu/gummadi03impact.html>
- [12] Roman Schmidt, " Gridella: an open and efficient Gnutella-compatible Peer-to-Peer System based on the P-Grid approach", <http://citeseer.ist.psu.edu/schmidt02gridella.html>
- [13] Karl Aberer, P-Grid: A self-organizing access structure for P2P information systems ,Sixth International Conference on Cooperative Information Systems (CoopIS 2001), Trento, Italy, Lecture Notes in Computer Science 2172, Springer Verlag, Heidelberg, 2001. <http://www.p->

- grid.org/Papers/CoopIS2001.pdf
- [14] www.webrankinfo.com/english/pagerank/random-walker.php
- [15] A. Adya, W. J. Bolosky, M. Castro, R. Chaiken, G. Cermak, J. R. Douceur, J. Howell, J. R. Lorch, M. Theimer, R. P. Wattenhofer, "FARSITE: Federated, Available, and Reliable Storage for an Incompletely Trusted Environment", 5th OSDI, Dec 2002. <http://citeseer.ist.psu.edu/adya02farsite.html>
- [16] Michael Young, "XML Step by Step", Microsoft Corporation, 2000
- [17] Bruce Eckel, "Thinking in Java, 3rd edition", Prentice-Hall, December 2002
- [18] Patrick W. Daly, "A LATEX Package to Place Bibliography Entries in Text", February 1999.
- [19] Broekstra, J., Ehrig, M., Haase, P., Harmelen, F., Menken, M., Mika, P., Schnizler, B., Siebes, R.: Bibster - a semantics-based bibliographic peer-to-peer system. In: Proceedings of the WWW'04 Workshop on Semantics in Peer-to-Peer and Grid Computing. (2004)
<http://citeseer.ist.psu.edu/broekstra04bibster.html>
- [20] Jacobson, Booch, Rumbaugh, "The Unified Software Development Process" Addison-Wesley 1999

Ηλεκτρονικές διευθύνσεις

- P2P δίκτυα
<http://en.wikipedia.org/wiki/Peer-to-peer>
<http://compnetworking.about.com/od/p2ppeertopeer/a/p2pintroduction.htm>
<http://p2p.info.pl/eng/mod-subjects-viewpage-pageid-46.phtml>
<http://www.compinfo-center.com/int/p2p.htm>
<http://www.eecg.toronto.edu/~jacobsen/mie456/slides/p2p-mie.pdf>
- JXTA
<http://www.jxta.org>
- JAVA
<http://java.sun.com>
- BibTeX
<http://en.wikipedia.org/wiki/BibTeX>
<http://ntrg.cs.tcd.ie/undergrad/4ba2.02-03/Intro.html>
http://artis.imag.fr/Membres/Xavier.Decoret/resources/xdkbibtex/bibtex_summary.html
http://www-ftp.lip6.fr/pub/tex-archive/info/bibtex/tamethebeast/ttb_en.pdf
- Bibster
<http://bibster.semanticweb.org>