

ΑΝΩΤΑΤΟ
ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ
ΚΡΗΤΗΣ

ΤΜΗΜΑ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ:

Υποστήριξη Σύγχρονων Συνεργατικών Καθηκόντων &
Μηχανισμοί ανάπτυξης σύγχρονων διεπαφών
Η περίπτωση χρήσης του eKoNEΣ

Βελλής Γεώργιος
Επιβλέπων καθηγητής: Δημοσθένης Ακουμιανάκης
Ημ/νια παρουσίασης: 17/5/2007

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ:

| | | |
|----------|--|-----------|
| 1 | Εισαγωγή | 6 |
| 2 | Ηλεκτρονικές κοινότητες και κοινωνικό λογισμικό | 7 |
| 2.1 | <i>Κοινωνικό Λογισμικό (Social Software)</i> | 7 |
| 2.2 | <i>Ιστορική αναδρομή και ορολογία</i> | 9 |
| 2.3 | <i>Διαφορά κοινωνικού λογισμικού από το “συμβατικό”</i> | 11 |
| 2.4 | <i>Ευκολίες που παρέχει το κοινωνικό λογισμικό στο Διαδίκτυο</i> | 11 |
| 2.5 | <i>Είδη δικτύων που χρησιμοποιούν κοινωνικό λογισμικό</i> | 12 |
| 2.6 | <i>Εφαρμογές κοινωνικού λογισμικού</i> | 14 |
| 2.7 | <i>Δυσκολίες σχεδίασης κοινωνικού λογισμικού</i> | 16 |
| 2.8 | <i>Πληροφορίες για τη σχεδίαση</i> | 17 |
| 2.9 | <i>Θετικά και αρνητικά για το χρήστη και την εφαρμογή</i> | 18 |
| 3 | Εισαγωγή στις έννοιες των συνεργατικών συστημάτων | 20 |
| 3.1 | <i>Model View Controller και Συνεργατικά συστήματα</i> | 22 |
| 3.1.1 | <i>Τι είναι η Model View Controller Αρχιτεκτονική?</i> | 22 |
| 3.1.2 | <i>Το Model View Controller στην πράξη</i> | 22 |
| 3.2 | <i>Μηχανισμοί Σύγχρονης Συνεργασίας</i> | 23 |
| 3.3 | <i>Διαχείριση συνεδρίας (Session Management)</i> | 24 |
| 3.4 | <i>Διαχείριση Δαπέδου (Floor Control)</i> | 24 |
| 3.5 | <i>Μοντέλο Αντιγράφων Αντικειμένων</i> | 25 |
| 3.5.1 | <i>Αρχιτεκτονική πλήρους Αντιγραφής</i> | 26 |
| 3.5.2 | <i>Αρχιτεκτονική μερικής Αντιγραφής</i> | 28 |
| 4 | Τεχνολογίες Δικτύου Συνεργατικών Framework | 31 |
| 4.1 | <i>Πρωτοκόλλα και Υπηρεσίες</i> | 31 |
| 4.2 | <i>Sockets</i> | 33 |
| 4.3 | <i>Common Object Broker</i> | 35 |
| 4.4 | <i>Remote Method Invocation</i> | 36 |
| 4.5 | <i>XML-Remote Procedure Call</i> | 38 |
| 4.6 | <i>Simple Object Access Protocol</i> | 41 |
| 5 | Τι είναι το eΚοΝΕΣ? | 46 |
| 5.1 | <i>Μια σύντομη επισκόπηση</i> | 46 |
| 5.2 | <i>Χρήστες του eΚοΝΕΣ και ρόλοι αυτών</i> | 47 |
| 5.2.1 | <i>Καθήκοντα διαχειριστή eΚοΝΕΣ</i> | 49 |
| 5.2.2 | <i>Καθήκοντα επιχειρηματικών εταίρων</i> | 49 |
| 5.2.3 | <i>Ο τελικός χρήστης</i> | 50 |
| 5.3 | <i>Ποια η θέση του Σύγχρονου Collaboration στο eΚοΝΕΣ?</i> | 50 |
| 5.3.1 | <i>Προϋποθέσεις για το χαρακτηρισμό ενός συστήματος ως Framework</i> | 51 |
| 5.3.2 | <i>Πλεονεκτήματα και Μειονεκτήματα ενός Framework</i> | 51 |

| | | |
|----------|--|-----------|
| 6 | Υλοποίηση του eKoNEΣ Collaborative Framework..... | 53 |
| 6.1 | <i>Το Σενάριο</i> | 53 |
| 6.2 | <i>Ο Collaboration Server.....</i> | 58 |
| 6.3 | <i>Ο διαχειριστής.....</i> | 62 |
| 6.4 | <i>Οι εταίροι</i> | 67 |
| 7 | ΒΙΒΛΙΟΓΡΑΦΙΑ | 73 |
| 8 | ΠΑΡΑΡΤΗΜΑ | 75 |

Ευχαριστίες

Θέλω να εκφράσω τις θερμότερες και πιο ειλικρινείς ευχαριστίες μου στους ανθρώπους που με βοήθησαν να φέρω εις πέρας την πτυχιακή μου. Πιο συγκεκριμένα θα ήθελα να ευχαριστήσω: τον κο Δημοσθένη Ακουμιανάκη, τον κο Νικόλαο Βιδάκη για την πολύτιμη βοήθειά τους. Τέλος θα ήθελα να ευχαριστήσω τους φίλους, συμφοιτητές και συνεργάτες Δημήτρη και Γιάννη στο εργαστήριο ISTLab για την άψογη συνεργασία που έχουμε.

ΠΡΟΛΟΓΟΣ

Η πτυχιακή αυτή είναι μέρος της δουλειάς που εκπονήθηκε στο ερευνητικό πρόγραμμα eKoNES¹(*eΚονικές κοινότητες Επιχειρηματικότητας & καινοτομίαΣ στην περιφέρεια*) και αφορά την ενότητα εργασίας που έχει να κάνει με τη σύγχρονη επικοινωνία μεταξύ χωρικά διεσπαρμένων μελών μιας ηλεκτρονικής κοινότητας.

Συγκεκριμένα άπτεται πολλών επιστημονικών γνωστικών αντικειμένων, άλλων αρκετά εξειδικευμένων και άλλων όχι και τόσο, λόγω της διεπιστημονικής θεματικής ευρύτητας που καλύπτει το γνωστικό αντικείμενο της συνεργασίας υποστηριζόμενης από υπολογιστή γενικότερα (*Computer Supported Cooperative Work*).

Ο συγγραφέας προϋποθέτει ότι ο αναγνώστης έχει μια βασική εξοικείωση με θέματα που αφορούν γραφικές βιβλιοθήκες αλληλεπίδρασης(GUI toolkits), σχεδίαση διεπαφών με τη βοήθεια αυτών καθώς και βασική γνώση της αρχιτεκτονικής σχεδίασης και του ενδότερου τρόπου λειτουργίας αυτών. Επίσης, προϋποθέτει ότι υπάρχει εξοικείωση σε θέματα που αφορούν τεχνολογίες δικτύου, προγραμματισμό καθοδηγούμενο από events(event-driven programming), αλλά και πιο εξειδικευμένων θεμάτων. Τέτοια θέματα έχουν να κάνουν παραδείγματος χάριν με επαύξηση(augmentation) και επέκταση(expansion) αντικειμένων γραφικών βιβλιοθηκών (π.χ. Java Swing) ή με θέματα που αφορούν τη διαλειτουργικότητα συνιστωσών τμημάτων (Layers, parts ή modules) ενός συστήματος (interoperability).

Κάποια από αυτά θα επεξηγηθούν όπου χρειαστεί, αλλά σε βαθμό τέτοιο ώστε να μην αναλωθούμε στην κατανόηση συνιστωσών τμημάτων και επιμέρους θεωρητικών αντικειμένων που ναι μεν είναι βασικές και απαραίτητες, αλλά θα μας οδηγήσουν στο να χάσουμε την ουσία πλατειάζοντας. (Θα δοθεί παρατάυτα χώρος για την επεξήγηση όποιων από τις τεχνολογίες θεωρούνται χρήσιμες καθώς και πηγές ώστε να μπορεί να κατατοπίζεται πλήρως και ανά πάσα στιγμή ο αναγνώστης)

¹ Βλέπε: <http://www.e-kones.teiher.gr>

1 Εισαγωγή

Όπως αναφέραμε και στον πρόλογο η πτυχιακή αυτή εργασία αφορά το ερευνητικό πρόγραμμα eKoNEΣ. Το eKoNEΣ έχει ως βασικό σκοπό να συνεισφέρει στην ανάπτυξη της θεωρητικής βάσης και της τεχνολογικής υποδομής για την σύσταση και λειτουργία ηλεκτρονικών χωριών ή ηλεκτρονικών κοινοτήτων τοπικής κλίμακας με σκοπό την διεπιχειρησιακή «εικονική δικτύωση» εταιρών (παραγωγικών επιχειρήσεων, ΟΤΑ & πολιτών-καταναλωτών), για την ενίσχυση της τεχνολογικής τους βάσης και την προαγωγή καινοτομιών (π.χ. νέα προϊόντα & υπηρεσίες) μέσω προηγμένων μορφών συνεργασίας. Το eKoNEΣ θα επιδείξει τα αποτελέσματά του στον κλάδο του τουρισμού ενώ στο μέλλον αναμένεται η κλιμάκωση της εφαρμογής σε άλλους κλάδους της τοπικής βιομηχανίας.

Συνοψίζοντας σκοπός μας είναι η ανάπτυξη-παραγωγή μιας συνιστώσας του eKoNEΣ, που θα αναλάβει τη διαχείριση και υποστήριξη σύγχρονων συνεργατικών καθηκόντων μεταξύ μελών μιας οποιασδήποτε κοινότητας. Τα κύρια μέρη της συνιστώσας αυτής είναι ο διαχειριστής συνεδρίας, ο ελεγκτής δαπέδου και το μοντέλο διαχείρισης αντίγραφου αντικειμένου(Object Replication Model). Τα παραπάνω εξασφαλίζουν τις βασικές ιδιότητες ενός συνεργατικού περιβάλλοντος που είναι η γνωστοποίηση γεγονότων, η επίγνωση εταιρών, και η ανάδραση ενός εταίρου στις πράξεις ενός άλλου.

2 Ηλεκτρονικές κοινότητες και κοινωνικό λογισμικό

Ο όρος «ηλεκτρονικό χωριό» ή «Electronic Village» χρησιμοποιείται για να περιγράψει τη χρήση τεχνολογικών εργαλείων επικοινωνίας και συνεργασίας για την υποστήριξη της πληροφόρησης και του διαλόγου μεταξύ μελών μιας κοινότητας σχετικά με θέματα τοπικού ενδιαφέροντος (π.χ. δραστηριότητες, επιχειρηματικότητα, διακυβέρνηση, κλπ).

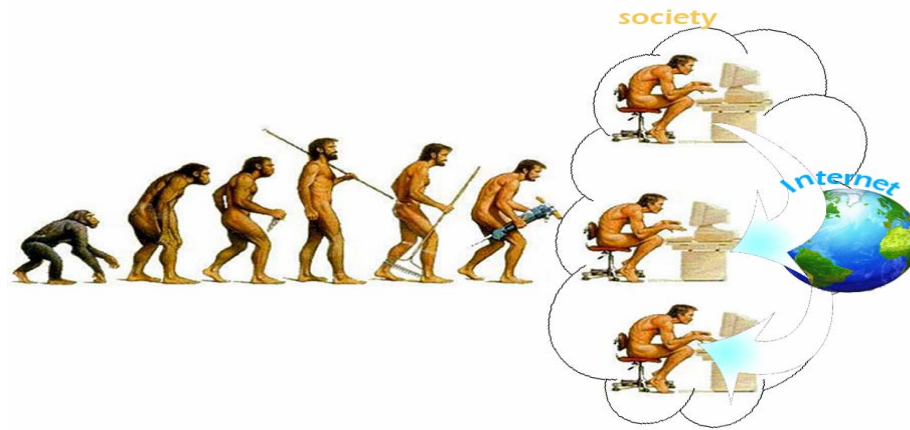
Για πολλούς, τα τοπικά ηλεκτρονικά χωριά αποτελούν μια ανταγωνιστική άποψη που αντιπαρέρχεται τα αρνητικά της παγκοσμιοποίησης υπερθεματίζοντας τη στενή σχέση μεταξύ εικονικών κοινοτήτων και φυσικών χώρων. Η ιδιάζουσα αυτή προσέγγιση, στην οποία βασίζεται και το έργο eKoNES, έχει κατά το πρόσφατο παρελθόν αποφέρει πολύ σημαντικά αποτελέσματα, των οποίων τα ποιοτικά χαρακτηριστικά σφραγίζουν την σκοπιμότητα και τη βιωσιμότητα τέτοιων εγχειρημάτων.

Η συνεργασία υποστηριζόμενη από υπολογιστή έχει δύο εκδοχές, τη συνεργασία μέσω σύγχρονης και ασύγχρονης επικοινωνίας. Πριν όμως ξεκινήσουμε την ανάλυση των δύο αυτών συνιστωσών, είναι χρήσιμο να μελετήσουμε κάποιες βασικές και θεμελιώδεις έννοιες στο χώρο αυτό, είτε μιλάμε για σύγχρονη είτε για ασύγχρονη επικοινωνία.

2.1 Κοινωνικό Λογισμικό (Social Software)

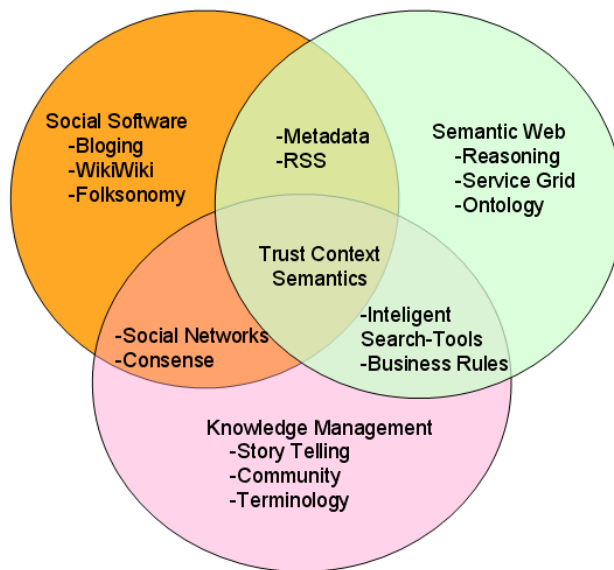
Για πολλά έτη τα e-mail και τα Usenet news (με τον όρο Usenet εννοούμε ένα σύστημα για την ανταλλαγή μηνυμάτων με τη μορφή άρθρων ταξινομημένα σύμφωνα με συγκεκριμένες κατηγορίες που ονομάζονται newsgroups) αποτέλεσαν την πλειοψηφία της χρήσης του Διαδικτύου ως εργαλείο για να διευκολύνουν την επικοινωνία μεταξύ των ατόμων. Τα τελευταία χρόνια έχουμε διάφορες νέες εφαρμογές σε αυτή την περιοχή που είναι γνωστές ως social software. Ο όρος social software ισχύει για οποιοδήποτε εργαλείο το οποίο επιτρέπει σε δύο ή περισσότερα άτομα να συνεργαστούν ενώ βρίσκονται σε διαφορετική θέση. Η συνεργασία μπορεί να εμφανιστεί σε πραγματικό χρόνο (σύγχρονη συνεργασία) ή σε διαφορετικούς χρόνους (ασύγχρονη συνεργασία), ενώ οι θέσεις μπορούν να αφορούν ηπείρους ή απλά ένα κτήριο γραφείων. Εν ολίγοις είναι ένα λογισμικό το οποίο επιτρέπει στους χρήστες να συναντώνται, να συνδέονται και να συνεργάζονται μέσω ηλεκτρονικών υπολογιστών χωρίς να βρίσκονται στον ίδιο χώρο. Ανεξαρτήτως της απόστασης ή της

μορφής συνεργασίας, ο απώτερος στόχος του social software είναι η διαμόρφωση κοινοτήτων γνώσης στις οποίες οι συμμετέχοντες δίνουν και λαμβάνουν συνεχώς πολύτιμες πληροφορίες.



Εικόνα 1

Η εξέλιξη των εργαλείων συνεργασίας και του λογισμικού που περιγράφεται από τον όρο groupware (ουσιαστικά χρησιμοποιήθηκε πριν εμφανιστεί ο όρος social software για να περιγράψει τα προγράμματα που βοηθούν τη συνεργασία ομάδων) είναι το social software. Ενώ όμως τα δύο πρώτα υιοθέτησαν μία από πάνω προς τα κάτω μέθοδο έχοντας δημιουργήσει πρώτα τις ομάδες εργασίας χωρίζοντας τα μέλη κατά έναν οργανωτικό ή λειτουργικό τρόπο και έπειτα προσθέτοντας τους στα εκάστοτε προγράμματα, το social software αλλάζει αυτή την ιεραρχία δίνοντας περισσότερη σημασία στις πληροφορίες και τους χρήστες και έπειτα επιτρέπει να δημιουργηθούν οι ομάδες με οποιοδήποτε τρόπο θεωρείται καλύτερος, χωρίς να έχουν τεθεί κάποιοι προκαθορισμένοι κανόνες. Στο social software για παράδειγμα έχουμε την εξής ιεραρχία: αρχικά κάποιος εγγράφεται στο σύστημα και στη συνέχεια αφήνεται στην κρίση του να συναναστραφεί με συγκεκριμένα άτομα και συγκεκριμένες ομάδες ανάλογα με τα ενδιαφέροντα και τις προτιμήσεις του.



Εικόνα 2

Χαρακτηριστικά παραδείγματα αυτού του “από κάτω προς τα πάνω” τρόπου σχεδίασης αποτελούν τα δίκτυα peer-to-peer που χρησιμοποιούνται για την ανταλλαγή δεδομένων, τα blogs, τα wikis κ.α.

2.2 Ιστορική αναδρομή και ορολογία

Το social software δεν έχει σταματήσει να εξελίσσεται τα τελευταία 50 περίπου χρόνια. Συγκεκριμένα οι κεντρικές ιδέες του social software υπάρχουν πολύ πιο “πίσω” στις ιδέες του Bush Vannevar για το “memex” το 1945 ενώ συνεχίζει το ταξίδι του μέσω πολλών όρων όπως Augmentation, Groupware και CSCW στις δεκαετίες του 60, 70, 80 και 90. Επίσης με την επισκόπηση των πολλών όρων που χρησιμοποιούνται για να περιγράψουν το σημερινό “social software” μπορούμε να ερευνήσουμε την προέλευσή του, διαπιστώνοντας πως υπάρχει πραγματικά ένας κύκλος ζωής σχετικά με τη χρήση της τεχνικής ορολογίας.

Πιο αναλυτικά: η πιο παλιά αναφορά στην οποία άνθρωποι χρησιμοποιώντας τους υπολογιστές συνεργάζονταν ο ένας με τον άλλον είναι από τη δεκαετία του '40. Κοντά στο τέλος του 2ου παγκοσμίου πολέμου, το 1945, ο Bush Vannevar έγραψε ένα δημιουργικό άρθρο σχετικά με το μέλλον του «υπολογισμού». Σε αυτό, περιέγραφε μια συσκευή που την ονόμασε “memex”, την οποία σήμερα μπορούμε να παρομοιάσουμε με τον προσωπικό υπολογιστή. Ήταν μια μηχανή αποθήκευσης δεδομένων η οποία είχε πολλές επιπλέον δυνατότητες όπως γρήγορη αναζήτηση

δεδομένων, απομνημόνευση παρόμοιων αναζητήσεων, και πληθώρα πληροφοριών για όλους τους τομείς. Στη συνέχεια το 1960 η ιδέα χρήσης υπολογιστών για συνεργασία μεταξύ των ανθρώπων εμφανίστηκε ξανά: οι ΗΠΑ(1958) ίδρυσαν την Advanced Research Projects Agency (ARPA), η οποία σε 8 μήνες από την ίδρυσή της ανέπτυξε τον πρώτο επιτυχή δορυφόρο. Το 1962 ο Dr. J.C.R. Licklider διορίστηκε για να διευθύνει την ARPA. Άλλαξε την ARPA έτσι ώστε να προσφέρει περισσότερα ερευνητικά προνόμια στα πανεπιστήμια. Στην ουσία, εξαιτίας των προσπαθειών του έγινε έρευνα με πολλούς και σημαντικούς καρπούς. Ήταν αυτή η έρευνα που οδήγησε τελικά στο ARPANET, στα εμπορικά συστήματα καταμερισμού χρόνου, και τελικά στο Διαδίκτυο.

Το Digital Media Laboratory ορίζει το CSCW ως: *“ένας διεπιστημονικός ερευνητικός τομέας συμπεριλαμβανομένης της πληροφορικής, των οικονομικών, της κοινωνιολογίας, και της ψυχολογίας. Η έρευνα του CSCW εστιάζει στην ανάπτυξη των νέων θεωριών και των τεχνολογιών για το συντονισμό των ομάδων ανθρώπων που εργάζονται από κοινού”*.

Ενώ ο όρος "groupware" έχανε σιγά σιγά την έννοιά του, μια νέα φράση, “social software” άρχισε να γίνεται όλο και πιο γνωστή. Εντούτοις, για τα πρώτα 15 έτη της ύπαρξής του, περισσότερο στη δεκαετία του '90 ,ο όρος χρησιμοποιήθηκε σπάνια έξω από τις πολύ ειδικευμένες ομάδες.

Ψάχνοντας με βάση την ημερομηνία, η πιο πρόωρη αναφορά στην οποία μπορούμε να βρούμε τον όρο “social software” είναι μια δημοσίευση το 1990 στην οποία συνδέεται με το ανοικτό υπερκείμενο (open hypertext). Η επόμενη αναφορά του social software είναι το 1992 στο οποίο το Xanadu του TED Nelson και το AMIX του Phil Salin καλούνται social software.

Από τα τέλη του 2002 ο όρος “social software” μπήκε στην πιο κοινή χρήση, πιθανώς εξαιτίας των προσπαθειών του Clay Shirky που οργάνωσε το “Social Software Summit” το Νοέμβριο του 2002.

Όταν ο Clay ρωτήθηκε αν ήταν η απώλεια νοήματος του όρου “groupware” που τον έκανε να διαλέξει τον όρο “social software” αυτός απάντησε ότι έψαχνε για κάτι το οποίο να συγκεντρώνει όλες τις χρήσεις του λογισμικού που υποστηρίζει τις αλληλοεπιδρώμενες ομάδες, ακόμα και όταν η αλληλεπίδραση ήταν offline. Ο όρος “groupware”, ήταν για αυτόν η προφανής επιλογή αλλά είχε μολυνθεί από την επιχειρηματική groupware εργασία. Ένας παλιός ορισμός του Clay για τον όρο social software ήταν: *“Το social software μεταχειρίζεται τις ομάδες ως πρώτης τάξεως*

αντικείμενα στο σύστημα”. Ωστόσο, ο Clay πιο πρόσφατα προτιμά το απλούστερο: «λογισμικό που υποστηρίζει την αλληλεπίδραση ομάδας»

Ο ορισμός της wikipedia(14/4/07) είναι ο εξής: “*To social software επιτρέπει στους ανθρώπους να συναντιούνται, να συνδέονται ή να συνεργάζονται μέσω της επικοινωνίας μέσω υπολογιστή και να σχηματίσουν online κοινότητες.*”

2.3 Διαφορά κοινωνικού λογισμικού από το “συμβατικό”

Είναι σημαντικό να τονίσουμε τη διαφορά του κοινωνικού λογισμικού σε σχέση με το λογισμικό που απευθύνεται σε ένα χρήστη. Στη δεύτερη κατηγορία, ένας μόνο χρήστης μπορεί να έχει εμπειρία αλληλεπίδρασης με την εφαρμογή κατά τη χρησιμοποίησή της. Σίγουρα η εφαρμογή μπορεί να χρησιμοποιηθεί από περισσότερους χρήστες αλλά αυτό συμβαίνει ξεχωριστά στον υπολογιστή του καθενός χωρίς να υπάρχει η δυνατότητα να μοιραστούν κοινές εμπειρίες, γεγονός το οποίο συμβαίνει με το social software. Κάποιος θα μπορούσε να ισχυριστεί ότι ο όρος κοινωνικό (social) θα ήταν προτιμότερο να αντικατασταθεί από όρους όπως ομάδα (group) ή κοινότητα (community). Όμως αντικαθιστώντας τον όρο αυτό θα χανόταν η ουσία της κοινής διαμοιραζόμενης εμπειρίας η οποία συμβαίνει πέρα από μία απλή ομάδα ή μία απλή κοινότητα.

2.4 Ευκολίες που παρέχει το κοινωνικό λογισμικό στο Διαδίκτυο

Το social software έχει κατακλύσει το Διαδίκτυο περιλαμβάνοντας πλέον εργαλεία τα οποία βοηθούν την επικοινωνία ομάδων εργασίας όσο και εργαλεία τα οποία έχουν ως στόχο την επικοινωνία ανθρώπων από οποιοδήποτε μέρος του πλανήτη για ψυχαγωγικούς λόγους(ακόμα και οι τρισδιάστατοι κόσμοι των ηλεκτρονικών παιχνιδιών στο Διαδίκτυο ανήκουν στο social software). Πριν από το Διαδίκτυο, εφευρέσεις όπως το τηλέφωνο και ο τηλεγράφος επέτρεπαν την επικοινωνία μεταξύ ανθρώπων έχοντας όμως σοβαρά μειονεκτήματα, όπως τον αποκλεισμό της επικοινωνίας μεταξύ μεγάλων ομάδων, την απουσία πραγματικού χρόνου επικοινωνίας και το μεγάλο κόστος. Αρκεί να σκεφτούμε τη δυσκολία που παρουσιάζει η συγκέντρωση μιας μεγάλης ομάδας για μία διάσκεψη. Η εύρεση του κατάλληλου χώρου και του κατάλληλου χρόνου ώστε να βολεύει όλα τα μέλη που

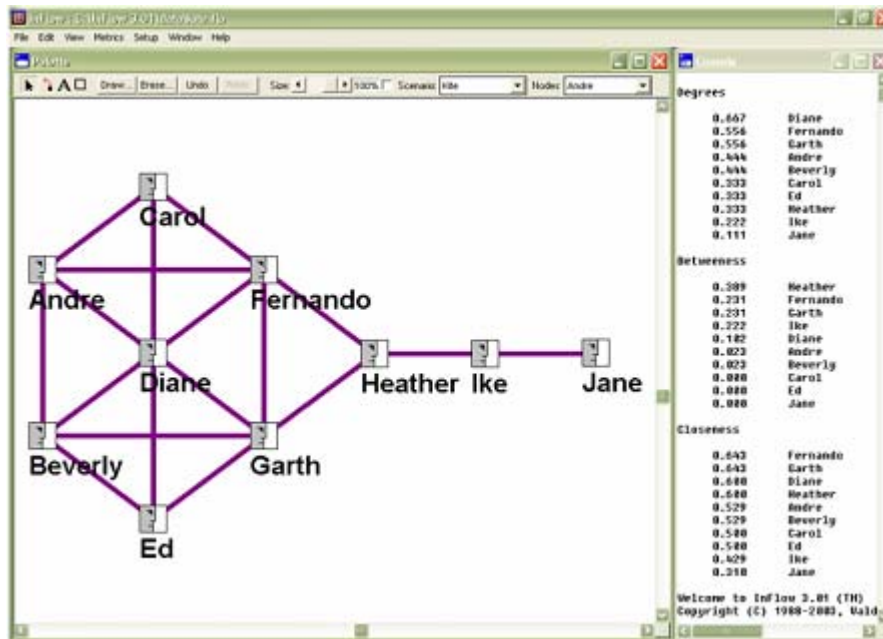
αποτελούν την ομάδα συνθέτουν τα δύο σημαντικότερα ζητήματα που πρέπει να αντιμετωπιστούν. Από την άλλη πλευρά, τα κοινωνικά εργαλεία του Διαδικτύου, με την ευκολία που έχουν στη χρήση (απλά αν σκεφτούμε πόσο εύκολο είναι να χρησιμοποιήσει κανείς ένα πρόγραμμα για chat) και την αποφυγή της αναγκαιότητας του πραγματικού χρόνου επιτυγχάνεται η υπερπήδηση όλων αυτών των περιορισμών που υπήρχαν.

Ενώ αρχικά το ενδιαφέρον των μεγάλων φορέων παροχής υπηρεσιών μέσω του Διαδικτύου(όπως π.χ. η Google) είχε στραφεί στις δραστηριότητες και τα ενδιαφέροντα της αναπτυξιακής κοινότητας (developers), πλέον όλη η προσοχή είναι στραμμένη στην ανάπτυξη υπηρεσιών με «κοινωνικό» περιεχόμενο μέσω της αλληλεπίδρασης των απλών χρηστών. Το ερώτημα που τίθεται είναι γιατί τα τελευταία χρόνια γίνεται τόσο ευρέως διαδεδομένη η χρήση social software. Την απάντηση σε αυτό το ερώτημα μπορούμε να συνοψίσουμε αναφέροντας τη ραγδαία τεχνολογική εξέλιξη και τα ποσά που επενδύονται για την ανάπτυξη τέτοιων ειδών λογισμικού. Επιπλέον όμως μπορούμε να αναφερθούμε στα εξής: Η διαθεσιμότητα εργαλείων χαμηλού κόστους και υψηλού εύρους ζώνης, όπως είναι τα blogs ή η πληθώρα χρηστών (η οποία πλέον μετριέται σε εκατομμύρια) που όντας κοινωνικοί και παρακινημένοι από τη μεγάλη αναφορά στο κοινωνικό λογισμικό, ακολουθούν τις εξελίξεις. Όλα τα παραπάνω οδηγούν στο συμπέρασμα ότι το social software θα αποτελέσει το επίκεντρο του ενδιαφέροντος στο κοντινό μέλλον.

2.5 Είδη δικτύων που χρησιμοποιούν κοινωνικό λογισμικό

Τα κοινωνικά δίκτυα (social networks) είναι κάποια από τα είδη δικτύων τα οποία χρησιμοποιούν κοινωνικό λογισμικό. Τα τελευταία χρόνια τα δίκτυα αυτά έχουν κερδίσει κατά πολύ το ενδιαφέρον των χρηστών και είναι ίσως τα δίκτυα τα οποία σκέφτεται κανείς όταν κάποιος αναφέρεται στο social software. Το μέγιστο μέγεθος ενός τέτοιου δικτύου (σε χρήστες) φτάνει τα 150 άτομα ενώ ο μέσος όρος βρίσκεται στα 124 άτομα. Στις κοινότητες αυτές, ένας αρχικός αριθμός ιδρυτικών μελών στέλνει μηνύματα προσκαλώντας μέλη των προσωπικών τους δικτύων να γίνουν μέλη του ιστοχώρου. Στη συνέχεια επαναλαμβάνεται η ίδια διαδικασία από τα νέα μέλη και έτσι σε σύντομο χρονικό διάστημα αυξάνεται ο αριθμός των μελών στο δίκτυο. Το κάθε μέλος περιγράφεται από το προσωπικό του προφίλ ενώ τα δεδομένα των χρηστών που περιγράφουν το προφίλ τους βρίσκονται αποθηκευμένα σε μία

κεντρική βάση δεδομένων. Σε αυτά οι χρήστες μπορούν να αναπτύξουν και να διατηρήσουν ένα δίκτυο φίλων ή συνεργατών για κοινωνικούς ή για επιχειρησιακούς λόγους αντίστοιχα. Ο χρήστης μπορεί λοιπόν να κρατάει επαφή με φίλους, να βρει κάποιον σύντροφο, να ψάξει για εργασία, να ανταλλάξει δεδομένα.



Εικόνα 3

Οι κοινότητες αυτές περιγράφονται με κόμβους και συνδέσμους μεταξύ των κόμβων (όπως παρουσιάζονται και στην εικόνα 3). Οι κόμβοι αντιστοιχούν στους χρήστες-μέλη και οι σύνδεσμοι παρουσιάζουν τις σχέσεις μεταξύ τους. Η μορφή που παρουσιάζουν τα δίκτυα αυτά καθορίζει τη χρησιμότητα που έχει το δίκτυο στα μέλη του. Γενικά, τα περισσότερο ελεύθερα δίκτυα, δηλαδή αυτά που χαρακτηρίζονται από πολλές κοινωνικές σχέσεις μεταξύ των μελών αλλά και με χρήστες από άλλες κοινότητες, σε σχέση με τα περισσότερο κλειστά, προσφέρουν μεγαλύτερο αριθμό ευκαιριών και πιο πολλές ιδέες στα μέλη τους. Αυτό απορρέει από το γεγονός ότι οι χρήστες που είναι μέλη σε παραπάνω από μία κοινότητες, είναι πιο πιθανό να έχουν πρόσβαση σε μεγαλύτερο όγκο πληροφοριών. Επιπλέον οι χρήστες αυτοί μπορούν να λειτουργήσουν κατά κάποιο τρόπο ως μεσίτες, ενώνοντας δύο δίκτυα τα οποία δε συνδέονται με κάποιο άμεσο τρόπο².

Μία ακόμα κατηγορία δικτύων που βασίζονται στο social software είναι τα προσωπικά δίκτυα (personal networks). Σε αυτά, προσωπικά δεδομένα

² Χαρακτηριστικές ιστοσελίδες είναι οι: <http://www.classmates.com> και <http://www.friendster.com>.

ανταλλάσσονται μεταξύ συγκεκριμένων private ομάδων μέσω εφαρμογών που χρησιμοποιούνται από αυτές. Με τη χρήση κοινών format για τα μεταδεδομένα (π.χ. RDF, XMP) και πρωτόκολλα δικτύου μεταφοράς δεδομένων (π.χ. Rendevouz) οι εφαρμογές διαμοιράζονται μεταξύ των χρηστών που αποτελούν την ιδιωτική ομάδα³.



Εικόνα 4

Τέλος, υπάρχουν τα δημόσια δίκτυα μεταφοράς δεδομένων (public sharing networks) στα οποία δίνεται η δυνατότητα σε μέλη να συμμετέχουν σε διαδικασίες όπως η δημιουργία, η δημοσίευση και ο σχολιασμός δεδομένων και πληροφοριών. Με την ύπαρξη μάλιστα πληθώρας ψηφιακών καταστημάτων αγορών ενδυναμώνεται ο ρόλος του χρήστη καθώς μπορεί να σχολιάσει την ποιότητα των προϊόντων ενισχύοντας ή όχι αυτό που λέγεται ψηφιακή φήμη (digital reputation). Ως αποτέλεσμα, σκοπός των καταστημάτων αυτών είναι να εξασφαλίσουν μια καλή φήμη έχοντας έτσι καλύτερες πωλήσεις. Να αναφέρουμε επίσης ότι στην κατηγορία αυτή ανήκουν τα wikis και τα web logs στα οποία οι χρήστες λειτουργούν με το διπλό ρόλο του αναγνώστη και του συγγραφέα. Υπάρχουν μηχανισμοί οι οποίοι ενισχύουν τη διασύνδεση μεταξύ αυτών των ιστοχώρων. Για παράδειγμα ένα blog μπορεί να γνωρίζει ότι κάποιος άλλος έχει σχολιάσει το άρθρο του. Τα wikis προσφέρουν τη δυνατότητα εγγραφής και διόρθωσης άρθρων από πολλούς χρήστες.

2.6 Εφαρμογές κοινωνικού λογισμικού

Στο social software ανήκει το Instant Messaging (IM), το internet Relay Chat (IRC) και τα Forums. Πριν την εμφάνιση του Instant Messaging ευρέως διαδεδομένη ήταν η χρήση του e-mail στο Διαδίκτυο. Το e-mail ανήκει στην κατηγορία του social

³ Χαρακτηριστική ιστοσελίδα είναι η: <http://www.apple.com/ilife>.

software. Όμως η επικοινωνία αυτή δεν είναι ζωντανή με αποτέλεσμα η απάντηση του παραλήπτη να μην είναι άμεση. Επιπλέον, οι διαδικασίες για την ανάγνωση, την απάντηση και την αποστολή του e-mail απαιτούν μία ακολουθία από “κλικς”, γεγονός που επιβραδύνει ακόμα περισσότερο τη διαδικασία. Οι παραπάνω λόγοι λοιπόν συνέβαλαν στην ανάπτυξη της δημοτικότητας του IM. Το IM επιτρέπει στο χρήστη να διατηρεί έναν κατάλογο ανθρώπων με τους οποίους επιθυμεί να έρχεται σε επικοινωνία. Ο χρήστης έχει τη δυνατότητα να στείλει μήνυμα σε όλες τις επαφές του και να συνομιλήσει ιδιωτικά με οποιονδήποτε από αυτούς επιθυμεί, με την προϋπόθεση ότι βρίσκονται και οι δύο συνδεδεμένοι στο δίκτυο. Παραδείγματα τέτοιων εφαρμογών είναι:

- MSN Messenger
- AOL Instant Messenger(βλέπε εικόνα 5)



Εικόνα 5

Το Internet Relay Chat (IRC) είναι ο πλέον ευρέως διαδεδομένος τρόπος επικοινωνίας μέσω του Διαδικτύου. Η ιδέα ξεκίνησε από τον Jarkko Oikarinen το 1988 στη Φινλανδία και σταδιακά εξαπλώθηκε σε όλο και περισσότερες χώρες με τελικό αποτέλεσμα την παγκόσμια πλέον χρήση του. Αυτό που κάνει είναι να επιτρέπει στους χρήστες να συνδέονται σε δωμάτια συζητήσεων (chat rooms) ενισχύοντας την δημόσια (ή ιδιωτική) επικοινωνία πολλών ατόμων ταυτόχρονα. Τα δωμάτια χωρίζονται σε θεματικές ενότητες και κάθε χρήστης συνδέεται σε αυτά ανάλογα με τα ενδιαφέροντά του. Μπορεί επίσης να δημιουργήσει ένα προσωπικό δωμάτιο και να προσκαλέσει άτομα σε αυτό. Χαρακτηριστική εφαρμογή είναι το mIRC (βλέπε εικόνα 6).

τελευταία δέκα. Με την εμφάνιση του Διαδικτύου εμφανίστηκε και το αντίστοιχο social software ώστε να καλύψει τις ανάγκες των χρηστών. Βέβαια, με την πάροδο του χρόνου έγινε κατανοητό ότι το αρχικό λογισμικό που σχεδιάστηκε δε θα ήταν δυνατόν να καλύπτει τις ανάγκες των χρηστών του παγκόσμιου ιστού καθώς ο αριθμός αυτών αυξανόταν με ραγδαίους ρυθμούς. Αρχικά εμφανίζονταν καινούργιες εφαρμογές οι οποίες προσέφεραν στους χρήστες μεγάλες ευκολίες, εργαλεία και καινοτομίες. Έτσι, ένα μεγάλο πλήθος χρηστών άρχιζαν να χρησιμοποιούν τις κοινωνικές αυτές εφαρμογές αξιοποιώντας τις συναρπαστικές νέες δυνατότητες. Οι αρχικοί στόχοι όμως με το πέρασμα του χρόνου άρχιζαν να παραμερίζονται. Αυτό οφείλεται στο γεγονός ότι οι χρήστες ασχολούνταν με θέματα ασήμαντα ή άσχετα σε σχέση με τους αρχικούς στόχους, με αποτέλεσμα να χάνεται η ουσία της εφαρμογής. Στο σημείο αυτό ήταν διαθέσιμες δύο “σκληρές” επιλογές: είτε η αυτοκαταστροφή της εφαρμογής, είτε η θέση κάποιων φραγμών για το ποιος μπορεί να ανήκει στην ομάδα και να χρησιμοποιεί την εφαρμογή. Έτσι λοιπόν, αρχικές υποθέσεις οι οποίες είχαν γίνει για τις ομάδες χρηστών όπως το άπειρο μέγεθος σε άτομα μιας ομάδας, ή η δυνατότητα σε οποιονδήποτε να γίνει μέλος και η απεριόριστη ελευθερία ενός ατόμου της ομάδας, τελείωσαν. Γεγονός και κανόνα αποτελούν για πολλές ομάδες η απαίτηση κάποιων χαρακτηριστικών ώστε να γίνει κάποιος μέλος και όταν πλέον γίνει του τίθενται περιορισμοί στα δικαιώματά του. Βλέπουμε λοιπόν τη δυσκολία που παρουσιάζει η σχεδίαση λογισμικού για ομάδες λόγω των αυξανόμενων απαιτήσεων και περιορισμών που προκύπτουν. Αυτό έχει ως συνέπεια την εξειδίκευση των σχεδιαστών κοινωνικού λογισμικού και τη διαφοροποίησή τους από σχεδιαστές λογισμικού το οποίο απευθύνεται σε ένα χρήστη. Η αλληλεπίδραση μεταξύ των χρηστών του λογισμικού αυτού αποτελεί την κύρια διαφορά και την αιτία του παραπάνω γεγονότος.

2.8 Πληροφορίες για τη σχεδίαση

Η προσοχή των σχεδιαστών κοινωνικού λογισμικού έχει στραφεί στη συγκέντρωση πληροφοριών που θα δώσουν λύσεις στο πώς να βελτιώσουν ή να δημιουργήσουν λογισμικό το οποίο θα βοηθά τα μέλη ώστε να συνεργάζονται και να επικοινωνούν καλύτερα. Αντλούνται πληροφορίες από μεμονωμένα άτομα για το πώς θα πρέπει να λειτουργεί και τι θα πρέπει να περιλαμβάνει το λογισμικό, θεωρώντας τη γνώμη του μέλους ως τον πιο σημαντικό παράγοντα. Έτσι, ακόμα και αν το λογισμικό προορίζεται για χρήση από μια ομάδα, σημαντικές αποφάσεις παίρνονται βασισμένες

στια απόψεις των μελών. Η γενική ιδέα συνοψίζεται στο εξής: η επιτυχημένη εφαρμογή κοινωνικού λογισμικού είναι αυτή που ικανοποιεί τους προσωπικούς στόχους των μεμονωμένων χρηστών ενώ ταυτόχρονα προωθεί και τους στόχους της ομάδας στην οποία ανήκουν οι χρήστες. Γεγονός πάντως αποτελεί ότι όντας κάποιος μέλος σε ένα group επηρεάζεται συνειδητά ή ασυνείδητα από τη γνώμη των υπολοίπων μελών ή τη γενική τάση η οποία επικρατεί σε θέματα αποφάσεων, προτιμήσεων κλπ..

2.9 Θετικά και αρνητικά για το χρήστη και την εφαρμογή

Ένας συνεχώς αυξανόμενος αριθμός εφαρμογών οι οποίες χαρακτηρίζονται κοινωνικές, όπως για παράδειγμα οι del.icio.us και Flickr, εμφανίζονται με την αρχή να προσφέρουν υπηρεσίες στους χρήστες προμηθευοντάς τους με τα κατάλληλα εργαλεία για εργασίες όπως επερωτήσεις, ανταλλαγή δεδομένων και τη δυνατότητα να μπορούν να επεμβαίνουν και να διαχειρίζονται τα δεδομένα της εφαρμογής. Έτσι λοιπόν, τέτοιες διεπαφές παρουσιάζονται στο χρήστη άκρως ελκυστικές καθώς αφαιρείται η περίπτωση των στατικών δεδομένων και παράλληλα ενισχύεται το κέρδος του χρήστη από τα δεδομένα τα οποία μπορεί να αξιοποιήσει. Από την άλλη πλευρά, η εφαρμογή (ή εταιρεία) μπορεί να έχει είτε κέρδος, είτε να δυσφημιστεί. Κέρδος με την έννοια ότι αποκτά δημοσιότητα με τους ικανοποιημένους χρήστες οι οποίοι αυξάνουν την ελκυστικότητα και τη διαφάνεια της, για παράδειγμα αναφέροντάς την στην προσωπική τους ιστοσελίδα ή blog. Με τον τρόπο αυτό δε χρειάζεται να ξοδεύουν χρήματα για διαφημίσεις, για την πληρωμή μιας εταιρίας δημοσίων σχέσεων, ή για έρευνα, παρά να συνεργαστούν με χρήστες οι οποίοι θα λειτουργήσουν το ίδιο καλά για την προώθηση(λόγω του ότι είναι ευχαριστημένοι) αλλά και για θέματα έρευνας για τη βελτίωση των προϊόντων. Δυσφήμιση μπορεί να προκύψει από δυσαρεστημένους πελάτες-χρήστες οι οποίοι για παράδειγμα σχολιάζουν αρνητικά κάποιο προϊόν.

Οι χρήστες του Διαδικτύου βέβαια αρχίζουν και συνειδητοποιούν ολοένα και περισσότερο τη συμμετοχή τους και τη συνεισφορά τους. Για παράδειγμα με το να βαθμολογήσουν την ποιότητα ενός προϊόντος κατανοούν ότι πλέον αποτελούν σημαντικό παράγοντα για την επιτυχία των επιχειρήσεων αυτών. Η παροχή αυτών των δωρεάν πληροφοριών απαιτεί χρόνο για το χρήστη και μάλιστα έχει προκληθεί ανησυχία σε περίπτωση όπου αφαιρεθούν όλες οι υποβολές χρηστών οι οποίες

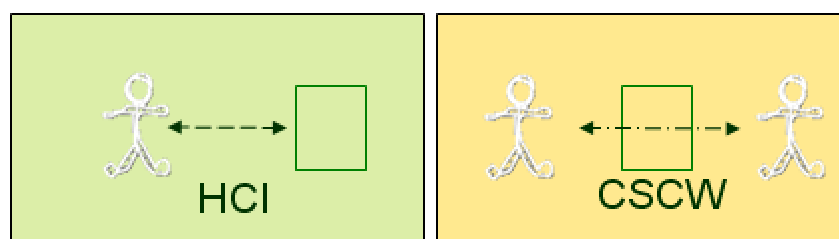
αναφέρονται σε θέματα κάτω από εμπορική άδεια. Επιπλέον προβληματίζονται λόγω του κατακλυσμού εφαρμογών και υπηρεσιών οι οποίες προσφέρουν παρόμοια ή συμπληρωματική λειτουργία. Με την ενασχόλησή τους με μία από αυτές ξοδεύουν αρκετό χρόνο και κόπο για να περάσουν σε μία άλλη η οποία μπορεί να έχει κάποιο επιπλέον χαρακτηριστικό. Μάλιστα, καθώς οι περισσότερες υπηρεσίες απαγορεύουν τη μεταφορά δεδομένων που έχουν καταβληθεί από το χρήστη, σε κάποια άλλη υπηρεσία, καταλαβαίνουμε πως ουσιαστικά ο χρήστης στη νέα υπηρεσία θα ξεκινήσει πάλι από την αρχή. Η ανάγκη για ελευθερία των δεδομένων μεταφράζεται στην πίεση που ασκείται από τους χρήστες ώστε οι ιστοχώροι να παρέχουν δωρεάν υπηρεσίες σε αυτούς. Καθώς τα εργαλεία και οι υπηρεσίες δεν είναι από μόνα τους αρκετά, οι πιέσεις αυτές έχουν επίσης ως αποτέλεσμα την τυποποίηση του format των δεδομένων, ώστε να είναι ευκολότερα στη μεταφορά και στην ανταλλαγή. Δύο βασικές τεχνολογίες οι οποίες διευκολύνουν την ανταλλαγή αυτή είναι η XML⁵ και το RDF⁶, μαζί με συγκεκριμένα λεξιλόγια όπως το FOAF (χρησιμοποιείται για την περιγραφή των χρηστών) και το RSS (για τα περιεχόμενα), που χρησιμοποιούνται ανάλογα με τις εκάστοτε ανάγκες.

⁵ Βλέπε: <http://www.w3c.org/xml>

⁶ Βλέπε: <http://www.w3c.org/rdf>

3 Εισαγωγή στις έννοιες των συνεργατικών συστημάτων

Επιχειρώντας μια βαθύτερη προσέγγιση των γενικών εισαγωγικών θεωρητικών αναλύσεων, ερμηνειών και όρων είναι χρήσιμο να οριοθετήσουμε και να αποσαφηνίσουμε τα όρια-σύνορα μεταξύ αυτού που λέμε “αλληλεπίδραση χρήστη-υπολογιστή” και “αλληλεπίδραση υποστηριζόμενη από υπολογιστή”(όπως βλέπουμε και στην εικόνα 7).



Εικόνα 7

Όπως λοιπόν σε κάθε επιστήμη έτσι και στην επιστήμη των υπολογιστών παρατηρούμε αυτό που ονομάζουν οι της “παιδαγωγικής επιστήμης” “διεπιστημονικότητα” ενός γνωστικού αντικείμενου, που σημαίνει το βαθμό στον οποίο εμπλέκονται σε ένα γνωστικό αντικείμενο, αντικείμενα άλλων επιστημών. Επιχειρώντας λοιπόν τη διεπιστημονική απόπλεξη των δύο αυτών τομέων θα λέγαμε ότι όσον αφορά την επικοινωνία ανθρώπου-υπολογιστή τα γνωστικά αντικείμενα που κατά παράδοση εμπλέκονται είναι αυτά της ψυχολογίας, του χειρισμού γραφικών αντικειμένων τόσο σε υψηλό όσο και σε χαμηλότερο επίπεδο: toolkit expansion και component augmentation Gnostic areas αντίστοιχα, καθώς επίσης και του τομέα των interactive-devices, όπου μιλάμε για θέματα-τεχνικές αλληλεπίδρασης όπως gesture recognition (digital-glove), eye catching κ.α.. Στο CSCW αντίστοιχα εμπλέκονται επιστήμες όπως αυτή της κοινωνιολογίας (μέσω των σταδίων⁷ π.χ. από τα οποία περνάει μια ηλεκτρονική κοινότητα πρακτικής όπως αυτή περιγράφεται & στο άρθρο: “Experience-based social and collaborative performance in an ‘electronic village’ of local interest”⁸), των δικτύων, των καταναμημένων συστημάτων ενώ παράλληλα παίζουν ρόλο οι φορητές συσκευές ως συστήματα αλληλεπίδρασης που κανείς δεν μπορεί να παραβλέψει, όπως και το έντονο ερευνητικό ενδιαφέρον δείχνει,

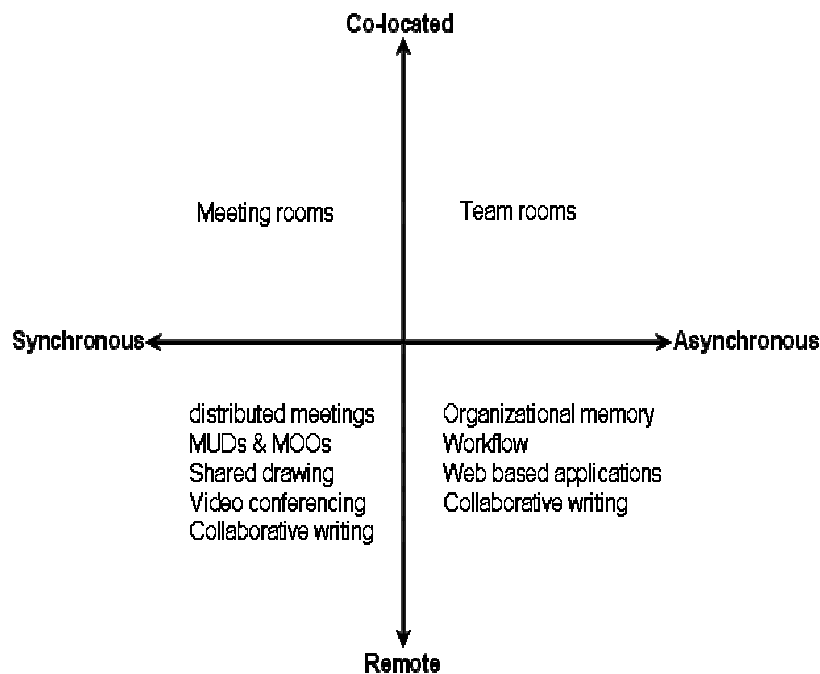
⁷ Βλέπε: Bruce Tuckman:1965, forming, storming, norming, performing κλπ.

⁸ Βλέπε: Akoumianakis D., Vidakis N., Vellis G., Milolidakis G., Kotsalis D. (2007). Experience-based social and collaborative performance in an ‘electronic village’ of local interest: The eKoNES framework, In Proceedings of the 9th International Conference on Enterprise Information Systems (ICEIS’2007), 12-16 June, Funchal, Madeira – Portugal.

προς την “επιστήμη” των φορητών συσκευών(PDA’s, mobile phones) και ως προς τις δυνατότητες που δύνανται να προσφέρουν, αλλά και της ανάπτυξης εξειδικευμένου λογισμικού σε αυτές.

Περνώντας σε μια πιο ειδική τοποθέτηση θα λέγαμε ότι η τεχνολογία συνεργασίας (Collaborative technology) αφορά τη μελέτη, τη σχεδίαση, την ανάπτυξη και τη χρήση υπολογιστικών συστημάτων που επιτρέπουν τη συνεργασία ομάδων ανθρώπων εκ του σύνεγγυς ή εξαποστάσεως με σύγχρονο ή ασύγχρονο τρόπο.

Χωροχρονικά λοιπόν θα μπορούσαμε να κατατάξουμε τα συστήματα συνεργασίας (CoSystems) όπως παρατηρούμε και στα παρακάτω ισοδύναμα διαγράμματα:



Εικόνα 8

σε τέσσερις κατηγορίες ανάλογα με το αν είναι: *σύγχρονα και στον ίδιο χώρο*, όπως είναι τα meeting rooms, ή η face του face συνομιλία, μια άλλη κατηγορία είναι η *σύγχρονη και απομακρυσμένη*, όπως το video conferencing, το shared drawing, το collaborative writing. Επόμενη κατηγορία είναι συστήματα τα οποία είναι ασύγχρονα και στον ίδιο χώρο, όπως post-it notes όπως παρατηρούμε και στην εικόνα 8, ενώ τέλος υπάρχουν τα συστήματα τα οποία είναι ασύγχρονα αλλά χωρικά

διεσπαρμένα ή αλλιώς κατανεμημένα, όπως το γνωστό e-mail, το fax και πολλά άλλα.

Θέμα αυτής της πτυχιακής εργασίας είναι η σύγχρονη και απομακρυσμένη επικοινωνία μέσω της υλοποίησης ενός Collaborative Framework το οποίο θα αποτελέσει κινητήριο μοχλό, πυρήνα της, αυτού που ονομάζουμε Computer Supported Cooperative Work.

3.1 Model View Controller και Συνεργατικά συστήματα

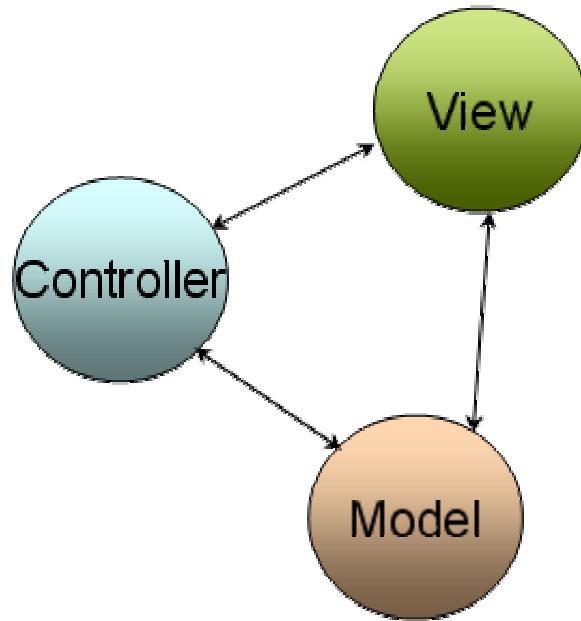
3.1.1 Τι είναι η Model View Controller Αρχιτεκτονική?

Το model view controller, είναι ουσιαστικά ένα πρότυπο που πρωτοεισήχθη αρχικά με τη γλώσσα προγραμματισμού Smalltalk και που ορίζει τον τρόπο δόμησης της αρχιτεκτονικής ενός συστήματος ως άθροισμα τριών διακριτών στοιχείων-μερών: του μοντέλου, της όψης και του ελεγκτή. Το μοντέλο είναι αυτό που κρατάει την πληροφορία που παρουσιάζεται στο χρήστη μέσω του view, του οποίου ρόλος είναι απλά να την εμφανίσει. Τέλος ο ρόλος του controller είναι να συντονίσει τα δύο αυτά μέρη και να δράσει ως ένα επίπεδο ανταλλαγής μηνυμάτων μεταξύ των, έχοντας ουσιαστικά ρόλο συντονιστή.

3.1.2 Το Model View Controller στην πράξη

Η σημασία και η βοήθεια αυτού του τρόπου δόμησης των συστημάτων με την πάροδο των χρόνων, απεδείχθη τόσο σημαντική που έχει φτάσει πλέον σε σημείο να θεωρείται καθιερωμένη και απαραίτητη, είτε αυτό αφορά GUI-toolkit architecture (βλέπε AWT, Swing: UI Delegate) είτε αφορά enterprise συστήματα⁹, όπου τα entity beans παίζουν το ρόλο του μοντέλου-persistent storage με τη βοήθεια του ORM (Object Relational Mapping, τα jsp's παίζουν ο ρόλο του view ενώ τα servlets το ρόλο του ελεγκτή και συντονιστή των παραπάνω) ή ακόμα και σε επίπεδο υλοποίησης μιας κλασικής desktop εφαρμογής με τη χρήση π.χ. του jgoodies binding API.

⁹ Λέγοντας Enterprise συστήματα εννοούμε large scale συνήθως, 3-tiered web-based εφαρμογές. Βλ. <http://java.sun.com/blueprints/patterns/MVC.html>



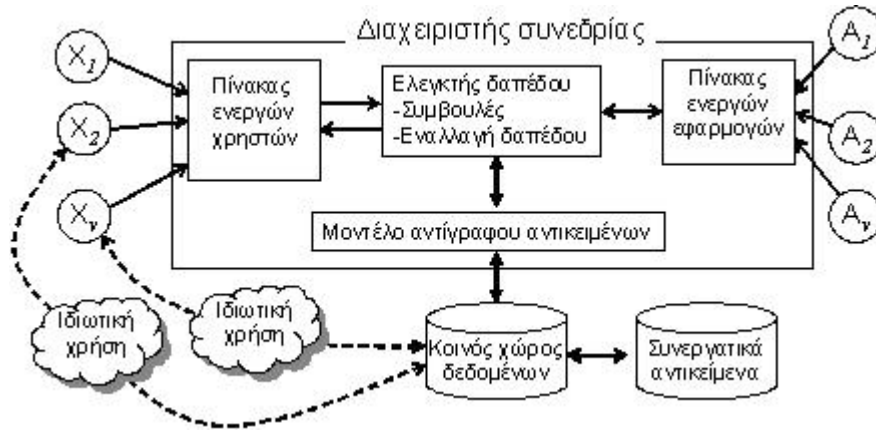
Εικόνα 9 Model View Controller

Ο χώρος των Collaborative συστημάτων είναι άρρηκτα συνδεδεμένος με αυτή τη λογική, με το μοντέλο, τον controller και το view να είναι distributed, ανάλογα με την αρχιτεκτονική θεώρηση που μελετάμε.

3.2 Μηχανισμοί Σύγχρονης Συνεργασίας

Οι μηχανισμοί εκτέλεσης συνεργατικών καθηκόντων αφορούν τόσο την ασύγχρονη συνεργασία μεταξύ των εταίρων όπου η έμφαση είναι στο διαμοιρασμό κωδικοποιημένης γνώσης (knowledge sharing), όσο και στη συγχρονισμένη συνεργασία όπου εμπλέκονται θέματα που αφορούν την οργάνωση του συνεργατικού καθήκοντος, την υποστήριξη συνεργατικών αντικειμένων, τις πολιτικές συνεργασίας & πρόσβασης, τον έλεγχο δαπέδου κλπ. Στο πλαίσιο αυτό αξίζει επίσης να γίνει διαχωρισμός μεταξύ συνεργατικών και μη-συνεργατικών εφαρμογών. Στην κατηγορία των συνεργατικών εφαρμογών περιλαμβάνονται οι εφαρμογές ή οι υπηρεσίες μιας κοινότητας που είναι σχεδιασμένες εξ αρχής για ομαδική χρήση (π.χ. συνεργατικά ημερολόγια, εργαλεία συζήτησης, κλπ), ενώ μη-συνεργατικές εφαρμογές θεωρούνται τα εργαλεία και οι εφαρμογές που χρησιμοποιούνται από μεμονωμένα μέλη για την εκτέλεση ατομικών καθηκόντων (π.χ. επεξεργαστής κειμένου, ηλεκτρονικό ταχυδρομείο, κλπ). Τα παραπάνω επηρεάζουν και καθορίζουν τεχνικά θέματα του περιβάλλοντος που υποστηρίζει την συνεργασία όπως το μοντέλο

διαχείρισης αντιγράφων αντικειμένων (object replication model) αλλά και τους κοινοτικούς μηχανισμούς αλληλεπίδρασης (π.χ. ειδοποίηση, επίγνωση, ανάδραση).



Εικόνα 10

3.3 Διαχείριση συνεδρίας (Session Management)

Η συνεδρία είναι το υψηλότερο λειτουργικό αντικείμενο συμπερίληψης όπου εκτελούνται όλα τα συνεργατικά καθήκοντα. Η διαχείριση συνεδρίας αφορά τον τρόπο με τον οποίο συνεργαζόμενοι χρήστες συμμετέχουν σε μια συνεδρία. Ο διαχειριστής συνεδρίας αναλαμβάνει την δημιουργία και διαγραφή μια συνεδρίας (παράσταση αντικειμένου τύπου 'συνεδρία'), την συμμετοχή σε μια εν' ενεργεία συνεδρία, την αποχώρηση από μια εν' ενεργεία συνεδρία, την αλλαγή παραμέτρων μιας συνεδρίας, την επισκόπηση ενεργών συνεδριών, την διατύπωση ερωτήσεων προς τους συμμετέχοντες σε μια συνεδρία. Για τους παραπάνω σκοπούς ο διαχειριστής συνεδρίας διατηρεί ένα πίνακα των ενεργών συνεδριών (Sessions) τον οποίο και ενημερώνει ανάλογα.

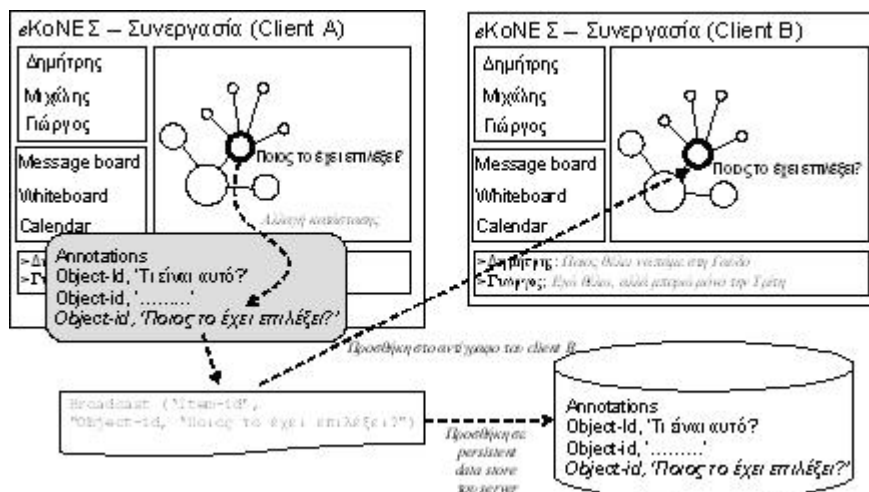
3.4 Διαχείριση Δαπέδου (Floor Control)

Η διαχείριση δαπέδου επιτρέπει χρήστες συνεργατικών εφαρμογών να χρησιμοποιούν και να διαμοιράζονται πόρους, χωρίς διαμάχες πρόσβασης. Ο διαχειριστής δαπέδου είναι υπεύθυνος για την παραχώρηση ή άρνηση παραχώρησης δαπέδου ανάλογα με την ενεργοποιημένη πολιτική δαπέδου, καταγραφή της κατάστασης των διαμοιρασμένων πόρων, αναμετάδοση αιτήσεων δαπέδου μεταξύ

δικτυακών τόπων, διαχείριση προσωρινών δικαιωμάτων πρόσβασης σε δεδομένα, εξουσιοδότηση ή μη της χρήσης των δεδομένων ανάλογα με τις πληροφορίες του διαχειριστή συνεδρίας, ανακοίνωση αλλαγών στην κατάσταση του δαπέδου στους χρήστες. Για τα παραπάνω ο διαχειριστής δαπέδου διατηρεί ένα κατάλογο όπου καταγράφονται οι καταστάσεις όλων των δαπέδων των ενεργών συνεδριών. Όταν μια εφαρμογή ενεργοποιείται, μια παράσταση αντικειμένου τύπου ‘δάπεδο’ αντιστοιχίζεται στη εφαρμογή. Σε κάθε στιγμή μόνο ο κάτοχος του δαπέδου έχει το δικαίωμα ελέγχου της εφαρμογής. Αιτήσεις δαπέδου αναμεταδίδονται σε ομότιμους διαχειριστές δαπέδου και ικανοποιούνται ανάλογα με την υφιστάμενη πολιτική δαπέδου (π.χ. αλλαγή δαπέδου, απόφαση αλλαγής από τον διαχειριστή, κατοχύρωση δαπέδου βάση κανόνα first come, first serve, κλπ).

3.5 Μοντέλο Αντιγράφων Αντικειμένων

Τελευταία συνιστώσα των μηχανισμών σύγχρονης συνεργασίας είναι αυτή του μηχανισμού αντιγραφής αντικειμένων(Object Replication Model).



Εικόνα 11

Στην εικόνα 11, καταγράφεται ένα υποθετικό σενάριο συνεργατικής δραστηριότητας το οποίο θα μας βοηθήσει να καταλάβουμε καλύτερα το ρόλο του Object Replication Model. Συγκεκριμένα, η υποθετική διεπαφή που παρατηρούμε (Εικόνα 11), βλέπουμε ότι παρουσιάζει στα μέλη της ομάδας που μετέχουν στη συγκεκριμένη υποθετική σύγχρονη συνεργατική σύνοδο και τα οποία εμφανίζονται

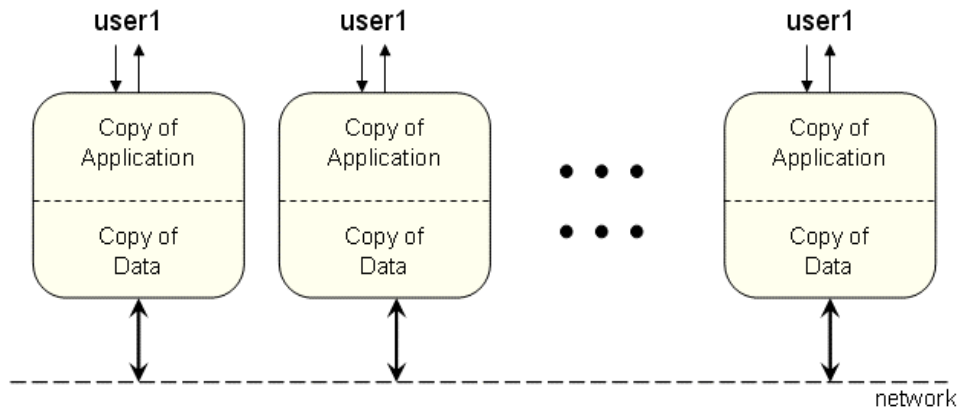
στα αριστερά, αντίγραφο ενός γραφικού αντικειμένου (π.χ. ένα χάρτη). Όταν πραγματοποιείται κάποια αλλαγή στην κατάσταση ενός αντικειμένου από ένα μέλος της ομάδας μέσω της client εφαρμογής του (π.χ. εισαγωγή σχολίου ‘Ποιος το έχει επιλέξει?’), αυτή η αλλαγή θα καταγράφεται και θα αναμεταδίδεται (broadcasting) στα υπόλοιπα μέλη της ομάδας ενημερώνοντας την διεπαφή. Αυτό σημαίνει ότι η αλλαγή θα αποτυπώνεται στο αντίγραφο του αντικειμένου στις client εφαρμογές των εταίρων και ταυτόχρονα θα καταχωρείται στον collaborative server έτσι ώστε να υποστηρίζεται η συνέχεια (persistency) της δραστηριότητας. Για την υλοποίηση της παραπάνω λειτουργικότητας πρέπει να αναπτυχθεί μια συνιστώσα που θα αναλαμβάνει την διαχείριση αντιγράφων αντικειμένων με σκοπό την ειδοποίηση (notification) εταίρων και την επίγνωση (awareness) ενεργειών που ο καθένας εκτελεί.

3.5.1 Αρχιτεκτονική πλήρους Αντιγραφής

Μέχρι στιγμής στη βιβλιογραφία υπάρχουν δύο τρόποι για να επιτύχεις αντιγραφή αντικειμένων, η πλήρης (Fully replicated architecture) αντιγραφή και η μερική (partially replicated architecture). Βέβαια αν ψάξει κανείς λίγο καλύτερα θα δει πως εμφανίζονται και άλλες που αναφέρονται ως ενεργητική αντιγραφή αντικειμένων, παθητική, υβριδική¹⁰, κ.α. Όλες αυτές οι ονομασίες όμως δεν είναι τίποτε άλλο εκτός από τις δύο που αναφέραμε στην αρχή με ελαφρές ενδεχομένως διαφοροποιήσεις.

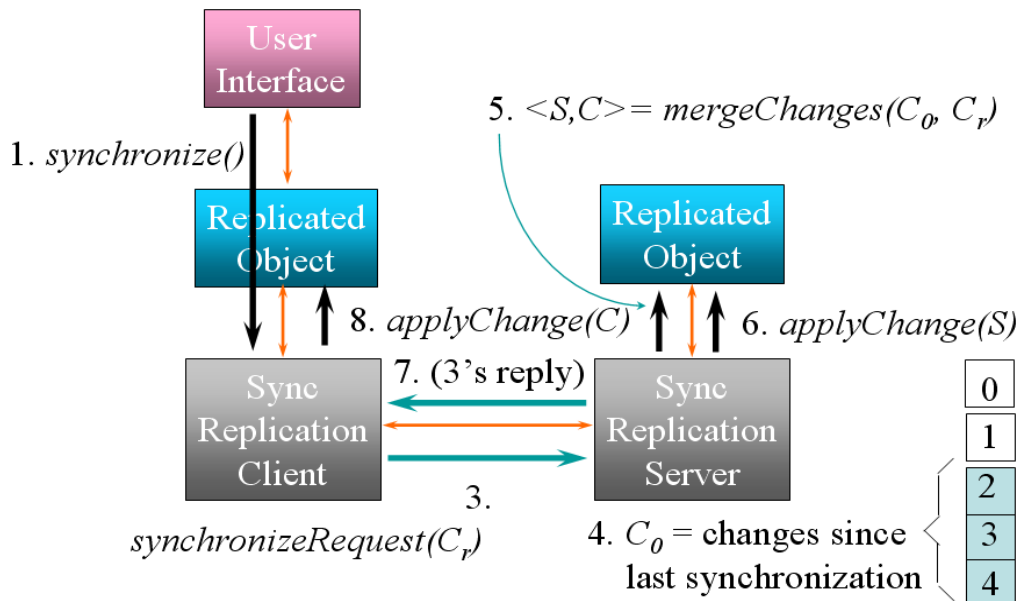
Σε συστήματα που υλοποιούν την πλήρη αντιγραφή αντικειμένων παρατηρούμε την αρχιτεκτονική που εμφανίζεται και στην εικόνα 12, όπου κάθε μηχανήμα εμπεριέχει και το μοντέλο(Model) και την όψη (View) αλλά και τον ελεγκτή (Controller).

¹⁰ Βλ. Cooperative Internet Computing Edited by Alvin et al. Kluwer Academic Publishers.



Εικόνα 12

Κάθε τέτοια αρχιτεκτονική, δηλαδή που κάθε μηχανή αποτελείται από αυτά τα τρία συνιστώσα και διακριτά τμήματα χαρακτηρίζεται ως fully replicated. Χαρακτηριστικό παράδειγμα τέτοιου είδους εφαρμογής είναι το sync, του οποίου τον τρόπο λειτουργίας θα περιγράψουμε παρακάτω ώστε να δούμε τι συμβαίνει και πως λειτουργεί ένα σύστημα που ακολουθεί αυτή τη λογική.



Εικόνα 13 Αρχιτεκτονική πλήρους αντιγραφής

Στην παραπάνω εικόνα βλέπουμε ένα client application (με τη μορφή που περιγράψαμε παραπάνω και που βρίσκεται αριστερά στην εικόνα) και ένα collaborative server(δεξιά στην εικόνα), ο οποίος αναλαμβάνει το να κρατά το latest state των data. Ο client λοιπόν ανά διαστήματα επικοινωνεί με το server λαμβάνοντας το τελευταίο state των data(sync), ενώ σε περίπτωση οποιασδήποτε αλλαγής με τα

δεδομένα που είχε ήδη εκπέμπεται ένα event locally ενημερώνοντας τον controller για τη συγκεκριμένη αλλαγή ενώ με τη σειρά του ο ίδιος ενημερώνει το view ώστε να αλλάξει απεικονίζοντας το Latest state των shared data.

Πιο αναλυτικά, κάθε αντικείμενο σε κάθε κατανεμημένη εφαρμογή περιέχει όλη εκείνη την πληροφορία που χρειάζεται ώστε να εμφανιστεί στην οθόνη του υπολογιστή χωρίς να χρειάζεται να γνωρίζει την κατάσταση οποιουδήποτε άλλου αντιγράφου. Συγκεκριμένα όλη η πληροφορία που αναπαρίσταται σε κάθε συνεργατικό αντικείμενο, είναι de-centralized, που σημαίνει ότι είναι αποθηκευμένη τοπικά στο μηχάνημα. Ας υποθέσουμε ότι έχουμε ένα μηχάνημα που περιέχει ένα γραφικό περιβάλλον με συνεργατικά διαδραστικά αντικείμενα, παραδείγματος χάριν ένα κουμπί (Collaborative-Button). Με το που δημιουργηθεί λοιπόν η διεπαφή το αντικείμενο αυτό ρωτάει τον collaborative server για την τελευταία κατάσταση του αντικειμένου ώστε να συγχρονίσει ανάλογα το τοπικό replica. Με αυτό τον τρόπο παραμένουν συγχρονισμένες όλες οι εφαρμογές όλων των συμμετεχόντων.

Πλεονέκτημα¹¹ αυτής της σχεδίασης είναι ότι σε περίπτωση που “πέσει” ο κεντρικός server που κρατάει τα αντίγραφα όλων των αντικειμένων, δεν καταρρέουν και οι διεπαφές των clients, παρά μένουν μόνο μη συγχρονισμένες. Άλλα πλεονεκτήματα είναι η μικρή απαίτηση σε εύρους ζώνης δικτύου και η μεγάλη ταχύτητα ανάδρασης σε τοπικό επίπεδο.

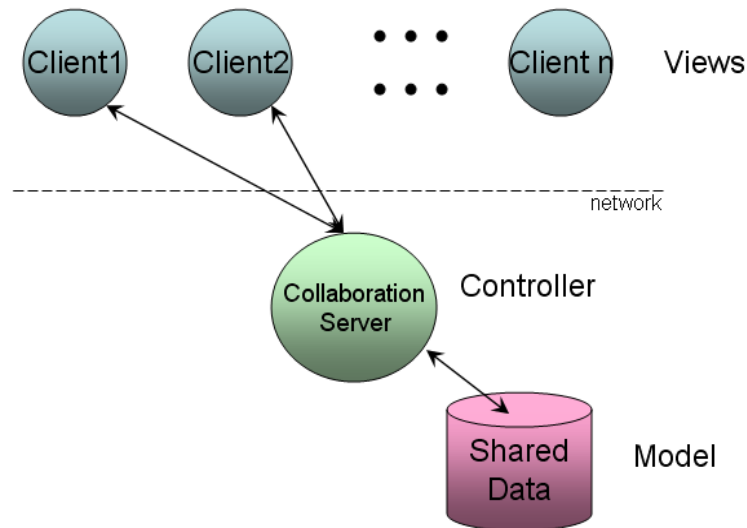
Τα μειονεκτήματα αυτού του τρόπου υλοποίησης εστιάζονται κυρίως στον συντονισμό των αντιγράφων, ενώ είναι χαρακτηριστικό ότι έχουμε η επανάληψη της πληροφορίας τόσες φορές όσες και τα μέλη που συμμετέχουν σε ένα collaborative session, (data redundancy).

3.5.2 Αρχιτεκτονική μερικής Αντιγραφής

Για να χαρακτηριστεί ένα σύστημα ως μερικής αντιγραφής (partially replicated) πρέπει να ακολουθεί την ακόλουθη φιλοσοφία που θα περιγράψουμε ακολούθως. Σε ένα τέτοιο σύστημα κάθε client application αυτό που έχει είναι μόνο ένα dummy view του μοντέλου που βρίσκεται distributed σε ένα collaborative server. Συγκεκριμένα κάθε φορά που θέλει να κάνει μια αλλαγή ένας χρήστης, αιτεί στο σύστημα για να πραγματοποιήσει την αλλαγή, το σύστημα με τη σειρά του αλλάζει

¹¹ Isenhour, 2001

το κοινό για όλους μοντέλο και ταυτόχρονα ενημερώνει και το view που έκανε αίτηση για την αλλαγή ώστε να την πραγματοποιήσει ενώ ταυτόχρονα ενημερώνει και τα υπόλοιπα dummy views ώστε να την εφαρμόσουν και αυτά.

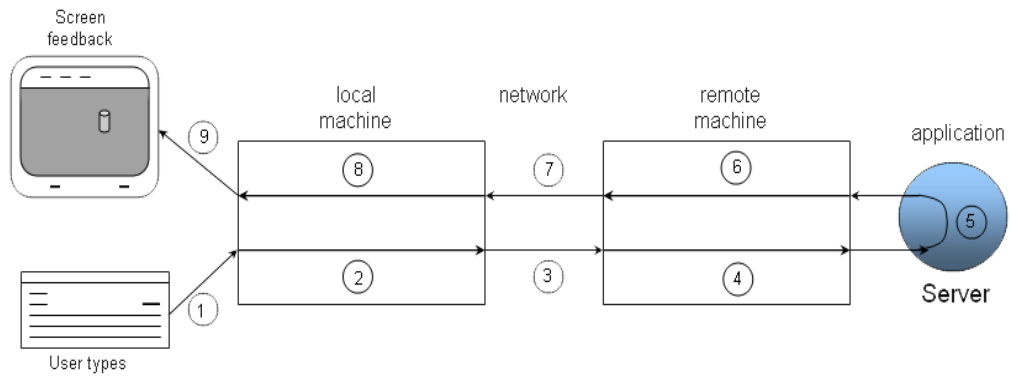


Εικόνα 14 Αρχιτεκτονική μερικής αντιγραφής

Ας δούμε συγκεκριμένα την αλληλουχία των γεγονότων που συμβαίνουν στην περίπτωση που ένας χρήστης πληκτρολογεί σε ένα διαμοιραζόμενο(shared) συνεργατικό αντικείμενο το οποίο υποστηρίζει τη δυνατότητα του να κάνεις typing. Με το που ξεκινάει ο χρήστης το typing εκπέμπεται ένα event το οποίο περνάει στο επίπεδο δικτύου της εφαρμογής, περνάει μέσω του δικτύου στο απομακρυσμένο αντικείμενο όπου βρίσκεται ο collaboration server. Ο server με τη σειρά του ανανεώνει το μοντέλο του συγκεκριμένου αντικειμένου και εν συνεχεία δίνει εντολή σε όλους όσους βλέπουν το συγκεκριμένο διαμοιραζόμενο αντικείμενο εντολή για να αλλάξουν το view τους όσον αφορά το state του συγκεκριμένου αντικειμένου βάσει της αλλαγής που αιτήθηκε και συνέβη στο κοινό καταναεμημένο μοντέλο του συγκεκριμένου αντικειμένου.

Εύκολα μπορεί να καταλάβει κανείς ότι μειονέκτημα αυτής της διαδικασίας είναι, καθυστέρηση της ανάδρασης κάθε αντικειμένου που εξαρτάται από την ταχύτητα του δικτύου ενώ ταυτόχρονα αυξάνει το φόρτο του δικτύου αφού για κάθε συμβάν πρέπει ταυτόχρονα να ενημερώνονται όλοι οι χρήστες. Άλλο χαρακτηριστικό αυτής της αρχιτεκτονικής είναι ότι ουδεμία αλλαγή μπορεί να συμβεί σε οποιοδήποτε συνεργατικό αντικείμενο σε περίπτωση που πέσει ο collaboration server παγώνοντας έτσι τη συμπεριφορά του κάθε collaborative object.

Πλεονέκτημα αυτού του τρόπου δόμησης είναι ότι υπάρχει πραγματικό real time interaction και synchronization μέσω του σύγχρονου τρόπου ενημέρωσης των replicated Objects.



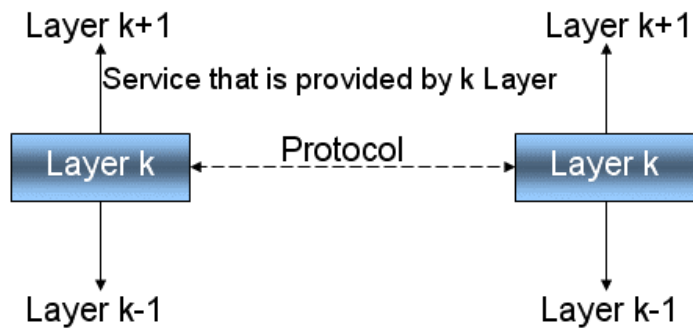
Εικόνα 15 User typing sequence diagram

4 Τεχνολογίες Δικτύου

Στο συγκεκριμένο κεφάλαιο θα ασχοληθούμε με τις πιο δημοφιλείς τεχνολογίες και τρόπους επικοινωνίας που χρησιμοποιούνται σήμερα μεταξύ χωρικά απομακρυσμένων μηχανών. Άλλες από αυτές είναι περισσότερο και άλλες λιγότερο δημοφιλείς ή αλλιώς προτιμητέες για διαφορετικούς λόγους η κάθε μια ανάλογα με την περίσταση και τις ανάγκες που υπάρχουν. Ο λόγος που τις αναφέρουμε είναι διότι θεωρούμε πως κατέχουν ξεχωριστή θέση σε οποιοδήποτε συνεργατικό Framework και ουδεμία υλοποίηση γίνεται χωρίς αυτές.

4.1 Πρωτοκόλλα και Υπηρεσίες

Δύο έννοιες που συχνά συγχέονται και που πρέπει να ξεκαθαρίσουμε είναι αυτή των υπηρεσιών και των πρωτοκόλλων. Υπηρεσία(service) είναι ένα σύνολο θεμελιωδών λειτουργιών που παρέχονται από ένα επίπεδο στο αμέσως ανώτερο επίπεδο. Η υπηρεσία καθορίζει ποιες λειτουργίες είναι προετοιμασμένο να εκτελέσει το επίπεδο εκ μέρους των χρηστών του, αλλά δεν λέει τίποτα για το πως υλοποιούνται αυτές. Η υπηρεσία σχετίζεται με τη διασύνδεση που υπάρχει ανάμεσα σε δύο επίπεδα, με το κατώτερο επίπεδο να είναι ο παροχέας της υπηρεσίας και το ανώτερο επίπεδο ο χρήστης της υπηρεσίας. Αντίθετα το πρωτόκολλο είναι ένα σύνολο κανόνων που καθορίζουν τη μορφή και τη σημασία των πακέτων ή μηνυμάτων που ανταλλάσσονται ανάμεσα στις ομότιμες οντότητες ενός επιπέδου. Οι οντότητες αυτές χρησιμοποιούν πρωτόκολλα για να υλοποιήσουν τον ορισμό των υπηρεσιών τους. Είναι όμως ελεύθερες να αλλάξουν τα πρωτόκολλα κατά βούληση, με την προϋπόθεση ότι δεν αλλάζουν την υπηρεσία που είναι ορατή στους χρήστες τους. Κατ' αυτή την έννοια, η υπηρεσία και το πρωτόκολλο είναι εντελώς ανεξάρτητα.

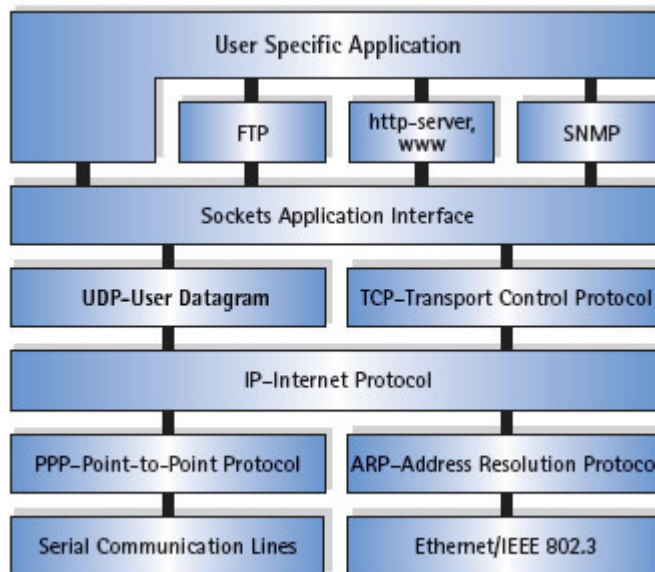


Εικόνα 16 Σχέση ανάμεσα υπηρεσίες και πρωτόκολλα

Πολλά παλαιότερα πρωτόκολλα δεν έκαναν διάκριση ανάμεσα στις υπηρεσίες και τα πρωτόκολλα. Στην πράξη, ένα τυπικό επίπεδο μπορεί να είχε μια θεμελιώδη λειτουργία “ΑΠΟΣΤΟΛΗ ΠΑΚΕΤΟΥ” με το χρήστη να παρέχει ένα δείκτη σε ένα πλήρως συναρμολογημένο πακέτο. Αυτή η σχεδίαση σήμαινε ότι όλες οι αλλαγές στο πρωτόκολλο ήταν άμεσα ορατές στους χρήστες. Οι περισσότεροι σχεδιαστές δικτύων θεωρούν πια τις σχεδιάσεις αυτές ως σοβαρές γκάφες και ως παραδείγματα προς αποφυγήν.

Περνάμε τώρα σε ένα από τα σημαντικότερα μοντέλα αναφοράς που χρησιμοποιείται μέχρι και σήμερα από πλήθος εφαρμογών: το TCP/IP, το οποίο αποτελεί τη βάση του Internet. Επειδή όλες οι τεχνολογίες που θα αναφέρουμε παρακάτω έχουν ως βάση το συγκεκριμένο πρωτόκολλο, καλό είναι να δούμε κάποια βασικά πράγματα για αυτό, καθώς στο τελευταίο κεφάλαιο θα δούμε πως θα προσθέσουμε κάποια εμβόλιμα επιπλέον επίπεδα ώστε να προσφέρουμε στο επίπεδο εφαρμογών επιπλέον υπηρεσίες.

Το TCP/IP, λόγω των απαιτήσεων που υπήρχαν εκείνη την εποχή οδηγήθηκε στην επιλογή ενός δικτύου μεταγωγής πακέτων που βασίζεται σε ένα ασυνδεδασμένο επίπεδο διαδικτύου. Το επίπεδο αυτό, που ονομάζεται *επίπεδο διαδικτύου* (internet layer), είναι ο ακρογωνιαίος λίθος ολόκληρης της αρχιτεκτονικής. Η δουλειά του είναι να επιτρέπει στους υπολογιστές υπηρεσίας να εισάγουν τα πακέτα τους σε οποιοδήποτε δίκτυο και αυτά να ταξιδεύουν ανεξάρτητα από τον προορισμό τους. Το επίπεδο διαδικτύου ορίζει μια επίσημη μορφή για τα πακέτα καθώς και ένα επίσημο πρωτόκολλο που ονομάζεται Πρωτόκολλο Διαδικτύου ή IP (Internet Protocol). Η δουλειά του επιπέδου διαδικτύου είναι να παραδίδει τα πακέτα IP εκεί όπου πρέπει να πάνε (Όπως παρατηρούμε και στην παρακάτω εικόνα, το επίπεδο διαδικτύου είναι το πιο χαμηλό μετά το φυσικό (επίπεδο Host προς δίκτυο)).



Εικόνα 17 Η στοίβα πρωτοκόλλων του TCP/IP

Το *επίπεδο συνόδου*, είναι αυτό που επιτρέπει σε χρήστες διαφορετικών μηχανών να εγκαθιδρύσουν συνόδους(sessions)μεταξύ τους.

Τέλος στο επίπεδο εφαρμογών, που είναι το υψηλότερο επίπεδο, περιέχεται μια ποικιλία πρωτοκόλλων που συχνά απαιτούνται από τους χρήστες. Σε αυτό το επίπεδο γράφονται όλες οι High Level εφαρμογές οι οποίες είναι πιο κοντά ή σε άμεσο Interaction με το χρήστη ανάλογα.

4.2 Sockets

Τα sockets αποτελούν το πιο low-level API για τη σύνδεση ενός υπολογιστή στο δίκτυο. Δεν υπάρχει εφαρμογή που να συνδέεται στο δίκτυο και να μην χρησιμοποιεί sockets. Τα sockets είναι ορατά στον προγραμματιστή ανάλογα με το API που χρησιμοποιεί. Π.χ. αν γράφει σε low-programming-level θα χειρίζεται ο ίδιος άμεσα τα ρεύματα εισόδου, εξόδου, τα wrapper classes που θα χρησιμοποιούν abstract Input και output stream. Αν π.χ. χρησιμοποιείται κάποιο πιο high-level api, άσχετα με το ότι ο προγραμματιστής δεν είναι σε άμεσο interaction με τα sockets, εντούτοις το high-level API που επέλεξε θα κάνει χρήση τους με transparent τρόπο προς αυτόν.

Όσον αφορά το επίπεδο μεταφοράς, τα sockets μπορούν να χρησιμοποιήσουν δύο τύπους μεταφοράς δεδομένων, το συνδεδεμοστρεφή και τον ασυνδεδεστικός, TCP και UDP αντίστοιχα. Το “*πρωτόκολλο UDP*” (Unreliable Datagram Protocol) το οποίο μπορεί δυνητικά να εφαρμοστεί στο επίπεδο μεταφοράς, δεν επιτρέπει ανοιχτή

επικοινωνία μεταξύ του υπολογιστή που στέλνει το μήνυμα και του υπολογιστή που το λαμβάνει. Δεν εξασφαλίζει όμως το ότι οι πληροφορίες θα φτάσουν με επιτυχία. Ο τρόπος αποστολής των πληροφοριών θυμίζει αυτόν του γραμματοκιβωτίου που τοποθετείται σε μια καθορισμένη ταχυδρομική διεύθυνση. Το “*πρωτόκολλο TCP*”(Transmission Control Protocol) εντούτοις παρέχει ακριβώς το επίπεδο αξιοπιστίας που λείπει από το UDP. Στο πρωτόκολλο TCP οι υπολογιστές συνδέονται απ’ ευθείας όπως στη συμβατική τηλεφωνική σύνδεση. Όση ώρα η σύνδεση είναι ανοιχτή ανταλλάσσονται πληροφορίες και στο τέλος η σύνδεση κλείνει. Η πληροφορία κατ’ αυτό τον τρόπο είναι μια ροή από bytes, εγγυημένα αξιόπιστη και σε μορφή FIFO(First In First Out). Κάθε πακέτο πληροφοριών που στέλνεται μέσω μιας σύνδεσης TCP πρέπει να περιέχει πληροφορίες οι οποίες προσδιορίζουν τη συγκεκριμένη σύνδεση καθώς και τη σχετική θέση του πακέτου μέσα στο μήνυμα. Μόλις ληφθεί το πακέτο ο υπολογιστής-αποδέκτης στέλνει ένα μήνυμα ότι το έλαβε σωστά. Εάν ο υπολογιστής που στέλνει το πακέτο δεν λάβει τέτοια επιβεβαίωση μέσα σε ένα καθορισμένο διάστημα, τότε το στέλνει εκ νέου.

Στον προγραμματιστή δίνεται βέβαια η δυνατότητα ανάλογα με τις ανάγκες του να επιλέξει ο ίδιος τον τρόπο σύνδεσης που επιθυμεί. Βέβαια δεν αποκλείεται ένα ολοκληρωμένο σύστημα να χρησιμοποιεί και τους δύο τρόπους επικοινωνίας ανάλογα με τις υπηρεσίες και τα καθήκοντα που είναι να εκτελεστούν εκείνη τη στιγμή¹².

Συμπερασματικά, τα sockets μας δίνουν τη δυνατότητα υλοποίησης εκείνης της διασύνδεσης, βάσει της οποίας μπορεί μια εφαρμογή να συνδέεται με μια άλλη μέσω οποιουδήποτε δικτύου υποστηρίζει το TCP/IP

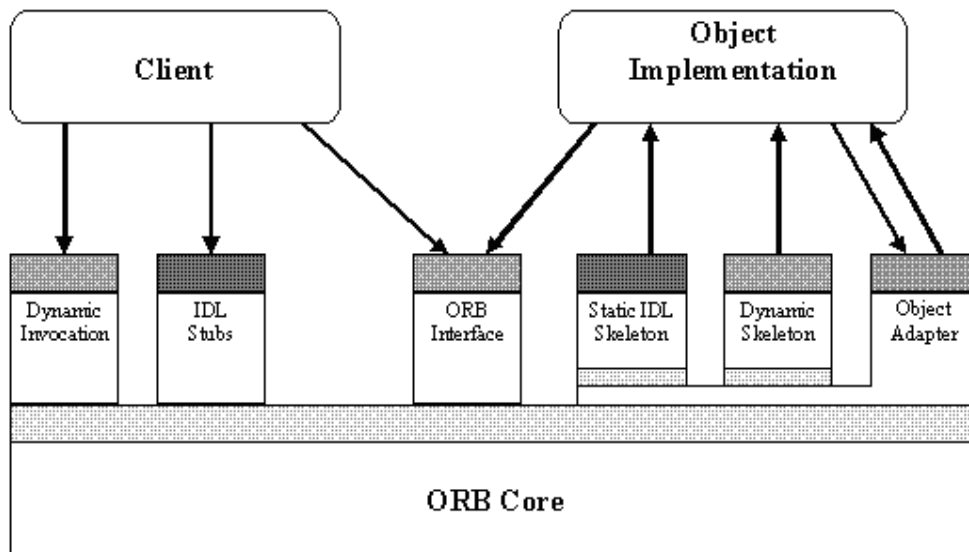
Στα πλεονεκτήματά των συγκαταλέγεται το ότι δεν υπάρχει καμία γλώσσα προγραμματισμού που να μην υποστηρίζει αυτό τον primitive τρόπο δικτύωσης. Είναι ο απλούστερος και ταχύτερος από όλους αφού δεν παρεμβάλλεται η επιπλέον πολυπλοκότητα και λειτουργικότητα επιπλέον επιπέδων, π.χ. SOAP Layer. Ένα ακόμη θετικό στοιχείο είναι ότι σε αυτή τη νηπιακή περίοδο, τεχνολογικά, των φορητών συσκευών(pda, cell phone, palmtop,...), ένας από τους λίγους προς το παρόν επιτρεπτούς τρόπους δικτύωσης των εφαρμογών που αναπτύσσονται πάνω σε αυτά είναι τα sockets. Ας μην ξεχνάμε ότι εισάγονται φοβεροί περιορισμοί όσον αφορά την ποικιλία των δυνατοτήτων που έχει κάποιος προγραμματίζοντας φορητές





¹² Π.χ. όπως φαίνεται και στο παράδειγμα Bingo της sun, στο java Tutorial, παρατηρούμε ότι χρησιμοποιούνται και οι δύο τρόποι επικοινωνίας.

συσκευές. Παράδειγμα αποτελεί η ανικανότητα κάθε Browser οποιασδήποτε φορητής συσκευής να εκτελέσει ένα applet.

4.3 Common Object Broker

Η OMG (Object Management Group) αναπτύσσει πρότυπα βασισμένη στην τεχνολογία του αντικειμένου. Σύμφωνα με την αρχιτεκτονική που προτείνεται από την OMG και την OMA (Object Management Architecture), το κάθε κομμάτι λογισμικού παρουσιάζεται σαν αντικείμενο (Object). Τα διάφορα αντικείμενα επικοινωνούν μεταξύ τους μέσω του ORB (Object Request Broker), ο οποίος αποτελεί το στοιχείο-κλειδί γι' αυτήν την επικοινωνία. Ο ORB παρέχει τους μηχανισμούς εκείνους μέσω των οποίων τα αντικείμενα κάνουν τις αιτήσεις και συλλέγουν τις αποκρίσεις. Η CORBA (Common Object Request Broker Architecture) αποτελεί το πρότυπο της OMG για τον ORB.



-  Διασυνδότης ανεξάρτητος από τον ORB
-  Συγκεκριμένα Stubs και Skeletons
-  Πιθανότητα πολλαπλών συνδέσμων αντικειμένου
-  Διασυνδότης εξαρτώμενος από τον ORB

Εικόνα 18 Αρχιτεκτονική Common Object Request Broker (CORBA)

Κατά την CORBA, ένας πελάτης ο οποίος ζητάει κάποιες υπηρεσίες από κάποιο αντικείμενο, κάνει μία αίτηση (*request*) η οποία μεταβιβάζεται στον ORB και ο οποίος είναι στη συνέχεια υπεύθυνος για την προώθηση της αίτησης προς τον

κατάλληλο εξυπηρετητή. Η αίτηση αυτή περιέχει όλες τις απαραίτητες πληροφορίες που απαιτούνται προκειμένου να υλοποιηθεί και αυτές είναι: το επιθυμητό αντικείμενο, επιθυμητές υπηρεσίες και τυχόν παράμετροι. Για να είναι κατανοητή όμως η αίτηση θα πρέπει να έχει μία κατάλληλη μορφή και γι' αυτό το λόγο γίνεται προς τον ORB μέσω διασυνδετών (interfaces: "Dynamic Invocation" και "IDL Stubs"). Για τον ίδιο λόγο η προώθηση της αίτησης προς τον εξυπηρετητή γίνεται μέσω διασυνδετών ("Static IDL Skeleton" και "IDL Skeleton"). Ένας διασυνδέτης αποτελεί μία περιγραφή ενός συνόλου από πιθανές λειτουργίες τις οποίες ένας πελάτης μπορεί να αιτήσει από ένα αντικείμενο. Ένα αντικείμενο ικανοποιεί έναν διασυνδέτη εάν μπορεί να προσδιοριστεί σαν το αντικείμενο-στόχος (target object) για κάθε δυνατή αίτηση η οποία περιγράφεται από τον διασυνδέτη.

Στο σημείο αυτό δεν θα πρέπει να αγνοήσουμε την ιδιότητα της κληρονομικότητας που χαρακτηρίζει τους διασυνδέτες και η οποία παρέχει τον συνθετικό εκείνο μηχανισμό που επιτρέπει σ' ένα αντικείμενο να υποστηρίζει πολλαπλούς διασυνδέτες.

Οι διασυνδέτες καθορίζονται πλήρως μέσω της OMG IDL (Interface Definition Language). Η IDL δεν αποτελεί γλώσσα προγραμματισμού, όπως οι γνωστές γλώσσες προγραμματισμού, αλλά μία καθαρά περιγραφική γλώσσα η οποία δεν περιλαμβάνει δομές αλγορίθμων ή μεταβλητές. Η γραμματική της αποτελεί υποσύνολο της ANSI C++ με πρόσθετες κατασκευές για την υποστήριξη μηχανισμών για την επίκληση απομακρυσμένων λειτουργιών. Οι πελάτες δεν είναι "γραμμένοι" σε OMG IDL αλλά σε γλώσσα για την οποία έχει προσδιοριστεί η αντιστοιχία της με την OMG IDL.

4.4 Remote Method Invocation

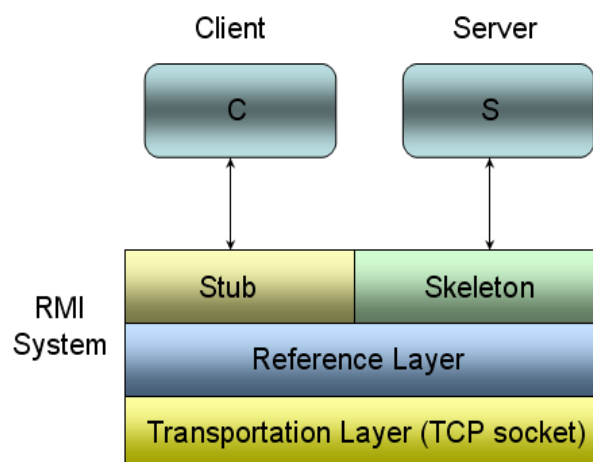
Όλοι έχουμε ακούσει τη φράση “το δίκτυο είναι ο υπολογιστής του μέλλοντος”. Για τη Java είναι επίσης γνωστό ότι “μια φορά γράφεις, μια φορά μεταγλωττίζεις, τρέχει παντού”. Ο παγκόσμιος ιστός(world wide web) είναι βασισμένος στην αρχή ότι οι πηγές των πληροφοριών μπορεί να βρίσκονται παντού στο Internet. Με τη βοήθεια ενός web browser μπορούμε να έχουμε πρόσβαση στις πηγές αυτές σαν να βρίσκονταν στο χώρο εργασίας του δικού μας υπολογιστή. Με άλλα λόγια, πέφτουν οι φραγμοί ανάμεσα στους διάφορους τύπους υπολογιστών και στις αποστάσεις μεταξύ τους. Η *επίκληση απομακρυσμένων μεθόδων*(Remote Method Invocation-RMI) επιτρέπει σε ένα αντικείμενο της java που τρέχει σε ένα υπολογιστή, να

καλέσει μια μέθοδο ενός άλλου αντικειμένου της java που εκτελείται σε άλλο μηχανήμα. Η προσπέλαση της μεθόδου αυτής γίνεται με τη βοήθεια του δικτύου.

Στο κατανεμημένο περιβάλλον, το αντικείμενο που καλεί ένα άλλο, χαρακτηρίζεται ως πελάτης (client), ενώ αυτό που δέχεται την κλήση λέγεται εξυπηρετητής ή διακομιστής (Server). Εάν το δεύτερο αντικείμενο καλέσει μια μέθοδο ενός τρίτου αντικειμένου, τότε μεταβάλλεται σε πελάτη για την κλήση αυτή ενώ το τρίτο αντικείμενο είναι ο εξυπηρετητής.

Το σύστημα RMI αποτελείται από τρία διακριτά επίπεδα(όπως φαίνεται και στο παρακάτω σχήμα). Αυτά είναι:

- Το επίπεδο Σκελετού/Στελέχους (Skeleton/Stub)
- Το επίπεδο Αναφοράς (Reference Layer)
- Το επίπεδο Μεταφοράς (Transport Layer)



Εικόνα 19 RMI αρχιτεκτονική

Το επίπεδο Στελέχους / Σκελετού, είναι υπεύθυνο για τη μετάδοση των δεδομένων από τον υπολογιστή-πελάτη προς το απομακρυσμένο αντικείμενο του εξυπηρετητή, καθώς και για την επιστροφή των αποτελεσμάτων από το απομακρυσμένο αντικείμενο στον πελάτη. Το επίπεδο αναφοράς ασχολείται με τη μεταφορά των δεδομένων σε πιο χαμηλό επίπεδο καθώς και με τα πρωτόκολλα που απαιτούνται για να γίνει η επικοινωνία Σκελετού/Στελέχους. Το επίπεδο μεταφοράς είναι γενικά υπεύθυνο για τη δημιουργία, τη διαχείριση και την παρακολούθηση των συνδέσεων. Παρακολουθεί επίσης και χειρίζεται όλες τις κλήσεις μεταξύ του εξυπηρετητή και των πελατών.

Τα στελέχη και οι σκελετοί δημιουργούνται με ένα βοηθητικό πρόγραμμα που ονομάζεται RMI compiler. Για να μπορεί να καλείται ένα απομακρυσμένο αντικείμενο πρέπει να καταχωρηθεί με ένα συγκεκριμένο όνομα στο μητρώο RMI(RMI registry) που βρίσκεται στον εξυπηρετητή. Το απομακρυσμένο αντικείμενο μπορεί τότε να προσπελαστεί με τη χρήση ενός URL που ακολουθεί το πρωτόκολλο RMI.

Τελειώνοντας οφείλουμε να σημειώσουμε ότι πρέπει οι μηχανές που επικοινωνούν με τη χρήση του RMI πρέπει να τρέχουν την ίδια έκδοση του java runtime environment ή τουλάχιστον συμβατές εκδόσεις. Μπορεί να θεωρηθεί μειονέκτημα της συγκεκριμένης τεχνολογίας ότι δεν υποστηρίζεται προς το παρόν από τη Java Mobile Edition (JME).

4.5 XML-Remote Procedure Call

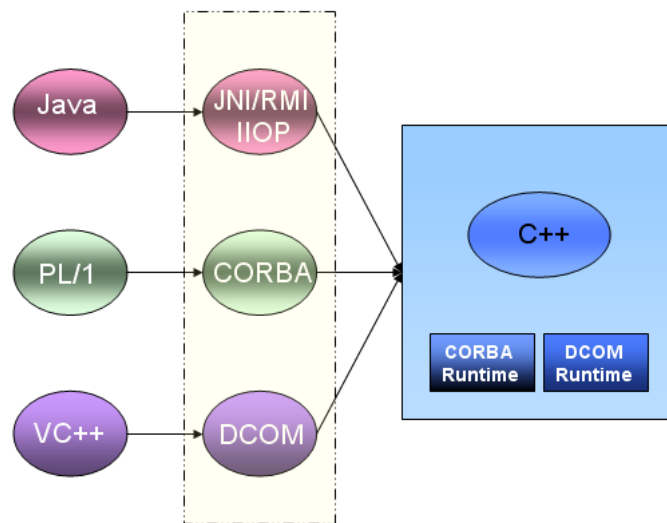
Τα τελευταία χρόνια έχουν γίνει πολλές προσπάθειες να δημιουργηθεί ένα πρότυπο πρωτόκολλο για απομακρυσμένες κλήσεις διαδικασιών(remote procedure calls, RPC), που είναι ένας τρόπος για ένα πρόγραμμα υπολογιστή να καλεί μια διαδικασία σε ένα άλλο πρόγραμμα επάνω σε ένα δίκτυο, σαν το Internet.

Συχνά, αυτά τα πρωτόκολλα δεν έχουν καμία σχέση με τις γλώσσες προγραμματισμού, επιτρέποντας σε ένα πρόγραμμα client γραμμένο σε C++ πχ. να καλεί ένα απομακρυσμένο server βάσης δεδομένων που έχει γραφεί σε java, χωρίς καμία από τις δύο πλευρές να ξέρει ή να ενδιαφέρεται για τη γλώσσα υλοποίησης της άλλης.

Οι προσπάθειες για RPC καθοδηγούνται σήμερα από τις μεγάλες ταχύτητες των υπηρεσιών Web, των προγραμμάτων δικτύωσης που χρησιμοποιούν το Web για να παρέχουν δεδομένα σε μια μορφή που μπορεί να αφομοιωθεί εύκολα από άλλο λογισμικό. Οι υπηρεσίες Web για να παρέχουν δεδομένα σε μια μορφή που μπορεί να αφομοιωθεί εύκολα από άλλο λογισμικό. Οι υπηρεσίες Web χρησιμοποιούνται για να μοιράζονται πληροφορίες πιστοποίησης κωδικού πρόσβασης ανάμεσα σε ιστοθέσεις, για να διευκολύνουν συναλλαγές ηλεκτρονικού εμπορίου ανάμεσα σε καταστήματα, για ανταλλαγή πληροφοριών ανάμεσα σε επιχειρήσεις και για άλλες νεωτεριστικές προσφορές.

Το 1998, μια νέα τεχνολογία εισήχθη σε αυτή την περιοχή: το XML-RPC ένα πρωτόκολλο για τη χρήση HTTP και XML για απομακρυσμένες κλήσεις διαδικασιών.

Το XML –RPC που παραθέτουμε σε αυτή την ενότητα έχει μια ουσιαστική διαφορά με τεχνολογίες που παραθέσαμε πιο πάνω(Remote Method Invocation και IDL - Common Object Request Broker Architecture), έλλειψη πολυπλοκότητας. Απεναντίας όμως οι πολύπλοκες αυτές τεχνολογίες έχουν σχεδιαστεί ώστε να παρέχουν στιβαρές λύσεις σε μια μεγάλη ποικιλία απομακρυσμένων εργασιών.

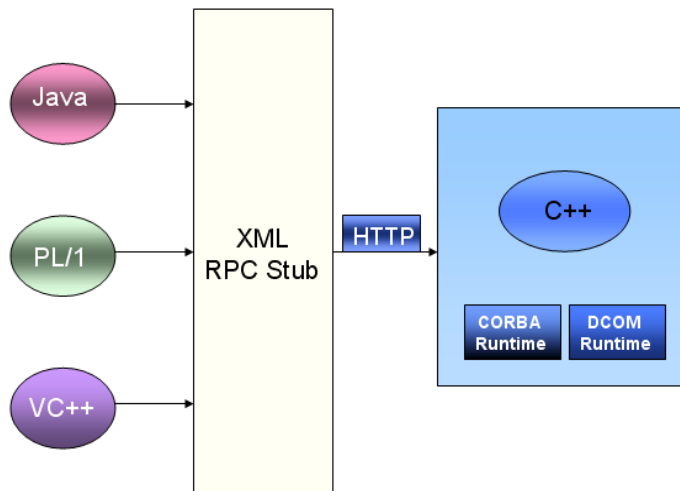


Εικόνα 20

Αυτή η ευφυΐα αποτελούσε ένα από τα εμπόδια για την υιοθέτηση υπάρχουσών προσπαθειών RPC.

Το 1998, ο Dave Winer της UserLand Software εξέδωσε την προδιαγραφή για το XML-RPC , ένα πρωτόκολλο απομακρυσμένων κλήσεων διαδικασιών, που η εταιρεία του είχε υλοποιήσει στο UserLand Frontier, ένα ανεξάρτητο πλατφόρμας σύστημα διαχείρισης περιεχομένου για το Internet. Το πρωτόκολλο αναπτύχθηκε σε συνεργασία με τη Microsoft.

Το XML-RPC ανταλλάσσει πληροφορίες χρησιμοποιώντας ένα συνδυασμό HTTP, του πρωτοκόλλου για το World Wide Web και της XML, μιας μορφής οργάνωσης δεδομένων ανεξάρτητης από το λογισμικό που χρησιμοποιείται για ανάγνωση και εγγραφή των δεδομένων.



Εικόνα 21

Στην παραπάνω εικόνα μπορούμε να δούμε και να καταλάβουμε καλύτερα το διαφορετικό τρόπο προσέγγισης του θέματος κλήσης απομακρυσμένων διαδικασιών με τις υπόλοιπες υπάρχουσες τεχνολογίες που παρουσιάσαμε πιο πάνω. Παρατηρούμε ένα ενιαίο τρόπο προσπέλασης του απομακρυσμένου αντικειμένου-μεθόδου ο οποίος ορίζεται και προσπελάζεται επομένως με τον ίδιο τρόπο ανεξαρτήτως γλώσσας προγραμματισμού.

Συνεχίζοντας πρέπει να αναφέρουμε ότι το XML υποστηρίζει συγκεκριμένους τύπους δεδομένων, κατά κύριο λόγο primitive, ενώ δίνει-παρέχει τη δυνατότητα μεταφοράς και πιο πολύπλοκων δομών, πάλι όμως αποτελούμενων από βασικούς(primitive). Συγκεκριμένα υποστηρίζονται οι εξής:

- *Base64* – Δυαδικά δεδομένα σε μορφή Base 64
- *Boolean* – Τιμές true-false που είναι 1 (true) ή 0 false
- *dateTime.iso8601*- Ένα string που περιέχει την ημερομηνία σε μορφή ISO8601 (όπως 20021009T19:20:15 για 7:20 μμ. (και 15 δευτερόλεπτα) της 9 Οκτωβρίου 2002)
- *double* – Προσημασμένοι αριθμοί κινητής υποδιαστολής οκτώ bytes
- *int* (καλείται επίσης και *i4*) – Προσημασμένοι ακέραιοι στην περιοχή από -2147483648 ως 2147483647, δηλαδή με ίδιο μέγεθος με τιμές int στη java
- *string* – Κείμενο

- *struct* – Ζεύγη ονόματος – τιμής σχετικών δεδομένων, όπου το όνομα είναι ένα string και η τιμή μπορεί να είναι οποιοσδήποτε από τους τύπους δεδομένων (αντίστοιχο με την κλάση HashMap της Java)
- *array* – Χρησιμοποιείται για να περιέχει πίνακες κάθε άλλου είδους δεδομένων, περιλαμβανομένων ακόμη και πινάκων.

Μια σημαντική έλλειψη του XML-RPC είναι ένας τρόπος αναπαράστασης δεδομένων σαν αντικείμενο. Αυτό το πρωτόκολλο δεν σχεδιάστηκε για αντικειμενοστραφή προγραμματισμό, παραταύτα μπορούμε να παραστήσουμε αρκετά περίπλοκα αντικείμενα με τους τύπους array και struct.

Από τη σχεδίασή του, το XML-RPC είναι ένα απλό πρωτόκολλο για απομακρυσμένες κλήσεις διαδικασιών, το οποίο είναι κατάλληλο για προγραμματισμό σε δίκτυο. Το πρωτόκολλο έχει γίνει ένα από τα βασικά στοιχεία των υπηρεσιών Web που υλοποιούνται από πολλούς προγραμματιστές λογισμικού για συστήματα Windows, Macintosh, Linux και Unix.

Το XML-RPC εκδόθηκε αρχικά σαν κομμάτι της συνεργασίας για κοινή ανάπτυξη ανάμεσα στη UserLand και τη Microsoft. Αποδείχθηκε μάλιστα τόσο δημοφιλές που περισσότερες από 60 υλοποιήσεις διατίθενται σήμερα για διάφορες γλώσσες και πλατφόρμες¹³.

Μετά την έκδοση του XML-RPC, η προδιαγραφή επεκτάθηκε από τη Microsoft, την IBM, τη Lotus, για να δημιουργήσουν ένα άλλο πρωτόκολλο RPC, με όνομα SOAP, Simple Object Access Protocol.

4.6 Simple Object Access Protocol

Το 1997, μεγάλες εταιρείες, όπως η Microsoft, άρχισαν να διερευνούν κατά πόσο ο κατακευματισμένος υπολογισμός μπορεί να βασιστεί στη γλώσσα XML. Ο σκοπός της έρευνας αυτής ήταν να γίνει εφικτή η επικοινωνία μεταξύ των εφαρμογών μέσω απομακρυσμένων κλήσεων διαδικασιών (Remote Procedure Calls – RPCs), χρησιμοποιώντας απλά πρωτόκολλα δικτύου, όπως το HTTP. Το 1999 έκανε την εμφάνισή του το SOAP¹⁴, ένας RPC μηχανισμός βασισμένος σε XML. Το 2000 ο οργανισμός W3C ασχολείται με την ιδέα αυτή και ύστερα από αρκετές αλλαγές,

¹³ Μια από τις δημοφιλέστερες υλοποιήσεις του είναι το Apache XML-RPC, μια υλοποίηση ανοιχτού κώδικα (open-source), από την εταιρία που έχει υλοποιήσει το Apache Servlet-jsp Container, το ant καθώς και άλλων πολύ γνωστών και δημοφιλών εργαλείων.

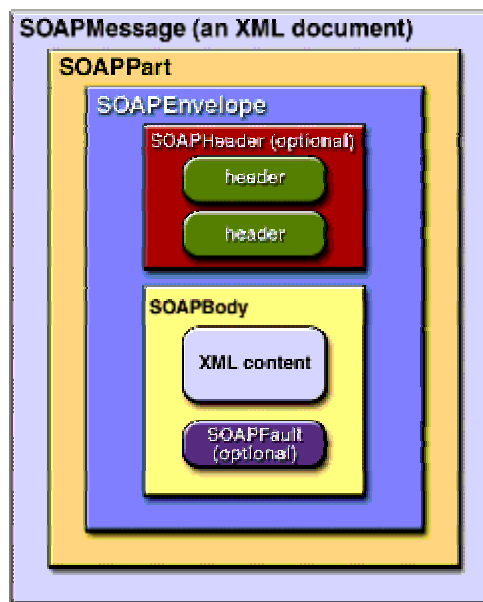
¹⁴ Για περισσότερες πληροφορίες ανατρέξτε στην ηλεκτρονική δι/ση: <http://www.w3.org/TR/soap/>

βελτιώσεις και τροποποιήσεις 2 ολόκληρων χρόνων, το 2003 δηλαδή, το SOAP με την έκδοση 1.2 γίνεται η προτεινόμενη προδιαγραφή – πρωτόκολλο για τις υπηρεσίες διαδικτύου.

Στον πυρήνα του, το SOAP, είναι μια προδιαγραφή για ένα απλό αλλά ταυτόχρονα ευέλικτο XML πρωτόκολλο δεύτερης γενιάς. Το σημαντικό είναι ότι καθώς η έρευνα ξεκίνησε από τον καταναμημένο υπολογισμό, το SOAP είναι το πλέον κατάλληλο πρωτόκολλο αφού ειδικεύεται σε τέτοια περιβάλλοντα. Το SOAP παρέχει τα κάτωθι χαρακτηριστικά και μηχανισμούς:

- 1) Μηχανισμός για τον ορισμό της πληροφορίας στην επικοινωνία – Στο SOAP, όλη η πληροφορία είναι καταχωρημένη σε ένα ξεκάθαρο και ταυτοποιήσιμο SOAP μήνυμα (SOAP message). Αυτό γίνεται μέσω ενός SOAP φακέλου (SOAP envelope), που περικλείει όλες τις απαραίτητες πληροφορίες. Ένα μήνυμα SOAP, μπορεί να έχει σώμα (body), το οποίο περιέχει πληροφορία σε XML δομή. Επίσης, μπορεί να διαθέτει έναν αριθμό από επικεφαλίδες (headers), οι οποίες ενσωματώνουν επιπλέον πληροφορίες έξω από το σώμα του μηνύματος.
- 2) Διεργασιακό μοντέλο – Αυτό το μοντέλο ορίζει ένα σύνολο κανόνων βάσει των οποίων γίνονται οι διαπραγματεύσεις μεταξύ των SOAP μηνυμάτων και του λογισμικού. Διακρίνεται από την απλότητά του.
- 3) Μηχανισμός για αντιμετώπιση σφαλμάτων – Το SOAP παρέχει το μηχανισμό αυτό με τη μορφή των SOAP faults, τα οποία όταν χρησιμοποιούνται μπορεί να προσδιοριστεί η πηγή που προκάλεσε το σφάλμα. Επιπλέον, παρέχουν τη δυνατότητα ανταλλαγής διαγνωστικών πληροφοριών μεταξύ των μελών της επικοινωνίας.
- 4) Μοντέλο επεκτασιμότητας – Το μοντέλο αυτό χρησιμοποιεί SOAP επικεφαλίδες για την υλοποίηση προεκτάσεων πάνω στο SOAP. Οι επικεφαλίδες περιέχουν κομμάτια από δεδομένα με δυνατότητα επέκτασης, τα οποία ταξιδεύουν μαζί με το μήνυμα και μπορούν να γίνουν στόχος για επέκταση σε συγκεκριμένους κόμβους του δικτύου.
- 5) Ευέλικτος μηχανισμός για αναπαράσταση δεδομένων – Ο μηχανισμός αυτός επιτρέπει στα δεδομένα σε οποιαδήποτε σειριακή μορφή κι αν βρίσκονται να αναπαρασταθούν σε XML μορφή.

- 6) Σύμβαση για αναπαράσταση των RPCs και των απαντήσεών τους σαν SOAP μηνύματα – Οι απομακρυσμένες κλήσεις διαδικασιών είναι αρκετά διαδεδομένες στον κατανεμημένο προγραμματισμό/ υπολογισμό και μπορούν να αναπαρασταθούν καλά μέσω του μηχανισμού αυτού σαν SOAP μηνύματα.
- 7) Πρωτόκολλο εγκαθίδρυσης σύνδεσης – Το πρωτόκολλο αυτό ορίζει μια αρχιτεκτονική για την οικοδόμηση συνδέσεων επικοινωνίας ώστε να είναι εφικτή η ανταλλαγή SOAP μηνυμάτων πάνω σε μέσα μεταφοράς και επικοινωνιακά κανάλια. Χρησιμοποιείται το HTTP πρωτόκολλο, καθώς είναι το πιο διαδεδομένο και ευρέως χρησιμοποιούμενο στο διαδίκτυο.



Εικόνα 22 Η δομή ενός SOAP μηνύματος

Όσον αφορά στα μηνύματα SOAP, εφόσον το πρωτόκολλο είναι βασισμένο στην XML, είναι αναμενόμενο τα μηνύματα να έχουν μια τέτοια μορφή και ουσιαστικά να μην είναι τίποτε άλλο, παρά XML έγγραφα. Το στοιχείο-ρίζα του μηνύματος είναι το soapenv:Envelope, το οποίο περικλείει το στοιχείο soapenv:Body, που περιέχει πληροφορία σχετιζόμενη με το σκοπό του μηνύματος. Το στοιχείο soapenv:Body με τη σειρά του μπορεί να περιέχει άλλα στοιχεία, τα οποία να σχετίζονται ή να αναπαριστούν την απομακρυσμένη κλήση διαδικασίας που πρόκειται να εκτελεστεί. Το όνομα της μεθόδου που θα κληθεί και το όνομα του στοιχείου αυτού περιέχονται μέσα στο σώμα του μηνύματος.

Τα μηνύματα αυτά στέλνονται μέσω πρωτοκόλλου HTTP με τη μέθοδο POST. Η απάντηση σε ένα τέτοιο μήνυμα, που και πάλι μεταφέρεται μέσω HTTP είναι και αυτή ένα SOAP μήνυμα. Το μήνυμα της απάντησης περιλαμβάνει κι αυτό από το συστατικό `soapenv:Envelope`, που περιέχει το `soapenv:Body`, το οποίο ενσωματώνει και τη βασική πληροφορία του μηνύματος, στην οποία συγκαταλέγεται και μια κωδικοποιημένη αναπαράσταση του αποτελέσματος της απομακρυσμένης κλήσης που πραγματοποιήθηκε.

Το SOAP παρέχει, όπως ήδη αναφέρθηκε, τη δυνατότητα σε ένα μήνυμα να συμπεριλαμβάνονται στοιχεία – επικεφαλίδες. Ο ρόλος τους είναι και ο σημαντικότερος λόγος που χρησιμοποιούμε τα μηνύματα SOAP και όχι απλά XML έγγραφα. Τα στοιχεία αυτά (headers) έχουν την ιδιότητα να αναπαριστούν ένα επιπρόσθετο και επεκτάσιμο είδος πληροφορίας το οποίο μεταφέρεται μαζί με το υπόλοιπο μήνυμα, χωρίς να τροποποιείται ο κυρίως πυρήνας του μηνύματος. Για να γίνει κατανοητό αυτό ας δούμε ένα παράδειγμα της αληθινής ζωής ακριβώς αντίστοιχο με την ιδέα των SOAP μηνυμάτων. Ας υποθέσουμε ότι θέλουμε να στείλουμε ένα έγγραφο αλλά και κάποιες επιπρόσθετες πληροφορίες χωρίς όμως να θέλουμε να μαρκάρουμε το έγγραφο με αυτές. Τοποθετούμε, λοιπόν, το έγγραφο στο φάκελο και στη συνέχεια προσθέτουμε μία ή δύο σελίδες χαρτί, οι οποίες περιγράφουν την επιπλέον πληροφορία που θέλαμε να στείλουμε. Η αντιστοιχία του παραδείγματος αυτού με το πρωτόκολλο που περιγράφουμε είναι ότι ο φάκελος είναι το στοιχείο `soapenv:Envelope`, το έγγραφο που θέλουμε να στείλουμε είναι το στοιχείο `soapenv:Body` και οι σελίδες με την επιπλέον πληροφορία είναι τα στοιχεία επικεφαλίδες (headers).

Η χρήση των επικεφαλίδων για την προσθήκη λειτουργικότητας σε ένα μήνυμα SOAP είναι γνωστή και σαν “*κατακόρυφη επέκταση*”, γιατί τα στοιχεία αυτά τοποθετούνται στην κορυφή του μηνύματος πριν το σώμα του.

Για λόγους ασφαλείας σε περιπτώσεις επεκτασιμότητας χρησιμοποιώντας στοιχεία επικεφαλίδες, η προδιαγραφή του SOAP ορίζει τη μεταβλητή `mustUnderstand`. Όταν η τιμή της είναι αληθής, τότε ο παραλήπτης του μηνύματος πρέπει να συμφωνήσει ότι αποδέχεται όλους τους όρους της επικεφαλίδας του μηνύματος προκειμένου να συνεχιστεί η επεξεργασία του. Στην περίπτωση που η τιμή της μεταβλητής αυτής είναι ψευδής, τότε η επικεφαλίδα έχει προαιρετικό ρόλο και η επεξεργασία του μηνύματος μπορεί να γίνει και από παραλήπτες που αγνοούν τις προδιαγραφές που θέτει η επικεφαλίδα.

Αντίστοιχα με την κατακόρυφη επεκτασιμότητα, υπάρχει και η οριζόντια, η οποία έχει να κάνει με την αντιστοίχιση διαφορετικών κομματιών του μηνύματος με διαφορετικούς παραλήπτες. Αυτό επιτυγχάνεται μέσω των μεσολαβητών SOAP (SOAP intermediaries). Αυτές οι εφαρμογές μπορούν να επεξεργάζονται μέρη του μηνύματος SOAP καθώς αυτό ταξιδεύει προς τον τελικό του προορισμό. Οι μεσολαβητές μπορούν να αποδέχονται και να προωθούν μηνύματα καθώς κάνουν και κάποιο είδος επεξεργασίας. Σε γενικές γραμμές, στον αληθινό κόσμο σπάνια ξεκινά ένα μήνυμα από κάποια τοποθεσία και φθάνει απευθείας στον προορισμό του. Για παράδειγμα ας πάρουμε το ηλεκτρονικό ταχυδρομείο. Όταν στέλνουμε ένα μήνυμα αυτό θα περάσει από διάφορους εξυπηρετητές και δρομολογητές προτού «εντοπίσει» τον προορισμό του. Αυτός είναι ένας από τους κυριότερους λόγους που χρειάζονται οι μεσολαβητές στα μηνύματα SOAP, καθώς επίσης χάρη σε αυτούς μπορούμε να έχουμε υπηρεσίες προστιθέμενης αξίας σε καταναμημένα συστήματα.

Ένα εργαλείο για την πραγματοποίηση SOAP συναλλαγών είναι ο Apache Axis, μια μηχανή SOAP ανοιχτού λογισμικού.

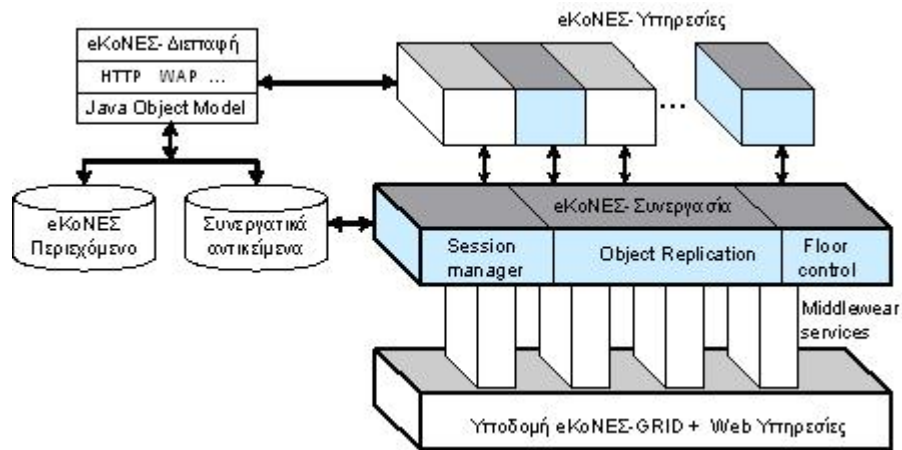
5 Τι είναι το eKoNEΣ?

5.1 Μια σύντομη επισκόπηση

Το πρόγραμμα eKoNEΣ εστιάζει στη διερεύνηση, σχεδιασμό, κατασκευή και πιλοτική επίδειξη τεχνολογικών εργαλείων που αποσκοπούν στην προαγωγή της καινοτομικής δραστηριότητας επιχειρηματικών φορέων της περιφέρειας της Κρήτης μέσω προηγμένων μορφών δικτύωσης και συνεργασίας. Το έργο εστιάζει στην ανάπτυξη ή σύσφιξη κοινοτικών ιστών (community webs) μεταξύ εταιρών (επιχειρηματικών, ερευνητικών, ΟΤΑ ή φορέων που εποπτεύονται από ΟΤΑ) μέσω συμπράξεων και συνεργασιών για την μεταφορά τεχνογνωσίας, την βελτίωση των τεχνικών ικανοτήτων των μελών και κατ' επέκταση την αύξηση της παραγωγικής δύναμης του κοινοτικού ιστού ή της ηλεκτρονικής κοινότητας.

Η προσέγγιση που προτείνεται βασίζεται στην ανάπτυξη προηγμένων εργαλείων που θα συνθέτουν ένα τεχνολογικό περιβάλλον βασισμένο στον παγκόσμιο ιστό, μέσω του οποίου θα μπορούν να γίνονται τα ακόλουθα:

- 1) e-Παρουσίαση: Επιχειρηματικοί φορείς θα μπορούν να εγγράφονται στο eKoNEΣ και χρησιμοποιώντας τα εργαλεία του να αποκτούν ατομική ηλεκτρονική παρουσίαση διαφημίζοντας υπηρεσίες / προϊόντα, παρουσιάζοντας συγκριτικά πλεονεκτήματα, κλπ. Η εγγραφή στο eKoNEΣ θα αποφέρει συγκριτικά πλεονεκτήματα στα μέλη και πρόσβαση σε υπηρεσίες προστιθέμενης αξίας (βλέπε παρακάτω)
- 2) e-Συνεργασία: Μέλη του eKoNEΣ θα έχουν πρόσβαση σε κλασικές μορφές ασύγχρονης και συγχρονισμένης επικοινωνίας και συνεργασίας όπως email, συζήτηση, συναντήσεις, πίνακα ανακοινώσεων, κλπ που θα καταγράφουν δυναμικά την αλληλεπίδραση των μελών της κοινότητας.
- 3) e-Services: Στην κατηγορία αυτή προβλέπεται να υποστηρίζονται εξειδικευμένες υπηρεσίες προστιθέμενης αξίας για τα μέλη της κοινότητας που θα αφορούν την ενημέρωση, διαβούλευση εταιρών, συναλλαγές, κλπ.



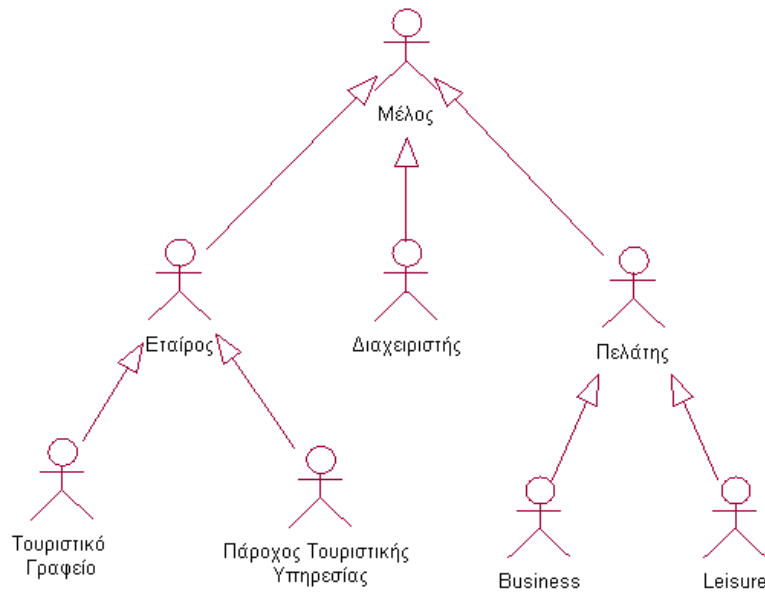
Εικόνα 23 Γενική όψη του συστήματος eKoNES

Το παραπάνω σχήμα επιχειρεί να συνοψίσει τα παραπάνω παρουσιάζοντας και μια πρώτη προσέγγιση για την λειτουργική δομή του συστήματος με αναφορά σε ένα ενδεικτικό σενάριο εφαρμογής.

Τέλος επειδή η πιλοτική έκδοση του έργου θα είναι στον τουρισμό τμήματα που παρουσιάζουμε και αφορούν το συγκεκριμένο έργο θα είναι άμεσα συνυφασμένα και προσαρμοσμένα για αυτόν.

5.2 Χρήστες του eKoNES και ρόλοι αυτών

Είναι πολύ βασικό για την κατανόηση όλου του concept του eKoNES, να καταλάβει κάποιος τους ρόλους των χρηστών που εμπεριέχονται στο eKoNES. Το eKoNES περιέχει τριών ειδών βασικούς τύπους χρηστών, όπως φαίνεται και στην παρακάτω εικόνα.



Εικόνα 24 Διαβάθμιση ρόλων χρήσης του eKoNES-Τουρισμός

Στο υψηλότερο επίπεδο η κατηγορία μέλος συνιστά ένα εγγεγραμμένο χρήστη που διατηρεί λογαριασμό και έχει διαβαθμιζόμενα δικαιώματα πρόσβασης στο περιεχόμενο του eKoNES. Ένα εγγεγραμμένο μέλος είναι είτε εταίρος, είτε πελάτης είτε διαχειριστής. Ο ρόλος του ‘διαχειριστή’ έχει ιδιαίτερη σημασία για το eKoNES και αφορά το συντονισμό της συνεργασίας μεταξύ των εμπλεκόμενων σε μια κοινότητα του eKoNES καθώς και τη γενικότερη διαχείριση του νέου περιεχομένου που προκύπτει ως αποτέλεσμα αυτής της συνεργασίας. Όπως θα δούμε και παρακάτω, ο ρόλος του ‘διαχειριστή’ εμπεριέχει σύνθετα καθήκοντα που αφορούν τόσο λειτουργικές όσο και μη-λειτουργικές απαιτήσεις.

Ο ρόλος ‘εταίρος’ εμπεριέχει δύο υπο-ρόλους – το ‘τουριστικό γραφείο’ και το ‘πάροχος τουριστικής υπηρεσίας’. Ο ρόλος ‘τουριστικό γραφείο’ διακρίνεται από το ρόλο ‘πάροχος τουριστικής υπηρεσίας’ προκειμένου να υπάρξει σαφής διαχωρισμός μεταξύ του ενδιάμεσου ή μεσάζοντα που είναι το τουριστικό γραφείο και των διαφόρων προμηθευτών τουριστικών προϊόντων στις κατηγορίες διαμονή, διατροφή, πληροφορία, ψυχαγωγία, κλπ. Τέλος, ο ρόλος ‘πελάτης’ αφορά στους τελικούς χρήστες των προϊόντων του eKoNES που μπορεί να είτε επισκέπτες σε επαγγελματική δραστηριότητα (business travelers) είτε τουρίστες (leisure travelers).

Καθένας από αυτούς εμπλέκεται με διαφορετικό τρόπο στη χρήση του συστήματος.

5.2.1 Καθήκοντα διαχειριστή eKoNEΣ

Ο διαχειριστής περιεχομένου του eKoNEΣ συνιστά ένα κεντρικό ρόλο χρήστη που έχει την οργανωτική και διοικητική κυρίως ευθύνη του περιεχομένου του eKoNEΣ. Η οργανωτική ευθύνη συνίσταται στον προσδιορισμό των κατάλληλων ροών εργασίας βάσει των οποίων νέο περιεχόμενο γίνεται διαθέσιμο και δημοσιεύεται μέσω του eKoNEΣ (π.χ. τα στάδια δημιουργίας ενός νέου πακέτου, δημοσίευση νέου περιεχομένου, κλπ). Η διοικητική ευθύνη αφορά δύο κύριες δραστηριότητες. Η πρώτη σχετίζεται με την διαχείριση του υπάρχοντος περιεχομένου, την ταξινόμησή του, την εξαγωγή χρήσιμων εμπειριών, μετα-πληροφοριών, και την κωδικοποίηση τους υπό μορφή επαναχρησιμοποιήσιμων templates. Η δεύτερη διοικητική δραστηριότητα αφορά τον συντονισμό και διαχείριση της ηλεκτρονικής κοινότητας, την δημιουργία νέων δομών όπου χρειάζεται, την σταδιακή κλιμάκωση επιμέρους ενεργειών όπως π.χ. διαφήμιση και την προσέλκυση νέων μελών.

Επομένως, ο συγκεκριμένος ρόλος είναι κεντρικός και ιδιαίτερα σημαντικός για την διαχείριση της κοινότητας και την επιτυχή ολοκλήρωση συνεργατικών ροών εργασίας όπου απαιτείται η συντονισμένη δράση και συνεργασία τόσο με επιχειρηματικούς εταίρους όσο και με τελικούς χρήστες για τη διαμόρφωση πακέτων και προσφορών

5.2.2 Καθήκοντα επιχειρηματικών εταίρων

Ο διακριτός αυτός ρόλος όπως ήδη προαναφέραμε γενικεύει τους υπο-ρόλους ‘τουριστικό γραφείο’ και ‘πάροχος τουριστικής υπηρεσίας’¹⁵. Και οι δύο υπο-ρόλοι αφορούν την προσφορά υπηρεσιών προς το πελάτη με τη διαφορά ότι ο ρόλος ‘τουριστικό γραφείο’ εμπεριέχει περισσότερο την έννοια του ενδιαμέσου μεσάζοντα (π.χ. ένα γραφείο οργάνωσης εκδρομών) ενώ ο ρόλος του ‘παρόχου τουριστικής υπηρεσίας’ εμπεριέχει την έννοια της προμήθειας προϊόντων και υπηρεσιών που συναθροίζονται για να δημιουργήσουν νέα πακέτα. Ο ρόλος του ‘παρόχου τουριστικής υπηρεσίας’ μπορεί να αναλαμβάνεται από ένα οργανισμό τοπικής αυτοδιοίκησης (π.χ. ένα δήμο ή μια νομαρχία) με στόχο την παροχή ενημερωτικών δελτίων τύπου προς τα μέλη του eKoNEΣ, από ένα Κέντρο Εξυπηρέτησης Πολιτών με στόχο την παροχή υπηρεσιών ηλεκτρονικής διακυβέρνησης σε τοπικό επίπεδο ή από μια επιχείρηση όπως μια ξενοδοχειακή μονάδα, ένα camping, μία ταβέρνα, ένα

¹⁵ Οι δύο παραπάνω υπο-ρόλοι απορρέουν από τον κλάδο του τουρισμού. Σε άλλο κλάδο της τοπικής βιομηχανίας θα υπήρχαν διαφορετικοί υπο-ρόλοι.

γραφείο ενοικίασεως αυτοκινήτων κ.α. Ο επιχειρηματικός εταίρος (ή business partner) θα μπορεί να προσφέρει τις υπηρεσίες του και να προβάλλεται μεμονωμένα μέσα από την κοινότητα, παρουσιάζοντας το εταιρικό του προφίλ και πληροφορίες που ενδιαφέρουν τους τελικούς χρήστες ή πελάτες (βλέπε παρακάτω). Ο ουσιαστικός ρόλος που καλείται να αναλάβει ο Business Partner στην κοινότητα, πέραν της ηλεκτρονικής παρουσίασης των υπηρεσιών του, είναι η ενεργή συμμετοχή και η συνεργασία με άλλους εταίρους και το διαχειριστή προκειμένου μέσα από ένα συνεργατικό περιβάλλον (την κοινότητα eKONEΣ-Τουρισμός) να προκύπτουν νέα προϊόντα και υπηρεσίες τα οποία θα καλύπτουν διαρκείς ή πρόσκαιρες ανάγκες της αγοράς τουρισμού.

5.2.3 Ο τελικός χρήστης

Οι τελικοί χρήστες ή πελάτες του eKoNEΣ-Τουρισμός είναι μεμονωμένοι πολίτες ή ομάδες πολιτών (π.χ., λέσχες, όμιλοι κλπ) με ποικίλα ενδιαφέροντα που μπορεί να συνδυάζονται με επαγγελματικές υποχρεώσεις ή διασκέδαση. Η χρήση του eKoNEΣ από χρήστες της κατηγορίας αυτής μπορεί να απορρέει είτε από το γεγονός ότι σκοπεύουν να επισκεφθούν κάποιο συγκεκριμένο τουριστικό προορισμό και επιθυμούν να συνδυάσουν την παραμονή τους με άλλες συμπληρωματικές δραστηριότητες (π.χ. επισκέψεις και περιήγηση σε επιλεγμένους χώρους), είτε από την ανάγκη πληρέστερης ενημέρωσης αναφορικά με πεπραγμένα στο ευρύτερο θεματικό περιβάλλον που καλύπτει το eKoNEΣ-Τουρισμός (π.χ. ενημέρωση για το πρόγραμμα εκδηλώσεων ενός δήμου ή οργανισμού), είτε τέλος από επαγγελματικές υποχρεώσεις (π.χ. οργάνωση ενός συνεδρίου). Σε κάθε περίπτωση οι τελικοί χρήστες μπορεί να έχουν περιοδικό ή μόνιμο ενδιαφέρον στο περιεχόμενο του eKoNEΣ-Τουρισμός.

5.3 Ποια η θέση του Σύγχρονου Collaboration στο eKoNEΣ?

Σημαντική θέση όσον αφορά την προαγωγή της λειτουργικότητας του ρόλου του συστήματος αλλά και της τεχνολογικής υποδομής του έργου αποτελεί το eKoNEΣ Συνεργασία, που μέσω των τεχνολογικών υποδομών που τμήμα των οποίων υλοποιήθηκε για τις ανάγκες της συγκεκριμένης πτυχιακής, προσφέροντας υπηρεσίες

επικοινωνίας σε πραγματικό χρόνο, ενισχύοντας στο μεγαλύτερο βαθμό τη συνεργατικότητα ανάμεσα σε μέλη και συντονιστές.

5.3.1 Προϋποθέσεις για το χαρακτηρισμό ενός συστήματος ως Framework

Τρία χαρακτηριστικά γνωρίσματα πρέπει να έχει ένα “σύστημα”, ώστε να μπορεί να χαρακτηριστεί ως Framework. Αυτά αφορούν το Abstraction όσον αφορά τη σχεδίαση, δηλαδή να μπορεί να καλύψει όσο το δυνατόν μεγαλύτερο πλήθος use cases πέρα από τα συγκεκριμένα για τα οποία έχει φτιαχτεί ώστε να καλύψει. Δηλαδή να είναι γενικής χρήσης. Σημαντικό ρόλο παίζει το να είναι αποτελεσματικό και προσαρμόσιμο, δηλαδή να μπορεί να αλλάξει εύκολα και με μικρές επεκτάσεις(expandability) να μπορεί να υποστηρίξει και να καλύψει εύκολα και γρήγορα νέες(flexibility, plasticity).

Κλείνοντας πρέπει να πούμε ότι εκτός από το abstraction στη σχεδίαση υπάρχει και το abstraction στην υλοποίηση. Προφανώς λαμβάνοντας υπόψιν μόνο το δεύτερο δεν μπορείς να χαρακτηρίσεις ένα σύστημα ως framework. Συνήθως όμως ένα καλό framework συνοδεύεται από καλή σχεδίαση.

5.3.2 Πλεονεκτήματα και Μειονεκτήματα ενός Framework

Ποιο άλλο θα μπορούσε να είναι το σημαντικότερο πλεονέκτημα εκτός από αυτό της επαναχρησιμοποίησης κώδικα; Και πόσο μάλλον όλος ο κώδικας αυτός αποτελεί ολόκληρο framework γεγονός που συνεπάγεται ότι έχει τα πλεονεκτήματα εις πολλαπλού απ’ ότι έχει για μια οποιαδήποτε άλλη κλάση-Object, small Objects collection. Κάτι τέτοιο σημαίνει ελάττωση χρόνου παραγωγής, μείωση κόστους και φυσικά επαναχρησιμοποίηση κώδικα: “*write once use anywhere*”.

Δεν χρειάζεται βέβαια να είναι κάποιος ειδήμων του χώρου για να καταλάβει το ότι αυτή η γενικότητα και αυτό το abstraction έχει και κάποια όρια ανάλογα με τα όρια της φαντασίας και της διορατικότητας των developers και των specific system requirements που έχουν τεθεί a priori. Το abstraction είναι ένα πολύπλοκο κομμάτι που απαιτεί πολύ σκέψη εμπειρία και διορατικότητα.

Εν τέλει σε αυτό το επίπεδο η καλύτερη λύση, δεν χαρακτηρίζεται μόνο από το αν και κατά πόσο λειτουργεί σωστά, αλλά και από το χρόνο και το κατά πόσο σε βάθος χρόνου μπορεί να συνεχίσει να υπάρχει το ίδιο framework εξυπηρετώντας τις ανάγκες για τις οποίες φτιάχτηκε με επιτυχία συνέπεια και ακρίβεια. Άρα ακόμη και το abstraction δεν ξεφεύγει από τη σοφή ρήση: “χρόνος είναι ίσον: ο καλύτερος συνεπέστερος και σωστότερος κριτής των πάντων”. Οπότε αφήνουμε στην κρίση του αναγνώστη να κρίνει την επιτυχία της αρχιτεκτονικής υλοποίησης

6 Υλοποίηση του eKoNES Collaborative Framework

Το μοντέλο που επιλέξαμε ακολουθεί το μοντέλο της partially replicated αρχιτεκτονικής, με το μοντέλο και τον controller του συστήματος να βρίσκονται αποκεντρωμένα. Τα client μηχανήματα (αυτά του διαχειριστή και των επιχειρηματικών εταίρων δηλαδή) είναι dummy views, τα οποία έχουν ως ρόλο την εμφάνιση με συγκεκριμένο γραφικό τρόπο της πληροφορίας όπως αυτή ορίζεται μέσα από τα δεδομένα της βάσης δεδομένων του eKoNES.

Συγκεκριμένα ως μοντέλο(persistent storage) της εφαρμογής επιλέξαμε τη mysql, μια ιδιαίτερα ευέλικτη γρήγορη και πολυχρησιμοποιημένη, multithreaded σχεσιακή βάση δεδομένων, η οποία είναι Open source ενώ παράλληλα είναι διαθέσιμη για πλήθος λειτουργικών συστημάτων. Τα υπόλοιπα τμήματα της υλοποίησης που αφορούν το διαχειριστή και τους εταίρους(participants) και τα οποία θα περιγραφούν αναλυτικά στη συνέχεια, είναι και αυτά, όπως και ο collaboration server, υλοποιημένα με τη γλώσσα προγραμματισμού java, ενώ είναι fully J2SE 6.0 compatible.

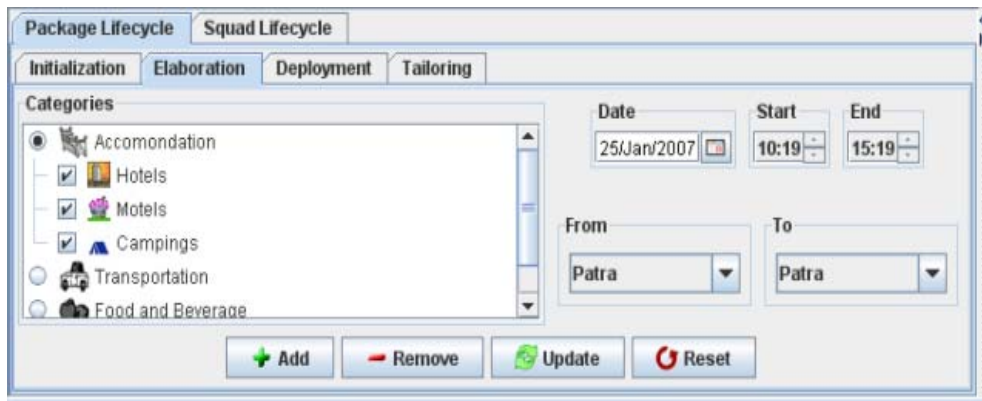
6.1 Το Σενάριο

Το σενάριο που θα μελετήσουμε αφορά τη δημιουργία και εξέλιξη ενός τουριστικού πακέτου, το οποίο δημιουργήθηκε από το διαχειριστή του eKoNES, λόγω εξωτερικών ερεθισμάτων που έλαβε μέσω του portal του eKoNES(όπως φαίνεται και στην παρακάτω εικόνα).

The screenshot shows the eKoNES forum interface. At the top, there is a search bar and a 'Sign In' link. Below the navigation menu, there are tabs for 'Categories', 'Recent Posts', and 'Statistics'. The main content area displays a thread titled 'Δημιουργία πακέτου για εκδρομή στη Λάρισα.' (Creating a package for a trip to Larissa). The thread starts with a post by 'Anonymous' at 5/17/07 5:40 PM, which says: 'Τα χαιρετίσματά μου στην κοινότητα. Θα ήθελα να εκφράσω το ενδιαφέρον μου για τη δημιουργία ενός νέου τουριστικού πακέτου στο νομό Λάρισας.' (Greetings to the community. I would like to express my interest in the creation of a new tourism package in the Larissa region). This is followed by two replies from users with question mark avatars, both at 5/17/07 5:41 PM. The first reply says: 'Συμφωνώ. Και εγώ βρίσκω την ιδέα πολύ καλη!' (I agree. And I find the idea very good!). The second reply says: 'Πραγματικά πολύ ενδιαφέρουσα ιδέα.' (Really very interesting idea.).

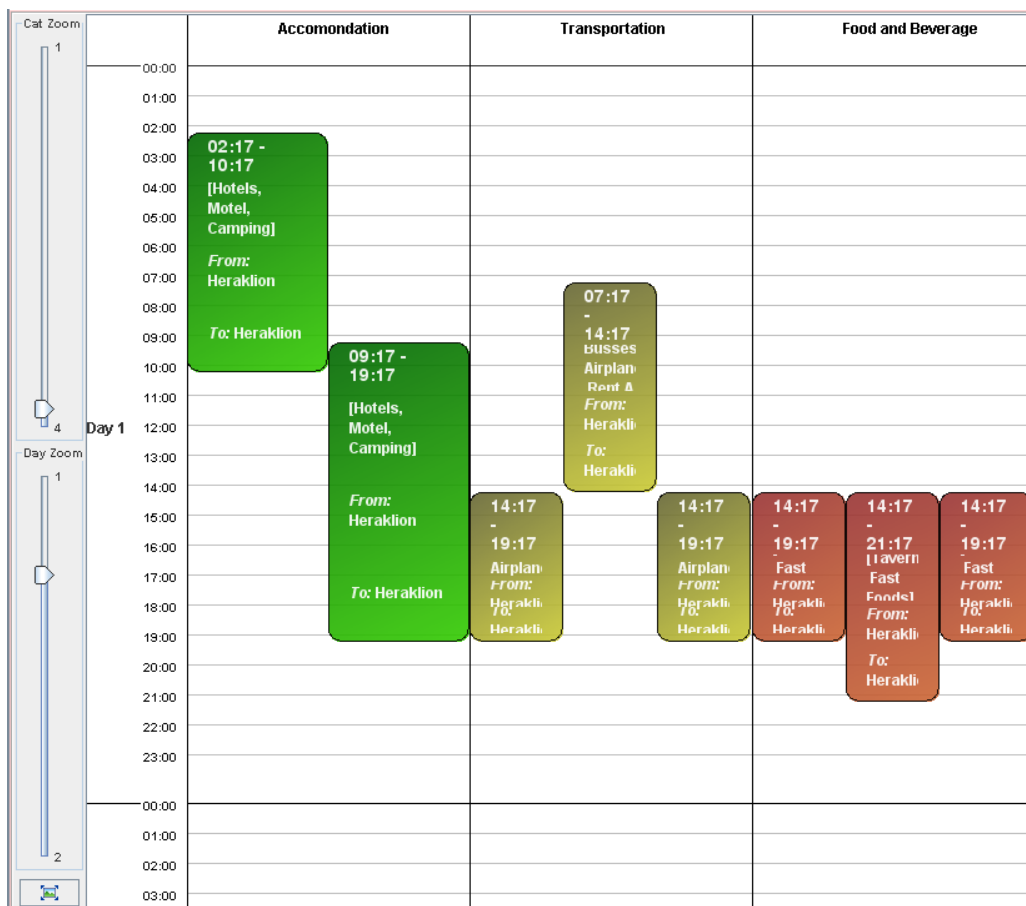
Εικόνα 25 Εκδήλωση ενδιαφέροντος για τη δημιουργία νέου τουριστικού πακέτου

Μετά το ερέθισμα που λαμβάνει ο διαχειριστής του eKoNES (eKoNES administrator), αποφασίζει αφού βλέπει έντονο ενδιαφέρον, τη δημιουργία ενός νέου τουριστικού πακέτου το οποίο περιλαμβάνει εκδρομή στο νομό Λάρισας. Πρώτη κίνηση του διαχειριστή είναι να μπει στην κεντρική κονσόλα διαχείρισης πακέτων και να δημιουργήσει ένα αφηρημένο κορμό του πακέτου, με τη βοήθεια ενός εργαλείου που επιτρέπει συνεργασία μεταξύ των μελών της κοινότητας (collaboratorium), εισάγοντας βασικές αρχικές παραμέτρους (στάδιο αρχικοποίησης πακέτου). Συγκεκριμένα εισάγει το όνομα του πακέτου, καθώς επίσης την αρχική και τελική ημερομηνία αυτού προσθέτοντας μια βασική περιγραφή.



Εικόνα 26 Εισαγωγή δραστηριοτήτων(activities) στο πακέτο Λάρισα

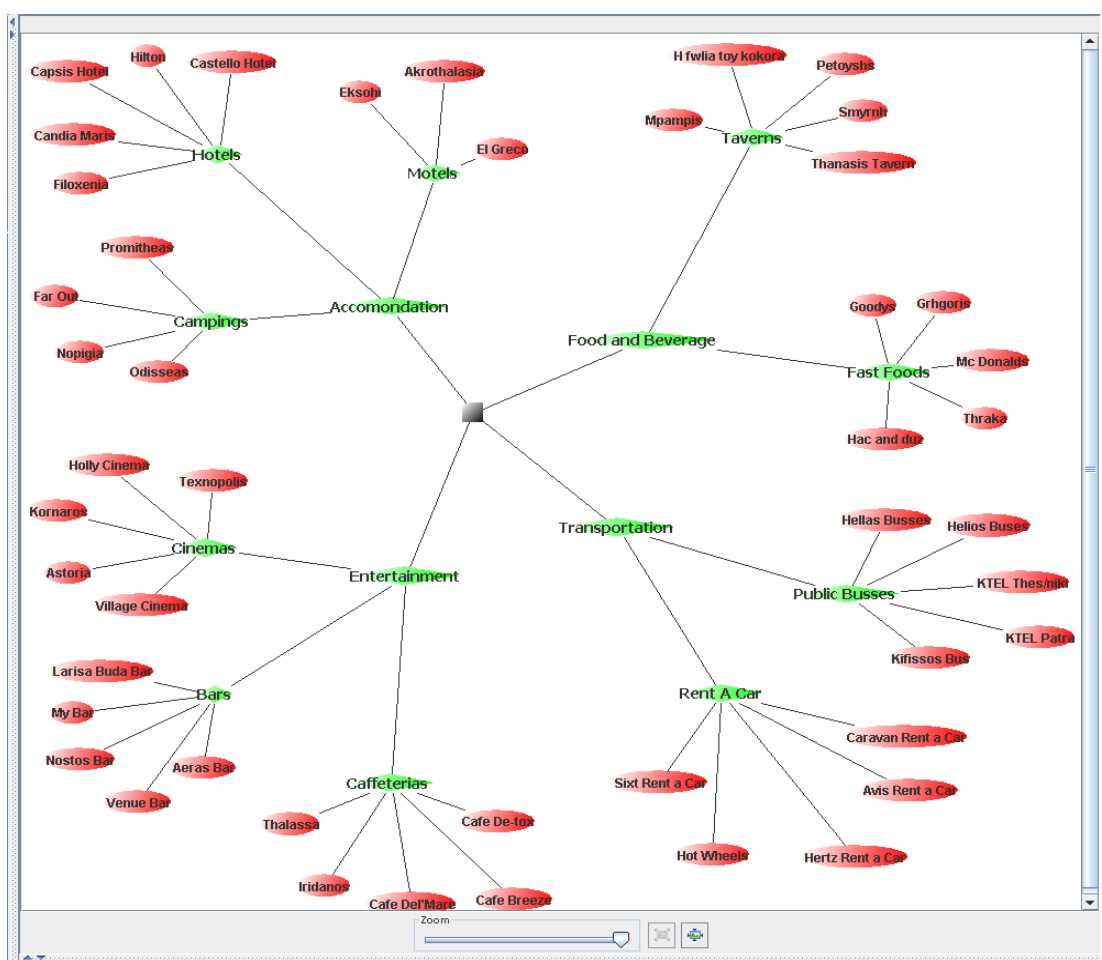
Στη συνέχεια ο διαχειριστής προχωράει στη διαμόρφωση της δομής του πακέτου εισάγοντας σε αυτό δραστηριότητες(activities) μέσω του collaboratorium. Οι δραστηριότητες αυτές εμφανίζονται με γραφικό τρόπο στο γράφο που φαίνεται στην παρακάτω εικόνα(γράφος του Elaboration), δίνοντας στο διαχειριστή τη δυνατότητα επισκόπησης της δομής του προς διαμόρφωση πακέτου.



Εικόνα 27 Γράφος του Elaboration

Η παραπάνω διαδικασία εισαγωγής δραστηριοτήτων ουσιαστικά καλεί τους εμπορικού εταίρους (που ανήκουν στις κατηγορίες που επέλεξα) να συμμετάσχουν στην ομάδα που θα υλοποιήσει ενεργά το νέο πακέτο.

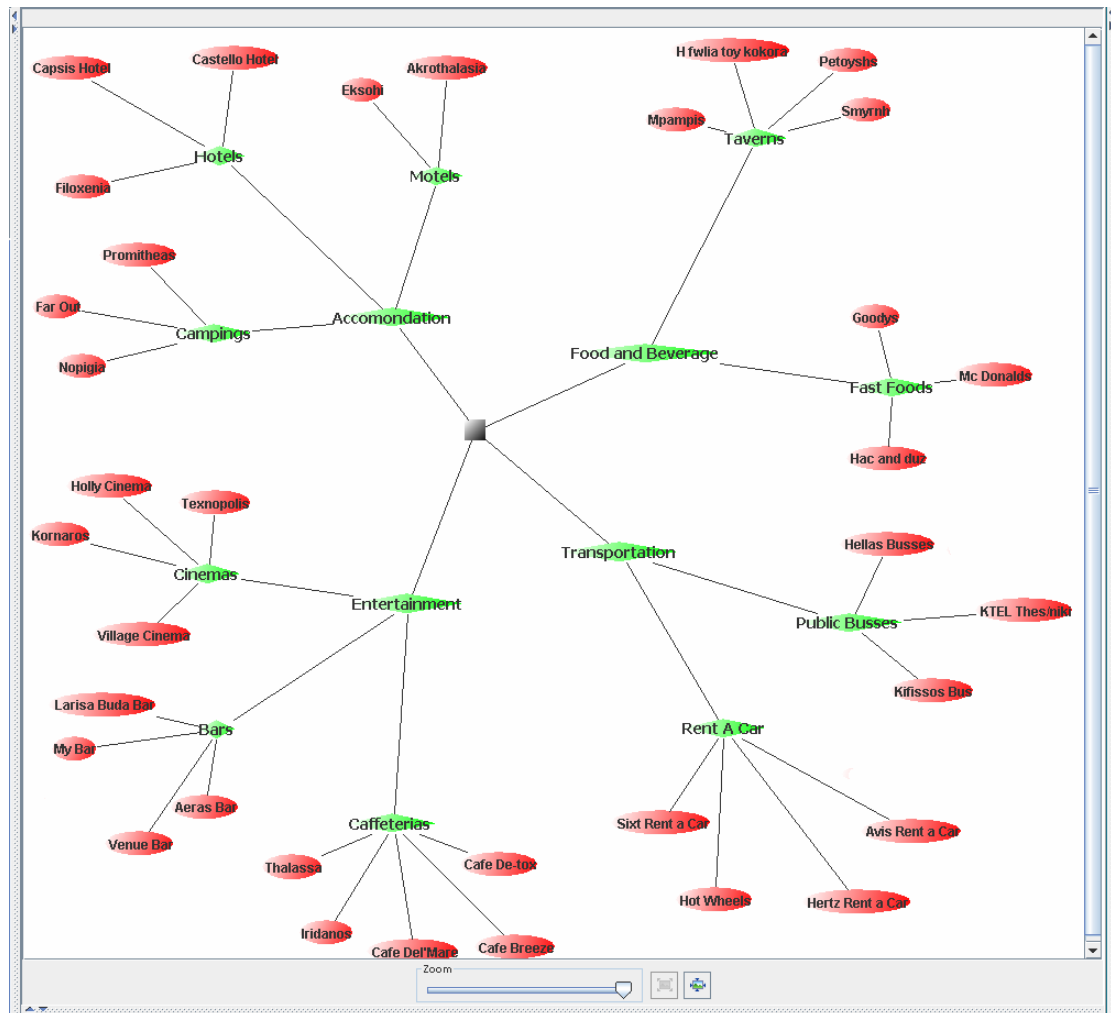
Η κοινότητα που συστήνεται έχει τη γραφική αναπαράσταση που φαίνεται στην παρακάτω εικόνα.



Εικόνα 28 Μορφή της υπό σύσταση κοινότητας του πακέτου Λάρισα

Στη συνέχεια η ομάδα που μετέχει στη διαμόρφωση του πακέτου, μέσω ασύγχρονης συνεργασίας (με τη βοήθεια των εργαλείων που προσφέρει το collaboratorium), οδηγούνται στη διαμόρφωση μιας πιο συγκεκριμένης δομής της ομάδας(squad) όσον αφορά την τελική εκδοχή του πακέτου Λάρισα. Μέσω των μηνυμάτων που ανταλλάσσονται στο message board(commitment messages, κλπ.)

οριστικοποιείται η δομή της ομάδας(squad) και το πακέτο παίρνει την τελική του μορφή όπως αυτό φαίνεται στο παράδειγμα της εικόνας 29.

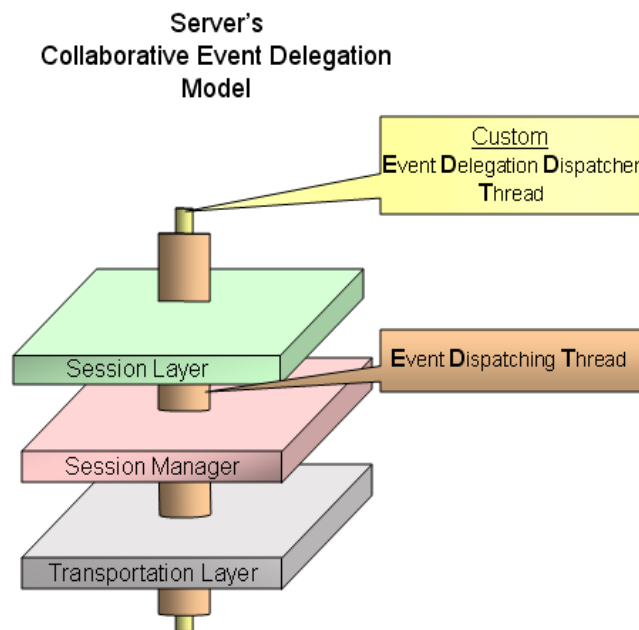


Εικόνα 29 Οριστική μορφή της κοινότητας του πακέτου Λάρισα

Για την υποστήριξη ορισμένων καθηκόντων και δράσεων (actions), που εκτελούνται ασύγχρονα, δημιουργείται η ανάγκη ύπαρξης ενός συνεργατικού εργαλείου-Framework σύγχρονης συνεργασίας, το οποίο θα βοηθήσει τα μέλη της κοινότητας στη λήψη γρήγορων αποφάσεων δίνοντας τους δυνατότητες που η ασύγχρονη επικοινωνία ως τεχνολογία δεν μπορεί να προσφέρει, ή τουλάχιστον δεν μπορεί να προσφέρει αποτελεσματικά.

6.2 O Collaboration Server

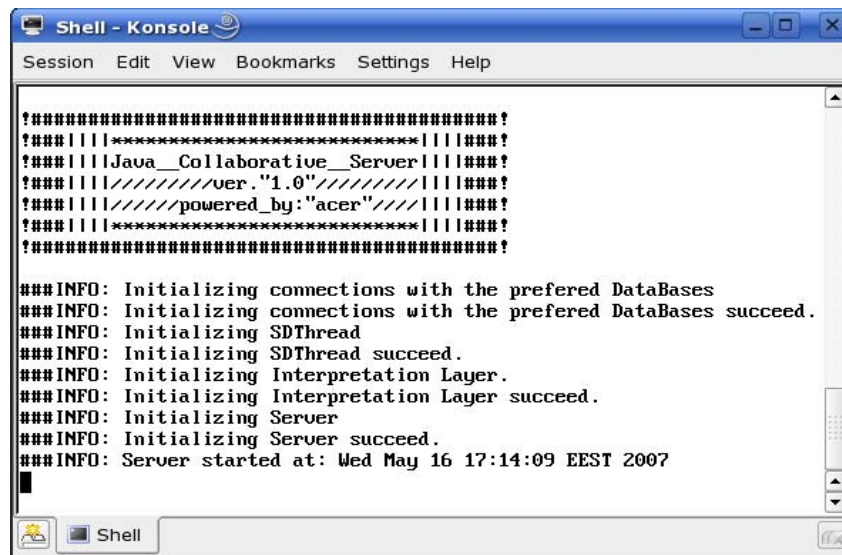
Ο collaboration server δομείται σε τρία βασικά επίπεδα, το επίπεδο μεταφοράς(Transportation Layer), το επίπεδο διαχείρισης συνόδου(session Management Layer) και το επίπεδο συνόδου(Session Layer). Αυτή η 3-επίπεδη(3-tired) αρχιτεκτονική έχει ως αποτέλεσμα τη δημιουργία ενός αρκετά ευέλικτου συστήματος, εύκολα επεκτάσιμου – αναβαθμίσιμου, με κάθε επίπεδο να έχει τελείως ξεκάθαρο και διακριτό ρόλο. Αυτό σημαίνει ότι σε περίπτωση που θελήσουμε να αλλάξουμε το API, με το οποίο “μιλάμε” με το δίκτυο, επειδή ίσως βγει σε μέλλοντα χρόνο κάποιο νεότερο και πιο εξελιγμένο API (π.χ. το SOAP2) από αυτό που τώρα χρησιμοποιούμε, θα μπορούμε πολύ εύκολα και γρήγορα να κάνουμε την αλλαγή, χωρίς να χρειαστεί να τροποποιήσουμε κάποιο σημείο οποιουδήποτε άλλου επιπέδου της εφαρμογής.



Εικόνα 30 Τα επίπεδα του Collaboration Server

Για την υλοποίηση του κατώτερου επιπέδου (μεταφοράς) επελέγη το Socket API της Java. Αυτό έγινε διότι τα sockets δεν εμπεριέχουν την πρόσθετη πολυπλοκότητα που ενσωματώνουν πιο πολύπλοκα API's, όπως π.χ. το SOAP, με αποτέλεσμα να γλιτώσουμε κατανάλωση σε resources ενώ παράλληλα είχαμε το πλεονέκτημα του ότι ξεκινώντας να χτίζεις από τόσο χαμηλό επίπεδο σου δίνεται η δυνατότητα να διαμορφώσεις και να κάνεις optimize το σύστημα στις προσωπικές σου ανάγκες και

απαιτήσεις. Η επιλογή αυτή βεβαίως και δεν μας περιορίζει στο μέλλον να χρησιμοποιήσουμε οποιαδήποτε άλλη τεχνολογία, αφού κάθε επίπεδο είναι δομημένο έτσι ώστε να κρύβει τον τρόπο με τον οποίο υλοποιεί τις υπηρεσίες που προσφέρει στα υπόλοιπα επίπεδα.

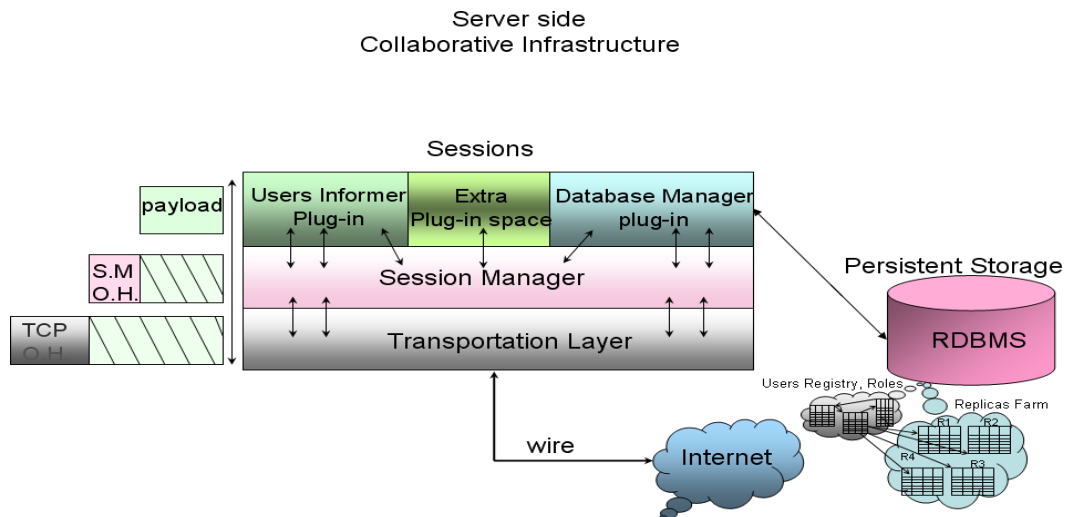


```
#####  
#####  
#####Java_Collaborative_Server#####  
#####ver."1.0"#####  
#####powered_by:"acer"#####  
#####  
#####  
###INFO: Initializing connections with the preferred DataBases  
###INFO: Initializing connections with the preferred DataBases succeed.  
###INFO: Initializing SDThread  
###INFO: Initializing SDThread succeed.  
###INFO: Initializing Interpretation Layer.  
###INFO: Initializing Interpretation Layer succeed.  
###INFO: Initializing Server  
###INFO: Initializing Server succeed.  
###INFO: Server started at: Wed May 16 17:14:09 EEST 2007  
█
```

Εικόνα 31 Startup του Java Collaborative Server(JColler), σε περιβάλλον Linux

Με το που ξεκινάει ο collaboration server συνδέεται με τις βάσεις δεδομένων που έχουν οριστεί μέσω ενός xml configuration file. Στη συνέχεια γίνεται το initialization αυτού που ονομάζουμε “Server Dispatching Thread”, και το οποίο ακολουθεί την ίδια λογική και φιλοσοφία με τον AWT Event Multicaster. Η υλοποίηση που αφορά αυτό το κομμάτι κάνει χρήση πολλών API’s(ένα από τα οποία είναι και το reflection API), με αποτέλεσμα τη δημιουργία ενός πολύ στιβαρού Event Delegation Model πάνω στο οποίο βασίζεται όλη η αρχιτεκτονική.

Στη συνέχεια γίνεται αρχικοποίηση του interpretation Layer, που είναι ουσιαστικά το επίπεδο, το οποίο μεταφράζει μερικώς τα εισερχόμενα xml μηνύματα, τα οποία και προωθεί στα υψηλότερα επίπεδα ανάλογα με τη λειτουργία που πρέπει-ζητείται να εκτελεστεί. Η δρομολόγηση γίνεται διαβάζοντας το pi (processing instruction) του xml μηνύματος που λαμβάνει, ενώ αφού το διερμηνεύσει επιτυχώς, το προωθεί μέσω του custom Event Delegation Model στους ανάλογους listeners.



Εικόνα 32 Η δομή του Collaboration Server(JColler)

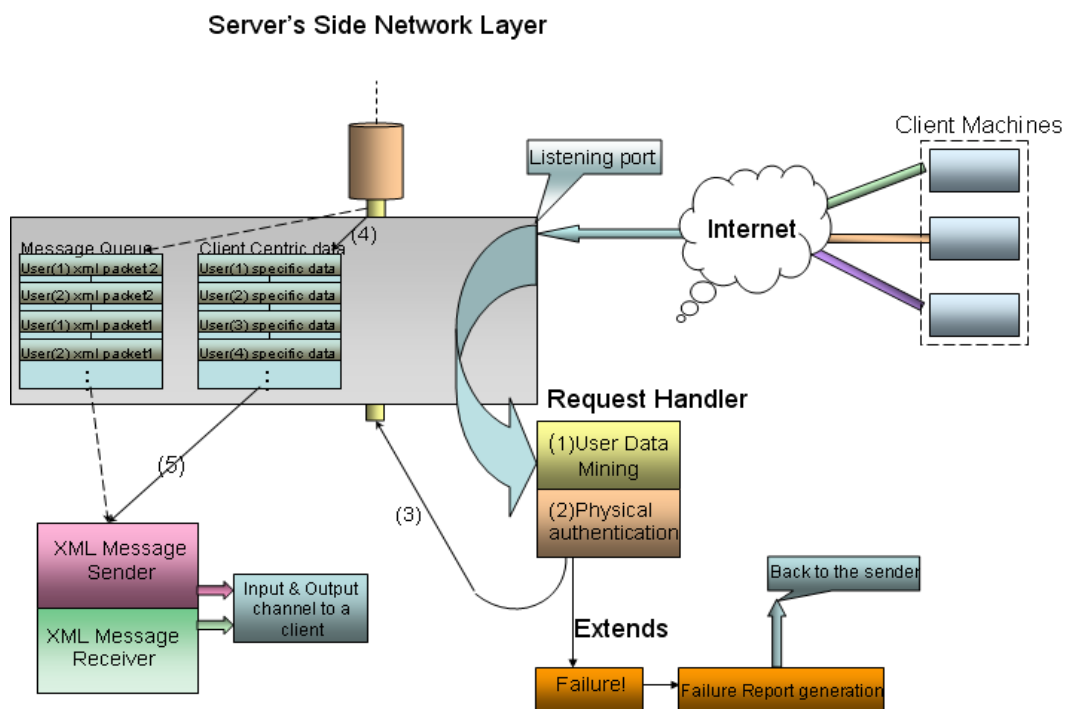
Το συντακτικό των pi's καθώς και της υπόλοιπης δομής των xml μηνυμάτων είναι σαφές για κάθε περίπτωση, και ορίζεται με τη βοήθεια του XMLSchema (μια πιο εξελιγμένη μορφή του DTD). Ουσιαστικό πλεονέκτημα των xml μηνυμάτων είναι ότι δημιουργούμε ένα επίπεδο αφαίρεσης, που μας απομονώνει και ελευθερώνει από λεπτομέρειες που έχουν να κάνουν με τη γλώσσα προγραμματισμού που είναι υλοποιημένη η κάθε τερματική εφαρμογή(dummy view). Αυτό σημαίνει ότι θα μπορούσε κάλλιστα ένας δυνητικός χρήστης του οποίου η εφαρμογή είναι γραμμένη σε C++ ή .NET, να μπορεί να επικοινωνήσει και να λάβει ενεργό ρόλο σε ένα collaborative session (εφόσον γνωρίζει και μιλάει σωστά το συντακτικό του JColler) χωρίς καμία αλλαγή στη λογική του συστήματος. Αυτό μας προσφέρει εξαιρετικά μεγάλη ευελιξία όσον αφορά το συγχρονισμό και τη συμμετοχή σε ένα collaborative session και άλλων μηχανημάτων εκτός από desktop, όπως π.χ. pda's ή palmtops, κα. Πρέπει βέβαια να τονίσουμε πως δεν αρκεί μόνο η xml για να επιτύχεις συνεργασία μεταξύ διαφορετικών υλοποιήσεων (εννοώντας σε διαφορετική γλώσσα προγραμματισμού), πόσο μάλλον σε τελείως διαφορετική συσκευή. Αφού τονίσουμε ότι τέτοια θέματα είναι αρκετά εξειδικευμένα και χαίρουν μεγάλης ερευνητικής δραστηριότητας, πρέπει να τονίσουμε ότι παρά το γεγονός ότι δεν ήταν στόχος της πτυχιακής αυτής το synchronous collaboration μεταξύ διαφορετικών συσκευών, παραταύτα δόθηκε μεγάλη σημασία στη δόμηση όλου του συστήματος ώστε να μπορεί σε μελλοντική του έκδοση να υποστηρίξει τόσο σύνθετα και εξελιγμένα χαρακτηριστικά. Η ουσία για να πετύχεις κάτι τέτοιο βρίσκεται στο να περιγράψεις

την πληροφορία με σωστό και αφηρημένο τρόπο ανεξαρτήτως γλώσσας προγραμματισμού.

Σημαντικό σημείο είναι να τονίσουμε πως για κάθε participant κρατιέται ένα μόνο κανάλι επικοινωνίας(back & forth), ανεξαρτήτως του αριθμού των collaborative sessions στα οποία μετέχει.

Επίσης σημαντικό σημείο είναι ένα custom implementation μιας FIFO ουράς (Queue), ρόλος της οποίας είναι να στοιβάζει μηνύματα που είναι προς αποστολή προς κάθε client. Συγκεκριμένα για κάθε client Thread γίνεται instantiated και μια ουρά. Αυτή την ουρά διαβάζει συνεχώς ο server για μηνύματα που είναι να αποσταλούν προς κάθε client χωριστά.

Για την επικοινωνία με όλα τα views που μετέχουν σε ένα collaborative Object, ο server χρησιμοποιεί ένα ταχύτατο τρόπο δημιουργίας xml μηνυμάτων JColler-Compliant.



Εικόνα 33 Βασική δομή του Network Layer του Collaboration Server

Στην παραπάνω εικόνα παρατηρούμε τη ροή των γεγονότων κατά τη διαδικασία εγγραφής ενός partner στο σύστημα. Αρχικά τα client μηχανήματα συνδέονται μέσω του listening port του JColler, ο οποίος με τη σειρά του κάνει mine τα login data διαβάζοντας το xml request(το συντακτικό του οποίου φαίνεται στην παρακάτω

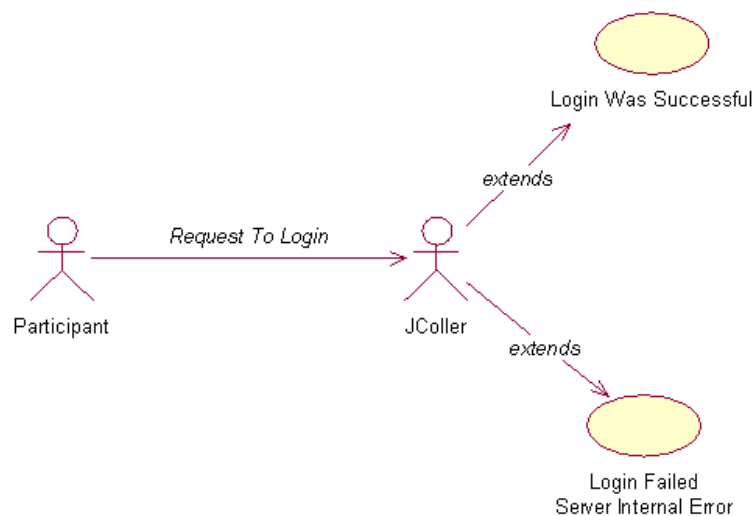
εικόνα) με τη βοήθεια του XPath API και εκτελεί το Physical authentication. Σε περίπτωση που το authentication είναι επιτυχές αποπέμπει ένα μήνυμα στο SDT (Server Dispatching Thread), το οποίο έχει ως αποτέλεσμα την εγγραφή του συγκεκριμένου χρήστη στο σύστημα και το άνοιγμα ενός καναλιού επικοινωνίας για send και receive data μέσω των κλάσεων XML Message Sender και Receiver αντίστοιχα. Σε περίπτωση βέβαια αποτυχίας ενημερώνεται η εφαρμογή του χρήστη με το κατάλληλο μήνυμα αποτυχίας, ενώ είναι προφανές ότι το κανάλι μέσω του οποίου έγινε η επικοινωνία κλείνει αμέσως.

```

<!-- REQUEST MESSAGE-->
- <ekones ver="1.0">
  <pi>request-login</pi>
  <parts>1</parts>
  - <userData>
    <userName>57</userName>
    <password/>
  </userData>
</ekones>

```

Εικόνα 34 Request Login xml Message



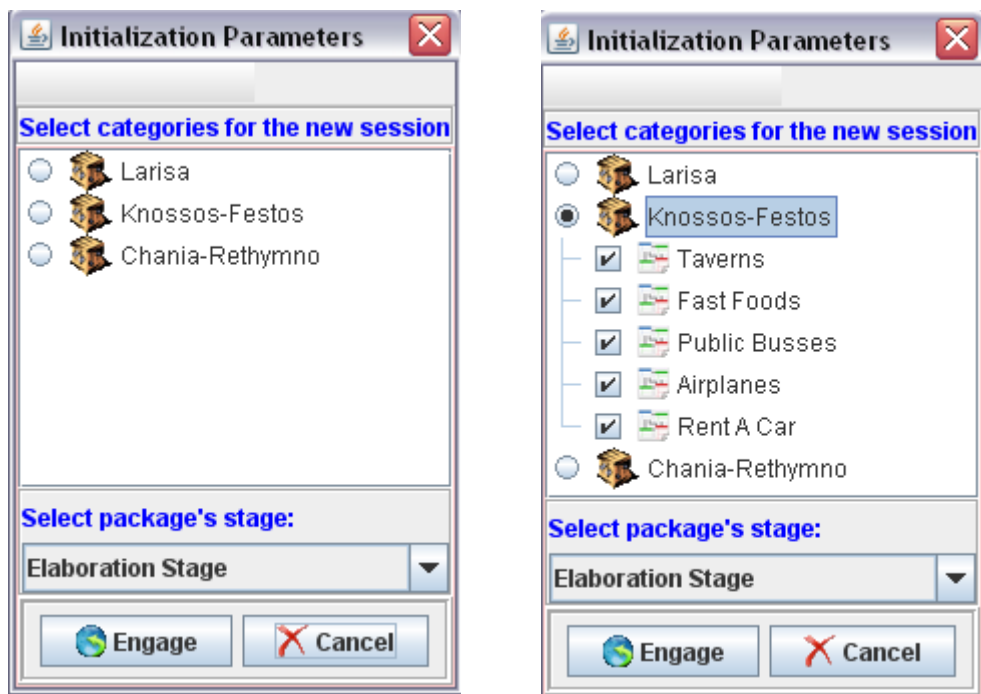
Εικόνα 35 UML use case για το request Login ενός χρήστη στο JColler

6.3 Ο διαχειριστής

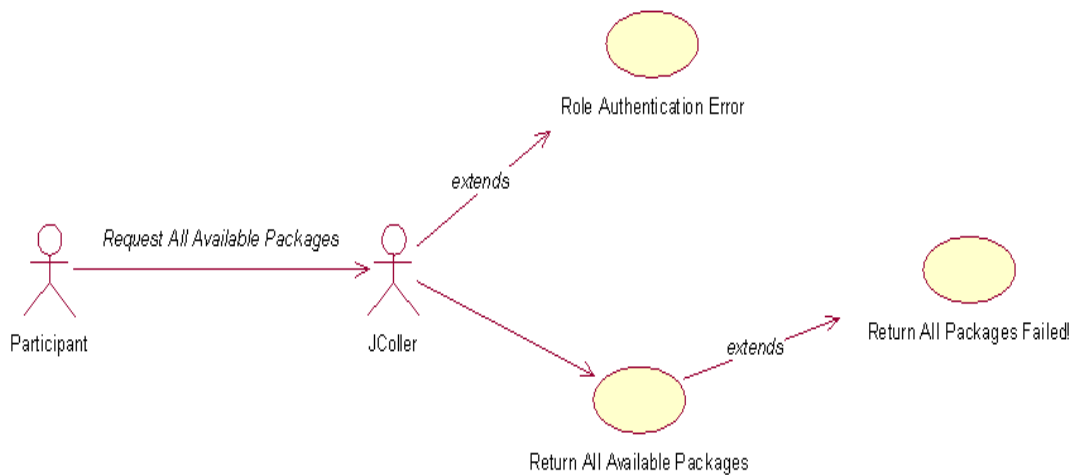
Ο ρόλος του Collaboration Administrator εστιάζεται στη δημιουργία μιας νέας σύγχρονης συνεργατικής συνόδου, στην καταστροφή της, στην τροποποίηση

βασικών παραμέτρων που αφορούν τα replicated Objects, καθώς επίσης και στην επιλογή βασικών παραμέτρων που αφορούν την προς δημιουργία σύνοδο.

Συγκεκριμένα κατά τη δημιουργία ενός νέου session, ο διαχειριστής καλείται να επιλέξει τις κατηγορίες που θέλει να μετέχουν στη συγκεκριμένη σύνοδο επιλέγοντας από τις συνολικά διαθέσιμες κατηγορίες που μετέχουν σε ένα πακέτο. Τέλος πρέπει να δηλώσει σε ποιο στάδιο του πακέτου θέλει να δημιουργήσει το νέο session, ενέργεια που ορίζει αυτομάτως το ποια συνεργατικά αντικείμενα πρέπει να γίνουν replicated σε κάθε μηχανήμα. Στο στάδιο του Elaboration π.χ. ο γράφος που πρέπει να γίνει replicated είναι αυτός που φαίνεται στην εικόνα 27.

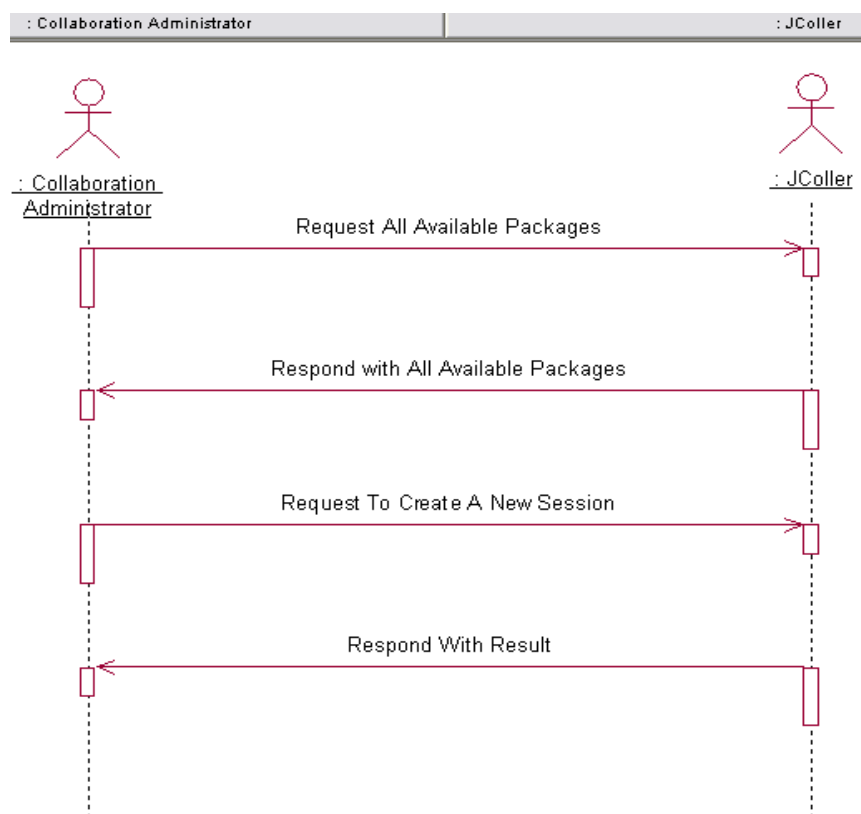


Εικόνα 36 Δημιουργία ενός νέου Collaborative Session



Εικόνα 37 UML διάγραμμα που αφορά τη δημιουργία ενός νέου Collaborative Session

Στο παραπάνω διάγραμμα περιγράφονται όλα τα πιθανά use cases κατά τη διαδικασία δημιουργίας μιας συνεργατικής συνόδου.

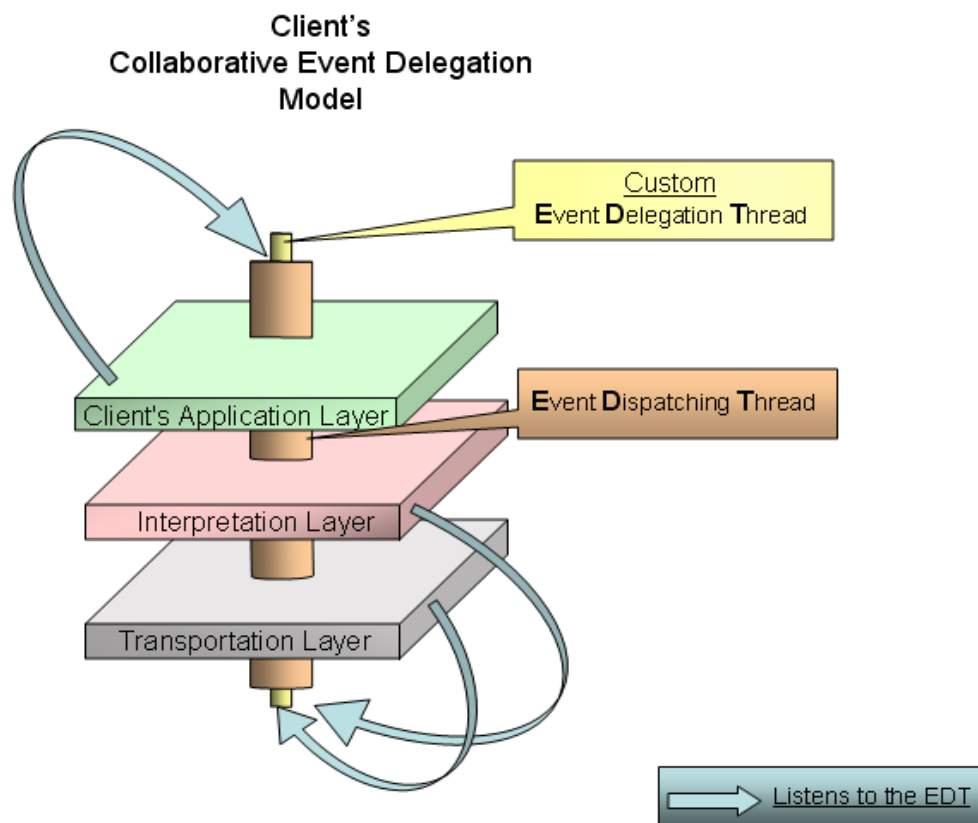


Εικόνα 38 Sequence diagram Δημιουργίας ενός νέου Collaborative Session

Η ροή των μηνυμάτων για τη δημιουργία ενός νέου session όπως φαίνεται και στο παραπάνω διάγραμμα καταστάσεων ακολουθεί την εξής αλληλουχία: αρχικά ο

διαχειριστής αιτεί για την ανάκτηση όλων των τρεχόντων πακέτων για τα οποία μπορεί να ξεκινήσει μια νέα συνεργατική σύνοδο, ενώ στη συνέχεια ο JColler απαντά με όλα τα διαθέσιμα πακέτα (Βλέπε εικόνα 34). Στη συνέχεια ο διαχειριστής αφού επιλέξει το πακέτο και τις κατηγορίες που θέλει να μετέχουν στο νέο session, στέλνει ένα xml request στο server αιτώντας για τη δημιουργία του με τις παραμέτρους που επέλεξε. Τέλος ο JColler απαντά αναλόγως στο request με το αν ήταν επιτυχές ή αν συνέβη κάποιο λάθος κατά τη διαδικασία δημιουργίας του.

Όπως ο collaboration server, έτσι και η εφαρμογή του διαχειριστή είναι χωρισμένη σε δύο επίπεδα, το επίπεδο μεταφοράς, το επίπεδο διεργασίας και το επίπεδο εφαρμογής. Και εδώ όπως και στο server υπάρχει ο ίδιος μηχανισμός αποστολής γεγονότων(Event Delegation Thread), όπως φαίνεται και στο παρακάτω σχήμα.



Εικόνα 39 Βασική δομή της αρχιτεκτονικής δόμησης της εφαρμογής του διαχειριστή

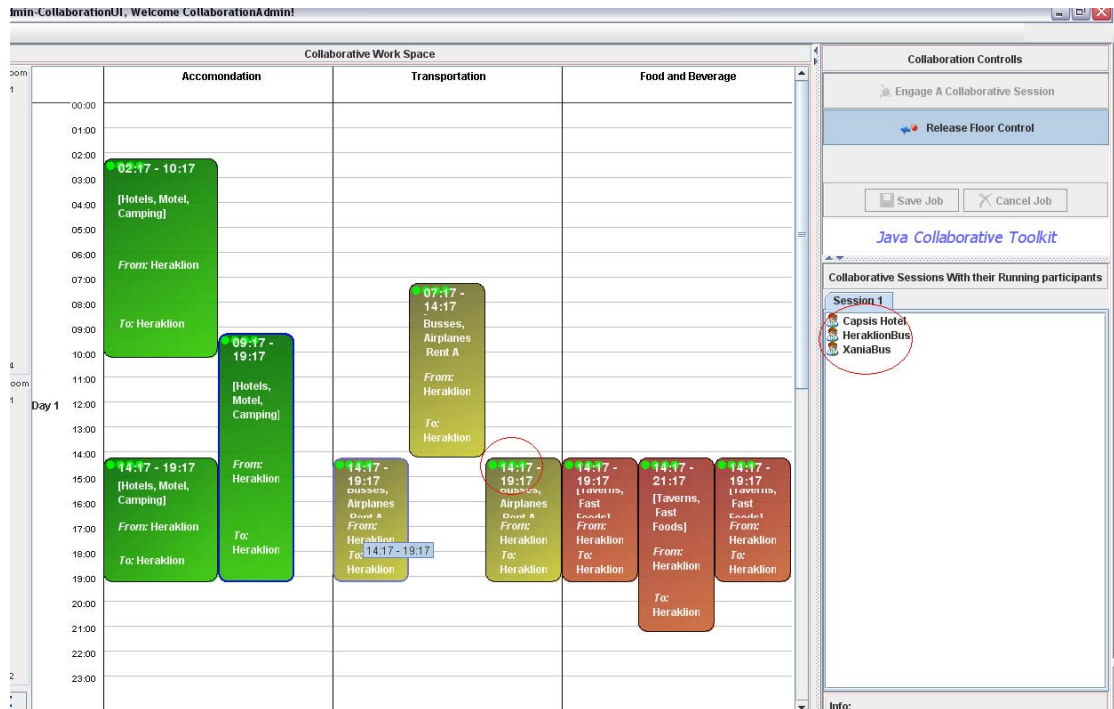
Ρόλος του επιπέδου μεταφοράς είναι η εγκαθίδρυση σύνδεσης με τον collaboration server και το άνοιγμα ενός καναλιού επικοινωνίας μέσω του οποίου μεταφέρονται τα κατάλληλα xml messages. Και εδώ η επικοινωνία μεταξύ των διαφόρων επιπέδων

γίνεται με τη βοήθεια του Event Delegation Thread. Το interpretation Layer, αναλαμβάνει τη μετατροπή των xml μηνυμάτων σε application specific events, τα οποία διερμηνεύονται κατάλληλα από το επίπεδο εφαρμογής, σαν να συνέβαιναν αυτά σε τοπικό επίπεδο.

The screenshot displays the 'Collaborative Work Space' interface. The main area is a calendar grid for 'Day 1' with columns for 'Accommodation', 'Transportation', and 'Food and Beverage'. Activities are represented by colored blocks with time ranges and descriptions. For example, in the Accommodation column, there are green blocks for '02:17 - 10:17' and '09:17 - 19:17', both labeled '[Hotels, Motel, Camping]'. In the Transportation column, there is a yellow block for '07:17 - 14:17' labeled 'Busses, Airplanes, Rent A'. In the Food and Beverage column, there are several blocks: a green one for '14:17 - 19:17' labeled '[Hotels, Motel, Camping]', a yellow one for '14:17 - 19:17' labeled 'Busses, Airplanes, Rent A', a red one for '14:17 - 19:17' labeled '[Taverns, Fast Foods]', and another red one for '14:17 - 19:17' labeled '[Taverns, Fast Foods]'. Each block also shows 'From: Heraklion' and 'To: Heraklion'. On the right side, there is a 'Collaboration Controls' panel with buttons for 'Engage A Collaborative Session', 'Release Floor Control', 'Save Job', and 'Cancel Job'. Below this is the 'Java Collaborative Toolkit' and a section for 'Collaborative Sessions With their Running participants' showing 'Session 1'. At the bottom, there is an 'HTTP MONITOR' section displaying XML messages and their responses.

Εικόνα 40 Η διεπαφή του administrator με τα replicated Objects στο κέντρο

Με το που δημιουργεί λοιπόν ο διαχειριστής ένα νέο Collaborative session, αυτομάτως γεμίζει ο διαμοιραζόμενος χώρος(Collaborative Work Space) με όλα τα activities που αφορούν το συγκεκριμένο πακέτο(πακέτο Λάρισα). Πάνω σε κάθε Replicated Object εμφανίζεται μια πράσινη βούλα η οποία δείχνει το πόσες φορές είναι replicated το συγκεκριμένο Object. Τη στιγμή που μόλις έχει δημιουργηθεί ένα νέο session, είναι λογικό να υπάρχει μια μόνο βούλα, αφού υπάρχει μόνο ένα replica κάθε αντικειμένου (αυτό του διαχειριστή). Παρατηρούμε επίσης ότι όλα τα shared Objects(activities) είναι ενεργά, αφού όπως φαίνεται στο δεξί πάνω μέρος της εικόνας 40, ο διαχειριστής έχει το Floor control του συγκεκριμένου session.



Εικόνα 41 Η διεπαφή του διαχειριστή κατά τη διάρκεια συνεργασίας με τρεις participants

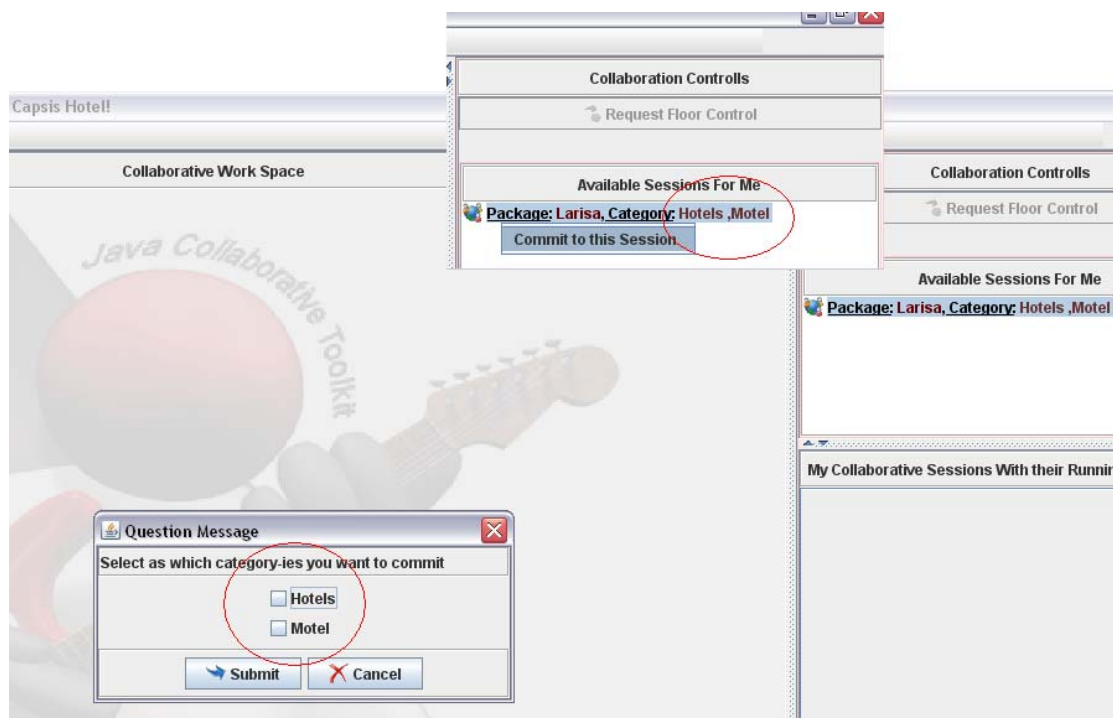
Στην παραπάνω εικόνα παρατηρούμε ένα στιγμιότυπο της διεπαφής του διαχειριστή ενώ βρίσκεται σε σύγχρονη συνεργατική σύνοδο με άλλα τρία μέλη, τα οποία και εμφανίζονται δεξιά της διεπαφής (“community” awareness). Παρατηρούμε τώρα ότι ο αριθμός των replicas έχει αυξηθεί σε σχέση με πριν σε τέσσερα. Αυτό είναι προφανές αφού τρεις φορές το κάθε αντικείμενο είναι replicated σε κάθε ένα από τα pc’s των participants που φαίνονται δεξιά της εικόνας 41 συν άλλη μια φορά που είναι replicated το κάθε Object, στο pc του administrator.

6.4 Οι εταίροι

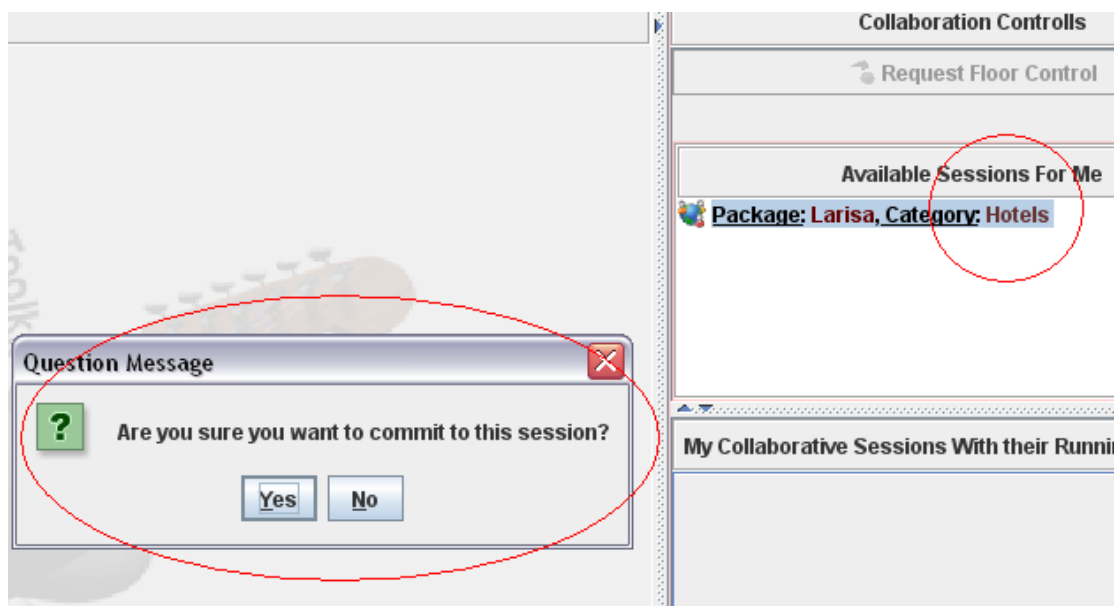
Η ίδια ακριβώς αρχιτεκτονική με του διαχειριστή έχει χρησιμοποιηθεί για τους εταίρους, με μόνη διαφορά προφανώς το Application Layer.

Με το που κάνει login ένας business partner, μέσω αυτοματοποιημένης διαδικασίας, ο server αναζητά όλα τα διαθέσιμα sessions που ενδεχομένως υπάρχουν και αφορούν τις κατηγορίες στις οποίες ανήκει ο συγκεκριμένος business partner. Σε περίπτωση που βρει ότι τρέχουν ήδη σύνοδοι που τον αφορούν, τον ενημερώνει απευθείας μέσω xml μηνύματος. Το σημαντικό σε αυτό το σημείο είναι το adaptability της εφαρμογής του εταίρου, ανάλογα με το πλήθος των κατηγοριών που ανήκει και που έχει επιλεγεί ότι μετέχουν σε ένα συγκεκριμένο collaborative

session. Συγκεκριμένα στην περίπτωση που ο εταίρος Capsis Hotel, έχει δηλώσει(κατά την εγγραφή του ως εταίρος στο eKoNEΣ) ότι ανήκει και στην κατηγορία Hotels και στην κατηγορία Motel, και ενώ ο διαχειριστής έχει επιλέξει για ένα collaborative session, να συμμετέχουν και οι δύο κατηγορίες, θα πρέπει να δοθεί στο χρήστη η δυνατότητα να επιλέξει το αν θα κάνει commit, στη μια ή στην άλλη ή και στις δύο.

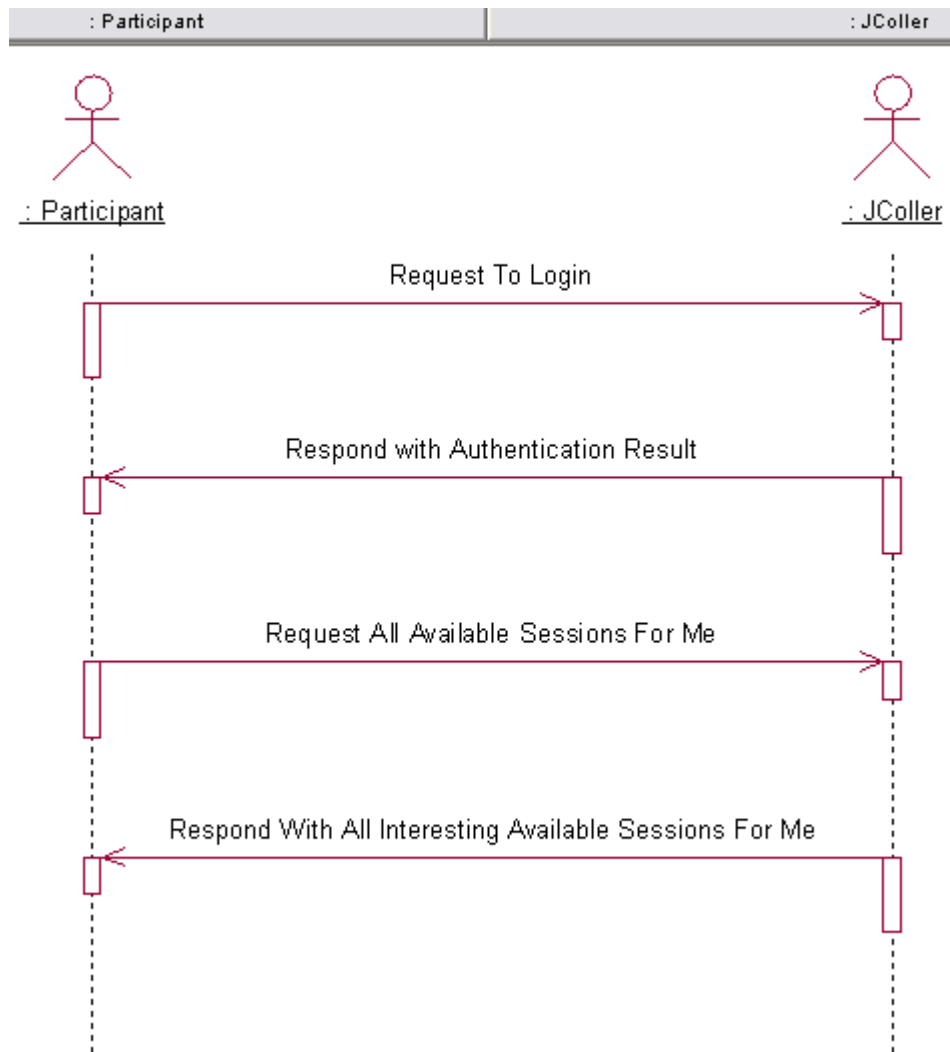


Εικόνα 42 Επιλογή του ως participant ποιας κατηγορίας θα μετέχει ο συγκεκριμένος participant

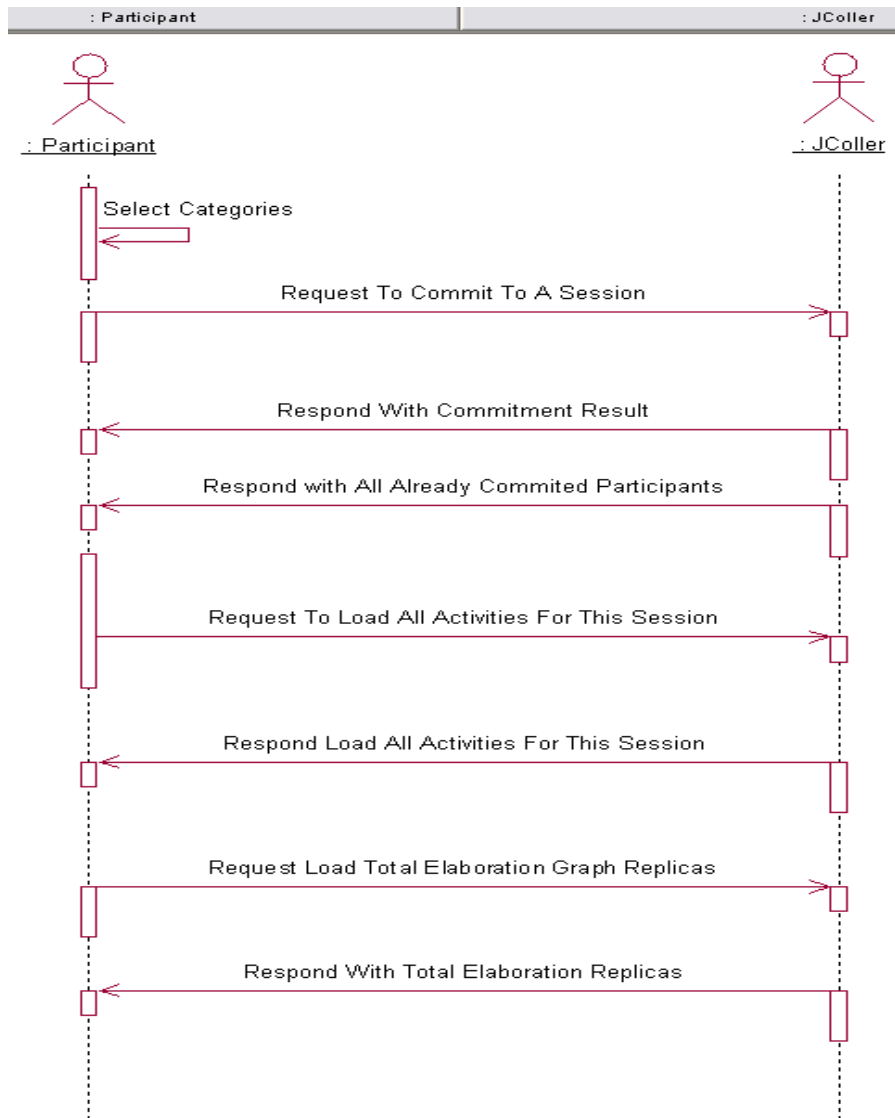


Εικόνα 43 Ερώτημα για commit σε ένα Collaborative session

Στις δύο παραπάνω εικόνες βλέπουμε πως προσαρμόζεται η διεπαφή ανάλογα με την πληροφορία που είναι να παρουσιαστεί. Συγκεκριμένα έχουμε multiple selection στην πρώτη μέσω ενός JDialog component, ενώ στη δεύτερη έχουμε single selection με JOptionPane.

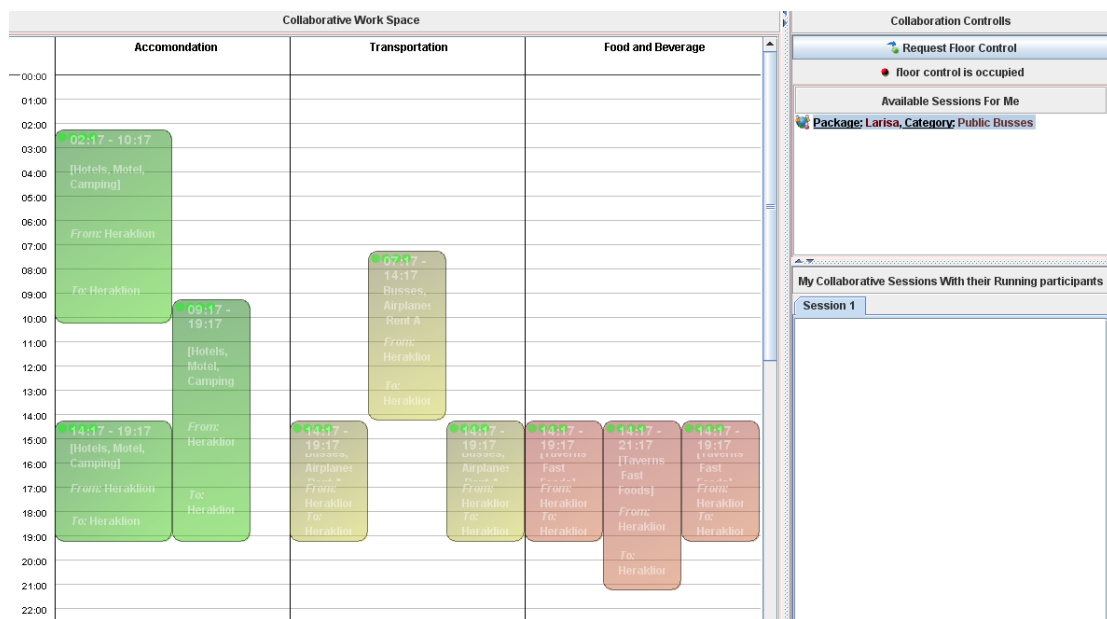


Εικόνα 44 Αίτηση για φόρτωση όλων των διαθέσιμων sessions για τον εταίρο



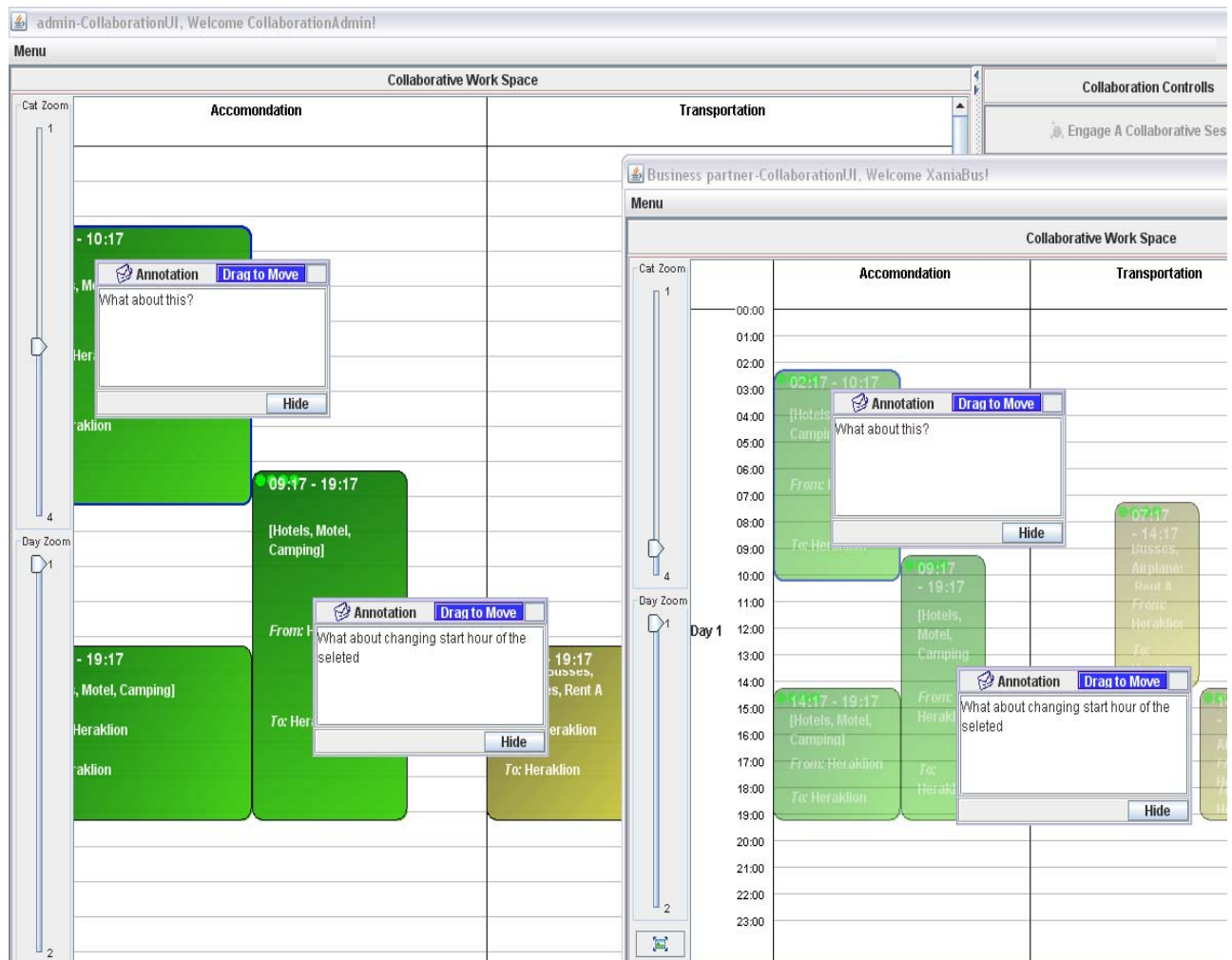
Εικόνα 45 Sequence diagram Εισόδου ενός χρήστη σε ένα collaborative session

Κατά τη διαδικασία εγγραφής ενός business partner σε ένα collaborative session, ακολουθούνται τα βήματα που φαίνονται στο παραπάνω sequence diagram. Συγκεκριμένα, ο εταίρος επιλέγει τις κατηγορίες(σε περίπτωση που έχει δηλωθεί ως πάροχος περισσότερων από μια) ως μέλος των οποίων θέλει να μετέχει στο collaborative session (προϋπόθεση για την εγγραφή του ως μέλος όλων των κατηγοριών στις οποίες ανήκει είναι να μετέχουν οι κατηγορίες αυτές στο collaborative session). Στη συνέχεια ο server κάνει ένα αίτημα στο server για commitment σε ένα συγκεκριμένο session. Ο server ακολούθως απαντάει στο request και ύστερα από την αποστολή όλων των μέχρι στιγμή μετεχόντων στο συγκεκριμένο session (community awareness), αποστέλλει το current state του μοντέλου της πληροφορίας, ώστε να γίνει Instantiated η διεπαφή του εταίρου.



Εικόνα 46 Ο γράφος του Elaboration στη διεπαφή του εταιρίου

Στην παραπάνω εικόνα παρατηρούμε ότι ο διαμοιραζόμενος χώρος έχει γεμίσει βάσει της πληροφορίας που έλαβε από το JColler. Παρατηρούμε εδώ ότι όλα τα activities πάνω στο γράφο είναι απενεργοποιημένα καθώς το floor control όπως φαίνεται πάνω δεξιά είναι occupied (στην συγκεκριμένη περίπτωση από τον administrator). Κάθε participant μπορεί να λαμβάνει και να αφήνει το floor control με τη βοήθεια των χειριστηρίων που υπάρχουν στις διεπαφές. Υπάρχει επίσης μέριμνα για το αν κάποιος χρήστης έχει το floor control για πολύ μεγάλο χρονικό διάστημα (το οποίο καθορίζεται μέσα από συγκεκριμένο xml configuration file στο server), το σύστημα να του το αφαιρεί ύστερα από ειδοποίηση.



Εικόνα 47 Multiple Replicated Annotations

Στην παραπάνω εικόνα παρατηρούμε δύο desktops, τα οποία βλέπουν τον ίδιο διαμοιραζόμενο χώρο, ενώ επιλέγουν τη χρήση annotations με σκοπό να κάνουν βασικές διευκρινήσεις πάνω στο γράφο. Ολοκληρώνοντας, πρέπει να πούμε ότι κλείνοντας ένα annotation κλείνουν και τα αντίστοιχα replicated Objects, ενώ χαρακτηριστικά όπως activity selection κλπ. υποστηρίζονται.

Κλείνοντας πρέπει αν επισημανθεί ότι στη μεριά του server έχει αναπτυχθεί σύστημα το οποίο κάνει τη μετατροπή τυποποιημένων xml μηνυμάτων σε SQL και αντίστροφα, ενώ στην πλευρά του διαχειριστή και των εταιρών έχει αναπτυχθεί σύστημα, το οποίο βρίσκεται στο επίπεδο διερμηνεύσης και το οποίο κάνει xml2Java and back conversions.

7 ΒΙΒΛΙΟΓΡΑΦΙΑ

Akoumianakis D., Vidakis N., Vellis G., Milolidakis G., Kotsalis D. (2007). Experience-based social and collaborative performance in an 'electronic village' of local interest: The eKoNEΣ framework, In Proceedings of the 9th International Conference on Enterprise Information Systems (ICEIS'2007), 12-16 June, Funchal, Madeira – Portugal.

Heath, C. C. & Luff, P. K. (1991) Work interaction and technology: studies in social ergonomics, *Proceedings of the International Ergonomics Association Conference*, Paris, pp. 340-352.

Heath, C. C. & Luff, P.K. (1992) Explicating face to face interaction, in N. Gilbert (ed.) *Researching Social Life*, London: Sage, 306-327

Heath, C. C. and P. K. Luff (1996). Convergent activities: collaborative work and multimedia technology in London Underground Line Control Rooms, in D. Middleton and Y. Engestrom (eds.), *Cognition and Communication at Work: Distributed Cognition in the Workplace*, Cambridge: Cambridge University Press, pp. 96-130

uff, P. K., Heath, C. C. & D. L. Greatbatch (1992). Tasks-in-interaction: paper and screen based activity in collaborative work. *Proceedings of the Conference on Computer Supported Cooperative Work*, Toronto, pp. 163- 170

Bannon, L. and K. Schmidt (1991). CSCW - four characters in search of a context. Studies in computer supported cooperative work - theory, practice and design. J. M. Bowers and S. Benford. Amsterdam, North Holland.

Bordeau, J. and B. Wasson (1997). Orchestrating collaboration in collaborative telelearning. Artificial intelligence in education. B. Boulay and R. Mizoguchi, IOS Press: 565-567.

M. Lynne Markus , Terry Connolly, Why CSCW applications fail: problems in the adoption of interdependent work tools, Proceedings of the 1990 ACM conference on Computer-supported cooperative work, p.371-380, October 07-10, 1990, Los Angeles, California, United States

Wanda J. Orlikowski, Learning from Notes: organizational issues in groupware implementation, Proceedings of the 1992 ACM conference on Computer-supported cooperative work, p.362-369, November 01-04, 1992, Toronto, Ontario, Canada

Brehmer, B. (1991). Distributed decision making: some notes on the literature. Distributed decision making : cognitive models for cooperative work. J. Rasmussen, B. Brehmer and J. Leplat. Chichester, England ; New York, Wiley.

Carstensen, P. H. and K. Schmidt (2002). Computer supported cooperative work: New challenges to systems design. Handbook of Human Factors. K. Itoh. Tokyo, [in press].

Coleman, D. and R. Khanna (1995). Groupware : technologies and applications. Upper Saddle River, NJ, Prentice Hall PTR.

Dillenbourg, P., M. Baker, A. Blaye and C. O'Malley (1996). The Evolution of Research on Collaborative Learning. Learning in humans and machines. Towards an interdisciplinary learning science. P. Reimann and H. Spada. London, Pergamon: 189-211.

Ellis, C. A., S. J. Gibbs and G. L. Rein (1991). "Groupware: some issues and experiences." Communications of the ACM 34(1): 38-59.

Ellis, C. A., S. J. Gibbs and G. L. Rein (1993). Groupware some issues and experiences. Readings in groupware and computer-supported cooperative work. R. M. Baecker, Morgan Kaufmann Publishers, Inc.

Grudin, J. (1994). "Computer-supported cooperative work: Its history and participation." IEEE Computer 27(5): 19-26.

Wilson, P. (1991). Computer supported cooperative work : an introduction. Oxford, England Norwell, MA, Intellect ; Sold and distributed in the U.S.A. and Canada by Kluwer Academic Publishers. ISBN 0-7923-1446-8

Michel Beaudouin-Lafon , Alain Karsenty, Transparency and awareness in a real-time groupware system, Proceedings of the 5th annual ACM symposium on User interface software and technology, p.171-180, November 15-18, 1992, Monterey, California, United States

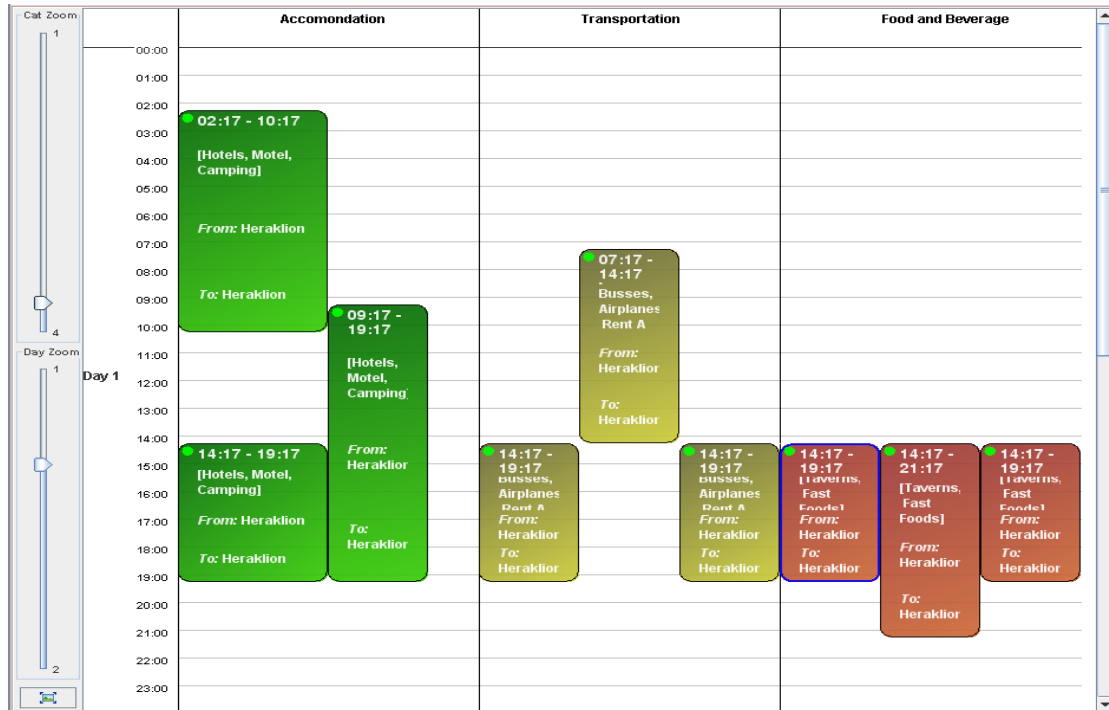
DOURISH, P. 1996. Open implementation and flexibility in CSCW toolkits. Ph.D. Dissertation. Department of Computer Science, University College, London, England.

Bowers, J., Pycock, J., Dean, G. and Rodden, T. 'Running the network: Supporting cooperative systems'. Submitted for publication.

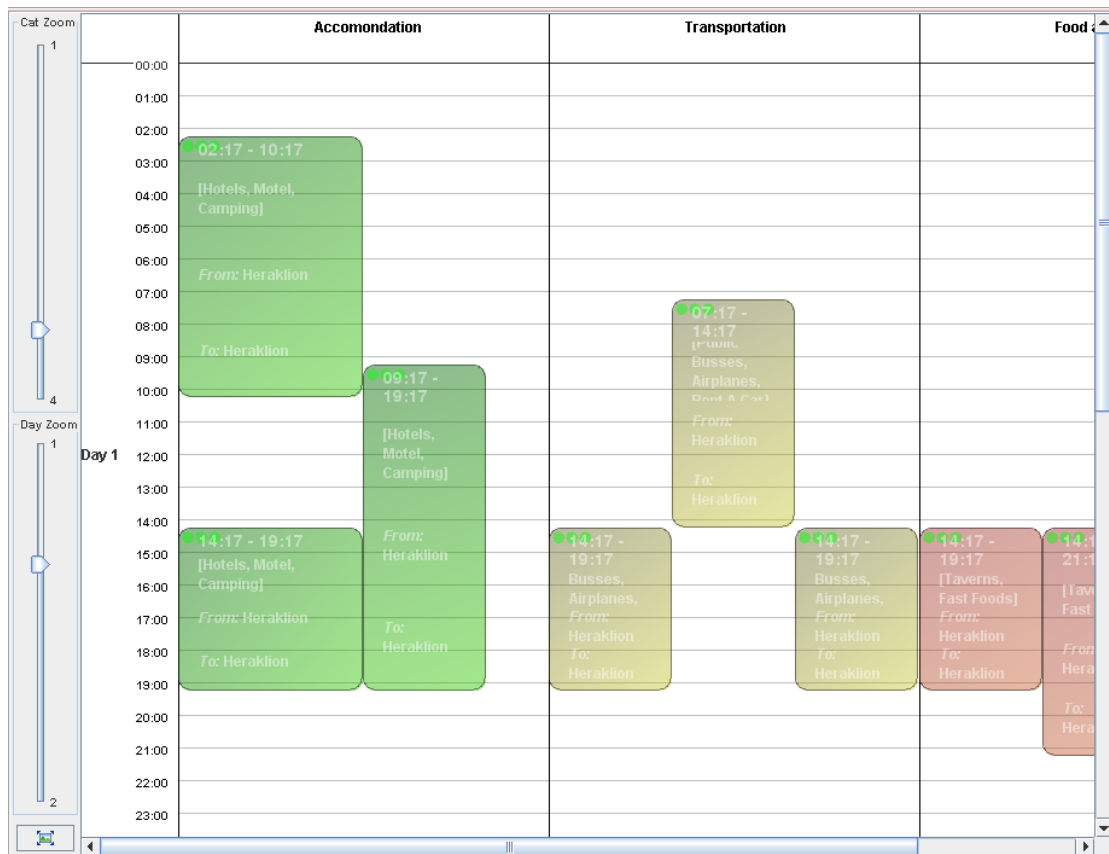
John Bowers , Tom Rodden, Exploding the interface: experiences of a CSCW network, Proceedings of the SIGCHI conference on Human factors in computing systems, p.255-262, April 24-29, 1993, Amsterdam, The Netherlands

Bullen, C. and Bennett, J. 'Groupware in practice: An interpretation of work experience'. In Proceeding~ of CSCW'90, ACM Press, New York, 1990.

8 ΠΑΡΑΡΤΗΜΑ



Εικόνα 48 Διαμοιραζόμενος χώρος με floor control



Εικόνα 49 Διαμοιραζόμενος χώρος χωρίς floor control