

ΜΕΘΟΔΟΛΟΓΙΑ ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ ΓΙΑ ΤΟ ΣΧΕΔΙΑΣΜΟ ΚΑΤΑΝΕΜΗΜΕΝΩΝ ΣΥΣΤΗΜΑΤΩΝ

Κ. Βασιλάκης, Π. Γεωργιάδης, Δ. Λέλης, Δ. Μουζάκης, Μ. Νικολαΐδου

**Πανεπιστήμιο Αθηνών
Τμήμα Πληροφορικής
Πανεπιστημιόπολις
157 71 Ιλίσια**

ΠΕΡΙΛΗΨΗ

Καθώς τα κατανεμημένα συστήματα διευρύνουν το πεδίο χρήσης τους και οι εφαρμογές που σχεδιάζονται γι' αυτά γίνονται ολοένα και πιο απαιτητικές, η βέλτιστη σχεδίαση τους γίνεται ιδιαίτερα σημαντική. Παράλληλα, η πολυπλοκότητα αυτής της σχεδίασης έχει αυξηθεί, λόγω της ύπαρξης πολλών εναλλακτικών λύσεων για κάθε απόφαση που πρέπει να ληφθεί και που αφορά την επιλογή δομικών στοιχείων του κατανεμημένου συστήματος, αλλά και της πληθώρας των παραμέτρων που καθορίζουν την τελική απόδοσή του. Καθίσταται έτσι επιτακτική η ανάγκη χρήσης εργαλείων που υποδέχονται την περιγραφή των απαιτήσεων του χρήστη από ένα κατανεμημένο σύστημα και προτείνουν τεκμηριωμένες λύσεις για το σχεδιασμό ενός συστήματος που να ικανοποιεί τις απαιτήσεις αυτές. Στην εργασία αυτή παρουσιάζουμε μία μεθοδική προσέγγιση στην κατασκευή ενός τέτοιου εργαλείου λογισμικού, στην οποία συνδυάζονται μέθοδοι, από το χώρο της τεχνητής νοημοσύνης για τη σχεδίαση του κατανεμημένου συστήματος, με τεχνικές προσομοίωσης για την εκτίμηση της απόδοσής του.

TOWARDS THE DEVELOPMENT OF AN EXPERT TOOL FOR DESIGNING DISTRIBUTED SYSTEMS

ABSTRACT

As the use of distributed systems is spreading and applications designed for such systems become more and more demanding, optimal design of distributed systems becomes a critical issue. Designing a distributed system has become more complex, due to the number of alternatives for each decision that must be made and because of the existence of many parameters which influence the overall performance of the distributed system. Thus, it is necessary to use of software tools, capable of accepting a description of the user's requirements and suggesting solutions to the problem of designing a distributed system which meets the user's requirements. In this paper we present a disciplined approach to the construction of such a software tool, which combines methods from the Artificial Intelligence domain, that are used in order to design the distributed system along with simulation techniques, used to estimate the system's overall performance.

I. ΕΙΣΑΓΩΓΗ

Ως κατανεμημένα ορίζονται τα συστήματα που συντίθενται από ένα σύνολο από υπολογιστές, οι οποίοι δε διαθέτουν κοινή μνήμη, αλλά δίνουν την εντύπωση στους χρήστες ότι συνδιαλέγονται με έναν υπολογιστή. Τέτοια συστήματα βασίζονται σε ένα σύνολο από ανεξάρτητους υπολογιστές που είναι ικανοί να λειτουργούν αυτόνομα και να ανταλλάσσουν

πληροφορίες μεταξύ τους μέσω ενός δικτύου [1]. Τα καταναμημένα συστήματα έχουν σχεδιαστεί ώστε να επιτρέπουν στους ανεξάρτητους υπολογιστές από τους οποίους αποτελούνται να χρησιμοποιούν τους διαμοιραζόμενους πόρους του δικτύου για να προσφέρουν υπολογιστικές δυνατότητες τουλάχιστον το ίδιο ευέλικτες και γενικές όσο αυτές ενός κεντροποιημένου υπολογιστικού συστήματος. Η πληρέστερη προσπάθεια για τον καθορισμό εκείνων των ιδιοτήτων, που χαρακτηρίζουν ένα υπολογιστικό σύστημα ως καταναμημένο, γίνεται στα πλαίσια του προγράμματος ANSA (Advanced Networked System Architecture), που ξεκίνησε το 1986 στη Μεγάλη Βρετανία και αποτελεί το βασικό τροφοδότη της επιτροπής ODP/DAF (CCITT-OSI), που υιοθέτησε την άποψη του ANSA στις διεθνείς συστάσεις X.900 (CCITT) και ISO 10746 (ISO) για τα καταναμημένα συστήματα [2].

Τα καταναμημένα συστήματα ([1], [3]) ανήκουν στην κατηγορία των *χαλαρά συνδεδεμένων* (loosely coupled) συστημάτων. Ανάλογα με το βαθμό ανεξαρτησίας υλικού και λογισμικού στο χώρο των καταναμημένων συστημάτων διακρίνονται τρεις μεγάλες κατηγορίες εφαρμογών:

- α. *Λειτουργικά συστήματα δικτύων*: Οι υπολογιστές που είναι συνδεδεμένοι σε ένα δίκτυο, λειτουργούν ανεξάρτητα και ανταλλάσσουν πληροφορίες, όποτε κριθεί αναγκαίο. Το πιο γνωστό παράδειγμα συστήματος αυτής της κατηγορίας είναι το λειτουργικό σύστημα δικτύων SUN NFS .
- β. *Καταναμημένες εφαρμογές*. Οι εφαρμογές ή τμήματα των εφαρμογών εκτελούνται σε ανεξάρτητους υπολογιστές, ο καθένας από τους οποίους τρέχει το δικό του λειτουργικό σύστημα. Χαρακτηριστικό παράδειγμα είναι ο καταναμημένος εξυπηρετής αρχείων Xerox (Distributed File Server - DFS) αλλά και πλήθος εφαρμογές, όπως οι βάσεις δεδομένων και το ηλεκτρονικό ταχυδρομείο, όταν υλοποιούνται σε περισσότερους από έναν υπολογιστές.
- γ. *Καταναμημένα λειτουργικά συστήματα*. Ένα καταναμημένο λειτουργικό σύστημα δίνει την εντύπωση πως λειτουργεί σαν ένα κεντροποιημένο λειτουργικό σύστημα, ενώ τμήματα του εκτελούνται σε έναν ή περισσότερους ανεξάρτητους υπολογιστές, όπως τα Locus, Amoeba, Mach ([4], [5], [6]).

Η αρχιτεκτονική ενός καταναμημένου συστήματος προσδιορίζεται από τα βασικά *δομικά στοιχεία* του συστήματος και τις αναγκαίες σχέσεις μεταξύ τους. Τα βασικά δομικά στοιχεία ενός καταναμημένου συστήματος είναι τα ακόλουθα:

- α. *υπολογιστικοί κόμβοι*, οι οποίοι παρέχουν την αναγκαία υπολογιστική ισχύ για την κάλυψη των αναγκών των χρηστών .
- β. *επικοινωνιακά στοιχεία*, τα οποία διασυνδέονται με τους υπολογιστικούς κόμβους αλλά και μεταξύ τους, προκειμένου να εξασφαλίσουν τη δυνατότητα ανταλλαγής πληροφοριών ανάμεσα στους υπολογιστικούς κόμβους, όταν κάτι τέτοιο είναι απαραίτητο.
- γ. *τερματικοί σταθμοί*, που αποτελούν τα σημεία πρόσβασης των χρηστών στο καταναμημένο σύστημα.
- δ. *συσκευές αποθήκευσης*, όπου αποθηκεύονται τόσο οι εφαρμογές όσο και τα δεδομένα.
- ε. *εφαρμογές*, τα προγράμματα δηλαδή που οι χρήστες εκτελούν. Οι εφαρμογές εκτελούνται σε έναν ή περισσότερους υπολογιστικούς κόμβους.
- στ. *δεδομένα*, που χρησιμοποιούνται από τις εφαρμογές. Το είδος των δεδομένων εξαρτάται από την εφαρμογή και μπορεί να είναι κείμενο, φωνή, εικόνα, κ.λπ.

Τα στοιχεία, που ανήκουν στις τέσσερις πρώτες κατηγορίες, διασυνδέονται μεταξύ τους και αποτελούν το δικτυακό υπόστρωμα στο οποίο βασίζεται το καταναμημένο σύστημα. Τα στοιχεία, που ανήκουν στις δύο επόμενες, χρησιμοποιούνται για να περιγράψουν τις καταναμημένες εφαρμογές που εκτελούνται στο σύστημα.

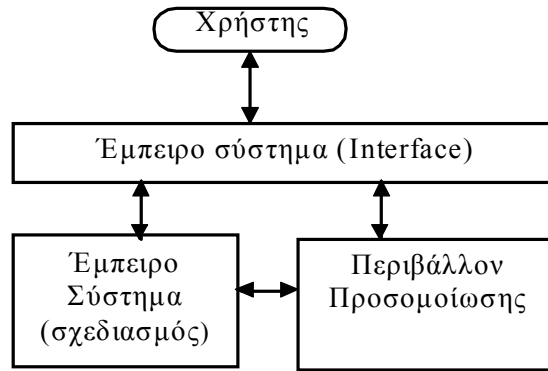
Ο σχεδιασμός μιας κατανεμημένης αρχιτεκτονικής είναι ιδιαίτερα πολύπλοκος. Η βασική αιτία της πολυπλοκότητας αυτής είναι η ανάγκη για επιτυχή συνδυασμό των ανεξάρτητων δομικών στοιχείων ([7]). Το πλήθος των εναλλακτικών λύσεων και η ανάγκη συνδυασμού γνώσεων από διαφορετικές γνωστικές περιοχές, όπως η αρχιτεκτονική υπολογιστών, η τεχνολογία λογισμικού και οι επικοινωνίες, καθιστούν επιτακτική τη χρήση εργαλείων λογισμικού (software tools), που θα βοηθήσουν στο σχεδιασμό κατανεμημένων συστημάτων. Η ανάγκη αυτή γίνεται πιο εμφανής όταν σχεδιάζονται κατανεμημένα συστήματα με πολλούς κόμβους και ανομοιογενείς απαιτήσεις. Ως αποτέλεσμα παρουσιάζεται η ανάγκη επινόησης διαφόρων εργαλείων που μπορούν να βοηθήσουν το σχεδιαστή του κατανεμημένου συστήματος. Οι επί μέρους γνώσεις όμως δε συνδυάζονται στο σύνολό τους από κανένα υπάρχον εργαλείο και έτσι το βάρος της υποστήριξης των αναγκών που έπρεπε να καλύπτουν οι παραλειπόμενοι τομείς μεταφέρεται στο χρήστη του συστήματος. Τα ήδη υπάρχοντα εργαλεία μπορούν να ταξινομηθούν στις εξής κατηγορίες:

- εκείνα που επικεντρώνονται στην τοποθέτηση υπολογιστικών πόρων (ή μέρους των υπολογιστικών πόρων) στο κατανεμημένο περιβάλλον, αφήνοντας τις αποφάσεις που αφορούν τις εφαρμογές καθώς και την αποτίμηση της απόδοσης στην αρμοδιότητα του χρήστη ([8], [9]).
- αυτά που εκτιμούν την απόδοση ενός κατανεμημένου συστήματος (ή τμήματός του) μεταφέροντας συνολικά την ευθύνη του σχεδιασμού στο χρήστη ([10], [11], [12], [13], [14], [15]).
- όσα ασχολούνται με την τοποθέτηση δεδομένων ([16], [17], [18]) ή/και υπολογιστικών πόρων ([19], [20]) σε τοποθεσίες ακολουθώντας "σκληρές" μαθηματικές μεθόδους.

Στις πιο πάνω κατηγορίες είναι εμφανής η έλλειψη τεχνικών ολοκλήρωσης τεχνογνωσίας από τον τομέα της τεχνητής νοημοσύνης και τον τομέα της προσομοίωσης. Η υστέρηση αυτή οφείλεται και στη ραγδαία εξέλιξη που παρατηρείται στον τομέα των κατανεμημένων συστημάτων και δυσχεραίνει περαιτέρω την ανάπτυξη εργαλείων σχεδίασης και προσομοίωσης τους, καθώς, λόγω της εξέλιξης αυτής, παρουσιάζονται απαιτήσεις για:

- τεχνικές με τις οποίες οι βάσεις γνώσεων και οι μηχανισμοί εξαγωγής συμπερασμάτων πάνω στους οποίους στηρίζονται τα έμπειρα σχεδιαστικά συστήματα, θα μπορούν να παρακολουθούν την εξέλιξη και να αξιοποιούν τα νέα τεχνολογικά δεδομένα.
- κατάλληλη μοντελοποίηση ώστε τα συστήματα προσομοίωσης να μπορούν με ελάχιστες ή καθόλου αλλαγές να προσαρμόζονται στα νέα δεδομένα.

Στα συστήματα υποστήριξης αποφάσεων, συστήματα δηλαδή που πρέπει να παράγουν τελικές και αιτιολογημένες απαντήσεις, εμφανίζονται μεθοδολογίες που ολοκληρώνουν τεχνικές από τους δύο τομείς. Σε μία από τις μεθοδολογίες αυτές, όπως φαίνεται στο σχήμα 1, βρίσκουμε την πιο στενή συνεργασία μεταξύ του έμπειρου συστήματος και του περιβάλλοντος προσομοίωσης, όπου το έμπειρο σύστημα περιλαμβάνει γνώση για το πως θα συνθέσει και θα τροποποιήσει το υπό σχεδίαση σύστημα ενώ το περιβάλλον προσομοίωσης μπορεί να εκτιμήσει την απόδοσή του. Το έμπειρο σύστημα (ή τμήμα του) μπορεί να αναλάβει και το ρόλο του "έξυπνου" front-end προκειμένου να προσφέρει στο χρήστη ένα πιο φιλικό περιβάλλον εργασίας και τον καθοδηγεί κατά τη διάρκεια της περιγραφής του προβλήματος.



Σχήμα 1 - Ολοκλήρωση Έμπειρου Συστήματος και Περιβάλλοντος Προσομοίωσης

Κεντρικός ερευνητικός μας στόχος είναι ο σχεδιασμός και η υλοποίηση ενός *Έμπειρου Συμβουλευτικού Συστήματος*, που θα αφορά στο σχεδιασμό ενός ολοκληρωμένου καταναμημένου συστήματος. Σκοπός του συστήματος αυτού είναι να προτείνει την ολοκληρωμένη αρχιτεκτονική του καταναμημένου συστήματος (δικτυακό υπόστρωμα και καταναμημένη εφαρμογή) σύμφωνα με τις ανάγκες του χρήστη και τη διαθέσιμη τεχνολογία και να εκτιμήσει την απόδοση της προτεινόμενης αρχιτεκτονικής χρησιμοποιώντας τεχνικές προσομοίωσης.

II. ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΟΥ ΕΜΠΕΙΡΟΥ ΣΥΜΒΟΥΛΕΥΤΙΚΟΥ ΣΥΣΤΗΜΑΤΟΣ

Το προτεινόμενο στην εργασία αυτή Έμπειρο Συμβουλευτικό Σύστημα ξεκινά από τη συλλογή των αναγκών του χρήστη και καταλήγει σε μία ολοκληρωμένη πρόταση για το σχεδιασμό ενός υπολογιστικού συστήματος που ανταποκρίνεται στις ανάγκες αυτές και, πιο συγκεκριμένα:

- σχεδιάζει το δικτυακό υπόστρωμα, που μπορεί να είναι ένα απλό τοπικό δίκτυο (LAN) ή ένα σύνολο από διασυνδεδεμένα δίκτυα (internetwork), τόσο τοπικά όσο και ευρείας περιοχής (WAN).
- σχεδιάζει τις καταναμημένες εφαρμογές (distributed applications), όπως βάσεις δεδομένων, που λειτουργούν στα πλαίσια του καταναμημένου συστήματος.

Η διαδικασία αυτή περιλαμβάνει τα ακόλουθα στάδια:

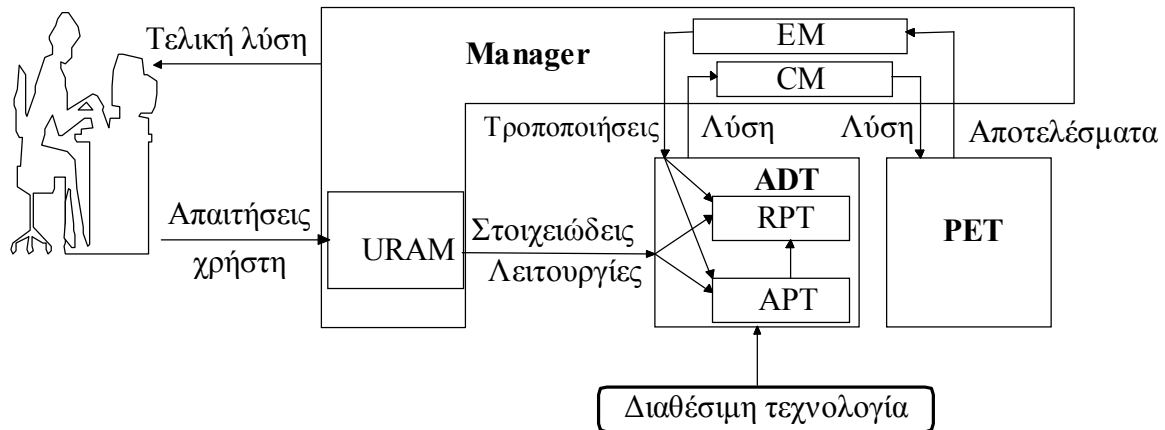
- α. συλλογή και ανάλυση των αναγκών του χρήστη.
- β. διαμόρφωση πρότασης για το σχεδιασμό ενός υπολογιστικού συστήματος.
- γ. εκτίμηση της απόδοσης του σχεδιασθέντος συστήματος, με πιθανή τροποποίηση του σχεδιασμού και επανεκτίμηση της απόδοσης (επιστροφή δηλαδή στο βήμα 2), αν τα αποτελέσματα δεν είναι ικανοποιητικά.

Η διεκπεραίωση της πιο πάνω διαδικασίας γίνεται από συνεργαζόμενα τμήματα λογισμικού, κάθε ένα από τα οποία είναι επιφορτισμένο με την επιτέλεση κάποιας από τις προαναφερθείσες λειτουργίες ή υλοποιεί βοηθητικές υπηρεσίες που είναι απαραίτητες για την ολοκλήρωση των λειτουργιών που παρέχονται από τα αυτόνομα τμήματα. Στο Έμπειρο Συμβουλευτικό Σύστημα διακρίνονται τα ακόλουθα τμήματα:

- α. ανάλυσης αναγκών του χρήστη (**URAM - User Requirement Analysis Module**).
- β. σχεδιασμού της προτεινόμενης αρχιτεκτονικής (**ADT - Architecture Definition Tool**), που είναι υπεύθυνο για τη διαμόρφωση των λύσεων. Το ADT χωρίζεται σε δύο υποσυστήματα, το APT και το RPT, η λειτουργικότητα των οποίων θα εξηγηθεί στη συνέχεια.
- γ. αποτίμησης της απόδοσης της αρχιτεκτονικής (**PET- Performance Evaluation Tool**), το οποίο αποτιμά την απόδοση των λύσεων που προτείνει το ADT.
- δ. επικοινωνίας ανάμεσα σε ADT-PET (**CM - Communication Module**)

- ε. αξιολόγησης των αποτελεσμάτων του PET (**EM - Evaluation Module**), που έχει ως στόχο να εκτιμήσει αν στη λύση που πρότεινε το ADT υπάρχουν προβλήματα, βασιζόμενο στα αποτελέσματα του PET.

Οι λειτουργίες των τμημάτων (1), (4) και (5) έχουν το κοινό χαρακτηριστικό ότι αφορούν το γενικότερο συντονισμό και έλεγχο της πορείας προς την τελική λύση του προβλήματος και έτσι τα τμήματα αυτά ομαδοποιούνται σε ένα πιο σύνθετο τμήμα λογισμικού, το οποίο καλείται **Manager**. Η ομαδοποίηση αυτή δε συνεπάγεται την απώλεια αυτονομίας των τμημάτων αυτών, αλλά γίνεται για καθαρά εννοιολογικούς και οργανωτικούς λόγους. Η συνολική αρχιτεκτονική του Έμπειρου Συμβουλευτικού Συστήματος φαίνεται στο σχήμα 2.



Σχήμα 2 - Συνολική Αρχιτεκτονική

Στις επόμενες παραγράφους θα αναλυθεί περαιτέρω η δομή και η λειτουργία των τμημάτων του Έμπειρου Συμβουλευτικού Συστήματος.

A. Ανάλυση των αναγκών του χρήστη (URAM)

Το πρώτο στάδιο λειτουργίας του Έμπειρου Συμβουλευτικού Συστήματος είναι η ανάλυση των αναγκών του χρήστη, τα αποτελέσματα της οποίας θα προωθηθούν για περαιτέρω επεξεργασία στο ADT. Ο όρος "ανάλυση των αναγκών του χρήστη" συμπεριλαμβάνει ([21]):

1. τη συλλογή της πληροφορίας από το χρήστη, ο οποίος καθοδηγείται στην περιγραφή του καταναμεμένου συστήματος μέσω κατάλληλα διαμορφωμένων ερωτηματολογίων. Η πληροφορία που συγκεντρώνεται από το χρήστη αποθηκεύεται σε μία βάση γνώσεων η οποία καλείται **ExKUser (Explicit Knowledge from the User)**.
2. τη συμπλήρωση της πληροφορίας που ο χρήστης εισήγαγε με στοιχεία τα οποία περιέχονται έμμεσα (implicitly) στην περιγραφή του χρήστη. Η εξαγωγή της υπονοούμενης (implicit) πληροφορίας γίνεται με το να δράσει πάνω στην ExKUser ένα σώμα κανόνων που καλείται **RUser** και ως αποτέλεσμα λαμβάνουμε τη βάση γνώσεων **ImKUser (Implicit Knowledge from the User)**. Η συμπλήρωση της πληροφορίας που εισάγεται από το χρήστη έχει ως αποτέλεσμα να απλουστεύεται η λειτουργία των επομένων τμημάτων του Έμπειρου Συμβουλευτικού Συστήματος, μια και όλη η πληροφορία που χρειάζονται θα είναι σαφώς προσδιορισμένη (explicit).
3. την απεικόνιση της πληροφορίας που εισάγει ο χρήστης, η οποία περιγράφει τις ανάγκες του, σε πληροφορία που περιγράφει απαιτήσεις σε φυσικούς πόρους (physical resources), η οποία είναι πιο κατάλληλη για τα επόμενα τμήματα του Έμπειρου Συμβουλευτικού Συστήματος. Η απεικόνιση γίνεται από το κατάλληλο τμήμα του Manager, το οποίο λαμβάνει ως είσοδο τις βάσεις γνώσεων ExKUser και ImKUser, καθώς και πληροφορία που

αφορά τον τρόπο που θα προχωρήσει στην απεικόνιση (η γνώση αυτή είναι αποθηκευμένη στη βάση γνώσεων SIBs, που περιγράφεται αναλυτικά στις επόμενες παραγράφους) και παράγει τη βάση γνώσεων **FinKUser** (**F**inal **K**nowledge from the **U**ser). Η FinKUser αποτελεί και την τελική έξοδο του τμήματος ανάλυσης των αναγκών του χρήστη.

Η απεικόνιση των αναγκών του χρήστη απλουστεύεται και τυποποιείται αν για την περιγραφή του κατανεμημένου περιβάλλοντος χρησιμοποιηθεί ένα σύνολο από παρεχόμενες υπηρεσίες (services). Ως **υπηρεσία (service)** σε ένα κατανεμημένο περιβάλλον ορίζεται, σύμφωνα με τη σύσταση R.30 της CCITT ([22]), οποιαδήποτε εργασία θα μπορούσε ο χρήστης να εκτελεί στο κατανεμημένο περιβάλλον και πρέπει να προσφέρεται από αυτό. Οι υπηρεσίες που είναι απαραίτητες για να καλύψουν πλήρως τις απαιτήσεις του χρήστη από μια κατανεμημένη εφαρμογή, καταγράφονται και αναλύονται σε περισσότερες ανεξάρτητες μεταξύ τους υπουπηρεσίες, που πρέπει απαραίτητα να προσφέρονται από το περιβάλλον. Η διάσπαση αυτή γίνεται σε πολλά επίπεδα, κάθε ένα από τα οποία αντιστοιχεί σε ένα συγκεκριμένο βαθμό αφαίρεσης, μέχρι να προσδιοριστούν **στοιχειώδεις υπηρεσίες (SIBs - Service Independent Building Blocks)**, που δεν αναλύονται περισσότερο και περιγράφουν βασικές λειτουργίες του συστήματος, όπως η μεταφορά δεδομένων ή η επεξεργασία τους. Οι στοιχειώδεις υπηρεσίες καταγράφουν τις βασικές απαιτήσεις του χρήστη από το πληροφοριακό σύστημα και αντιστοιχούν σε στοιχειώδεις λειτουργίες του κατανεμημένου συστήματος. Με βάση τις στοιχειώδεις υπηρεσίες υπολογίζονται στην ολότητά τους οι απαιτήσεις του χρήστη για πόρους του δικτύου και χρησιμοποιούνται για να τροφοδοτήσουν το ADT, το οποίο θα συνθέσει την "βέλτιστη δυνατή" λύση για τη σχεδίαση του κατανεμημένου συστήματος. Κάθε υπηρεσία ενός επιπέδου μπορεί να συμμετέχει στη σύνθεση περισσότερων από μιας υπηρεσιών κάποιου ανώτερου επιπέδου. Οι υπηρεσίες σε όλα τα επίπεδα περιγράφονται με παραμέτρους έτσι ώστε να είναι δυνατή η αποτύπωση όχι μόνο των ποιοτικών αλλά και των ποσοτικών αναγκών του χρήστη.

Η ανάλυση των αναγκών του χρήστη σε SIBs θα πρέπει να είναι πλήρης και μονοσήμαντη, με την έννοια ότι πρέπει να είναι σε θέση να περιγράψει οποιαδήποτε απαίτηση του χρήστη από το κατανεμημένο σύστημα και ότι κάθε απαίτηση απεικονίζεται σε ένα καλά ορισμένο σύνολο από SIBs. Η απεικόνιση των αναγκών του χρήστη γίνεται σύμφωνα με τον αλγόριθμο που περιγράφεται πιο κάτω:

- α. στην υπηρεσία που θέλουμε να απεικονίσουμε, που έστω ότι ανήκει στο επίπεδο i , αναγνωρίζονται διακριτές υπουπηρεσίες που χρησιμοποιούνται κατά την εκτέλεσή της. Η κάθε μία από τις υπουπηρεσίες αυτές θα ανήκει τουλάχιστον στο $i + 1$ επίπεδο (μπορεί έτσι να ανήκει και σε επίπεδο υψηλότερης αριθμητικά τάξης, δηλ. $i + 2$, κ.τ.λ.).
- β. αν οι υπουπηρεσίες που προέκυψαν από το προηγούμενο βήμα έχουν ήδη αναλυθεί, τότε ο αλγόριθμος σταματά. Αν όμως κάποια από τις υπουπηρεσίες είναι καινούρια, τότε προστίθεται στο σύνολο των διαθέσιμων υπηρεσιών του $i + 1$ επιπέδου. Αν η υπουπηρεσία χρήζει περαιτέρω ανάλυσης, τότε ακολουθείται η ίδια διαδικασία, από το πρώτο βήμα, για την καινούρια υπουπηρεσία.
- γ. αν κάποια από τις υπουπηρεσίες που προέκυψαν υπάρχει, αλλά βρίσκεται στο επίπεδο λ , όπου $i \geq \lambda$, τότε η υπουπηρεσία αυτή μετακινείται από το επίπεδο που βρισκόταν στο επίπεδο $i + 1$. Αν η υπουπηρεσία αυτή έχει αναλυθεί με τη σειρά της, τότε οι υπουπηρεσίες που χρησιμοποιεί μετακινούνται κατάλληλα ώστε να διατηρείται πάντα ο περιορισμός ότι κάθε υπηρεσία αναλύεται σε υπουπηρεσίες χαμηλότερου επιπέδου.

Ο αλγόριθμος αυτός επαναλαμβάνεται μέχρι να αναλυθούν όλες οι υπουπηρεσίες που προέκυψαν κατά τη διάρκεια της ανάλυσης.

Κατά την εφαρμογή της διαδικασίας αυτής θα πρέπει να δίνεται προσοχή στο να μη δημιουργούνται περιττές ή επικαλυπτόμενες (μερικά ή ολικά) υπηρεσίες στα διάφορα επίπεδα. Εάν διαπιστωθεί ύπαρξη περιττής υπηρεσίας, τότε αυτή απλά διαγράφεται, ενώ για τις επικαλυπτόμενες υπηρεσίες θα πρέπει να υπάρξει ένας διαχωρισμός τους σε πιο στοιχειώδεις υπηρεσίες που δεν θα επικαλύπτονται ή/και μια νέα ανάλυση της υπηρεσίας ή των υπηρεσιών που περιλαμβάνουν τις υπουπηρεσίες που προκαλούν το πρόβλημα.

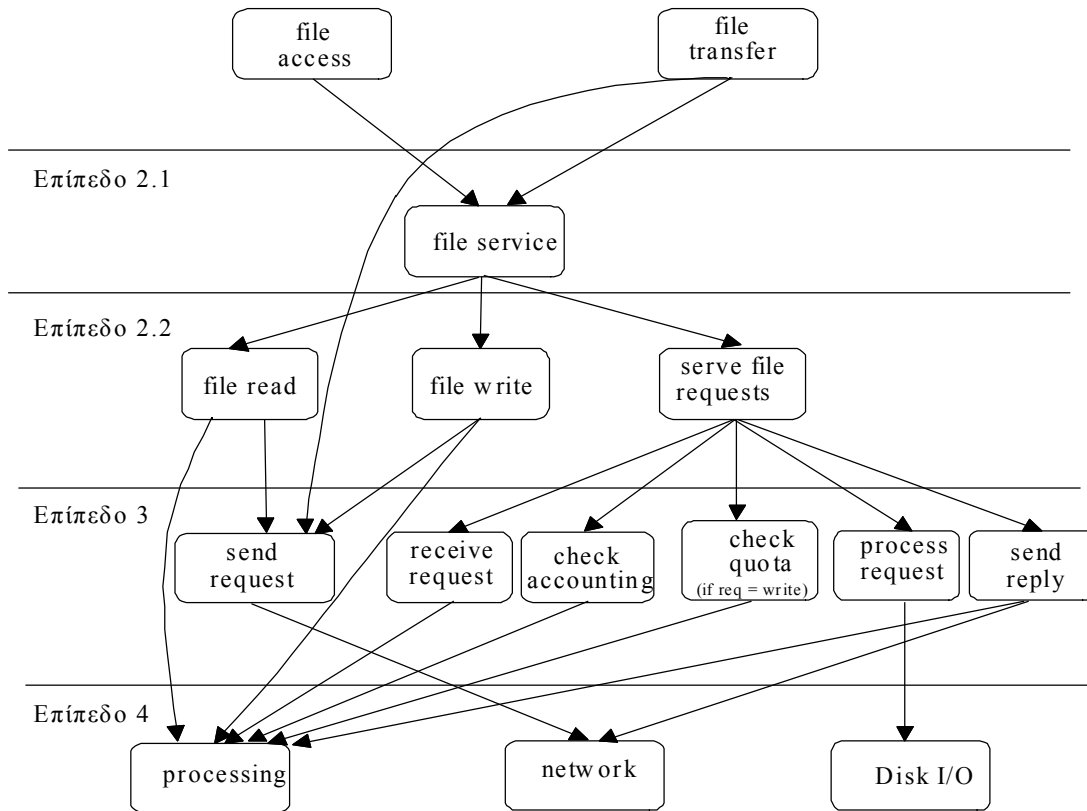
Με τη χρήση της μεθοδολογίας αυτής καταλήγουμε σε ένα σύνολο από στοιχειώδεις υπηρεσίες που αντιστοιχούν σε ανεξάρτητες βασικές λειτουργίες του κατανεμημένου συστήματος, με αποτέλεσμα τον υπολογισμό των απαιτήσεων σε μεταφορά δεδομένων, υπολογιστική και αποθηκευτική ικανότητα των δομικών στοιχείων του κατανεμημένου συστήματος. Αυτό μας επιτρέπει να σχεδιάσουμε με μεγαλύτερη ευκολία την αρχιτεκτονική κάθε περιοχής και να επιλέξουμε τα κατάλληλα πρωτόκολλα για τα 7 επίπεδα του OSI, ώστε να καλύπτουν τις απαιτήσεις του χρήστη.

Η ανάλυση που έχει γίνει μέχρι τώρα έχει αναδείξει τέσσερα επίπεδα υπηρεσιών που αντιστοιχούν στους κάτωθι βαθμούς αφαίρεσης:

1. το επίπεδο 1 αντιστοιχεί στο user (ή application) interface του συστήματος. Εδώ βρίσκονται όλες οι υπηρεσίες που είναι άμεσα προσπελάσιμες από το χρήστη και τις εφαρμογές που χρησιμοποιούν το σύστημα.
2. το επίπεδο 2 αντιστοιχεί στις υπηρεσίες του συστήματος, εδώ δηλαδή φαίνονται, σε πολύ υψηλό επίπεδο αφαίρεσης, οι υπηρεσίες που το σύστημα διαθέτει για τη διαχείριση πόρων, όπως π.χ. η υπηρεσία διαχείρισης αρχείων (file service), η υπηρεσία διαχείρισης διεργασιών (process manager), η υπηρεσία διαχείρισης ονομάτων (name service) κ.ο.κ. Το επίπεδο 2 χωρίζεται σε δύο υποεπίπεδα, τα υποεπίπεδα 2.1 και 2.2 που αντιστοιχούν στις γενικές υπηρεσίες του συστήματος (π.χ file service) και στις πιο εξειδικευμένες λειτουργίες που μπορεί να παρέχει κάθε γενική υπηρεσία (π.χ file read, file write).
3. στο επίπεδο 3 βρίσκονται οι διαδικασίες που υλοποιούν τις υπηρεσίες του δευτέρου επιπέδου. Οι διαδικασίες αυτές αντιστοιχούν κυρίως σε υψηλού επιπέδου διαδικασίες βιβλιοθηκών του συστήματος αλλά και σε διαδικασίες που δεν παρέχονται σε βιβλιοθήκες, αλλά είναι ειδικά γραμμένες για τη διεκπεραίωση κάποιας λειτουργίας.
4. το επίπεδο 4 αντιστοιχεί στους φυσικούς πόρους που μπορεί να διαχειρίζεται το σύστημα, όπως επεξεργαστές, δίσκοι, φυσικές συνδέσεις μεταξύ nodes κ.λπ. Το επίπεδο αυτό περιλαμβάνει έτσι τις πλέον στοιχειώδεις υπηρεσίες που προσφέρει ένα υπολογιστικό σύστημα και έτσι δεν μπορεί παρά να είναι το τελευταίο επίπεδο ανάλυσης των υπηρεσιών.

Στο σχήμα 3 φαίνονται αναλυμένες οι υπηρεσίες *προσπέλασης αρχείων (file access)* και *μεταφοράς αρχείων (file transfer)*. Και οι δύο ανήκουν στη γενικότερη *υπηρεσία αρχείων (file service)* του συστήματος, η οποία παρέχει τους εξυπηρετές αρχείων καθώς και δύο είδη υπηρεσιών για πελάτες (ανάγνωση και εγγραφή αρχείων). Στο τρίτο επίπεδο φαίνονται τα διακριτά βήματα που ακολουθούνται από τους πελάτες ή τους εξυπηρετές για να διεκπεραιωθούν οι λειτουργίες του επιπέδου 2.2, ενώ στο τέταρτο επίπεδο απεικονίζονται οι φυσικοί πόροι που απαιτούνται από κάθε υπηρεσία ανώτερου επιπέδου.

Επίπεδο 1



Σχήμα 3 - Απεικόνιση των Επιπέδων των Υπηρεσιών

B. Architecture Definition Tool (ADT)

Το ADT συνθέτει ένα γενικευμένο καταναμημένο περιβάλλον, το οποίο αποτελείται από **περιοχές (locations)** συνδεδεμένες μεταξύ τους. Ο διαχωρισμός σε περιοχές γίνεται με χωροταξικά κριτήρια. Το ADT δέχεται το σύνολο των στοιχειωδών υπηρεσιών (SIBs) που προδιαγράφουν τις ανάγκες του χρήστη από το URAM και είναι υπεύθυνο για τη διαμόρφωση των λύσεων που θα κληθούν να καλύψουν τις ανάγκες αυτές. Η κάθε λύση που διαμορφώνεται από το ADT πρέπει να περιλαμβάνει:

- μία πρόταση σχεδιασμού των καταναμημένων εφαρμογών που εκτελούνται στα πλαίσια του καταναμημένου συστήματος (π.χ. βάσεις δεδομένων, τράπεζες πληροφοριών διαφόρων τύπων, όπως φωνή, ψηφιακή εικόνα, δεδομένα, ηλεκτρονικό ταχυδρομείο). Η λειτουργία αυτή περιλαμβάνει την τοποθέτηση των αρχείων και των εφαρμογών στις πιο κατάλληλες τοποθεσίες (file and application placement). Η λειτουργία αυτή καλύπτεται από το υποσύστημα APT (Application Placement Tool) του ADT.
- μία πρόταση για τη σύνθεση του δικτυακού υποστρώματος πάνω στο οποίο θα λειτουργούν οι εφαρμογές. Στην πρόταση αυτή συμπεριλαμβάνονται η επιλογή των πρωτοκόλλων με τα οποία οι εφαρμογές θα επικοινωνούν, προκειμένου να έχουμε τη βέλτιστη δυνατή απόδοση και η επιλογή των φυσικών πόρων (μονάδων επεξεργασίας, μονάδων αποθήκευσης και γραμμών επικοινωνίας) που θα απαρτίζουν το καταναμημένο σύστημα. Η λειτουργία αυτή καλύπτεται από το υποσύστημα RPT (Resource Placement Tool) του ADT.

Λόγω των πολλών παραμέτρων που υπεισέρχονται στο σχεδιασμό των καταναμημένων συστημάτων και των πολλών εναλλακτικών λύσεων που μπορούν να υιοθετηθούν σε κάθε περίπτωση, είναι αδύνατο να χρησιμοποιηθούν "σκληρά" μαθηματικά μοντέλα περιγραφής των αναγκών του χρήστη (γνωστά στην διεθνή βιβλιογραφία ως *normatif* και *prescriptif*), δηλαδή περιγραφές των οποίων όλα τα δεδομένα είναι αριθμητικά, συνδέονται με μαθηματικές σχέσεις και συνοδεύονται από μια συνάρτηση της οποίας ζητείται η βελτιστοποίηση σύμφωνα με κάποια κριτήρια ([23]). Στην περίπτωση μας αυτά τα μοντέλα θα απαιτούσαν είτε απαγορευτικούς χρόνους υπολογισμού, είτε επίπονη σύλληψη και σχεδιασμό από την πλευρά του χρήστη, δηλαδή αναλυτική περιγραφή με αριθμητικές σχέσεις των εργασιών που πρέπει να επιτελεστούν. Επίσης, είναι δύσκολο να βρεθεί μια συνάρτηση περιγραφής της τοπολογίας του δικτύου, της οποίας να ζητείται η βελτιστοποίηση. Το ADT έτσι χρησιμοποιεί εμπειρικές μεθόδους, βασισμένες σε συμβολικό υπολογισμό, ο οποίος υποστηρίζεται από εργαλεία που είναι ενσωματωμένα στον κινητήρα λογισμού (*inference engine*) του. Ο κινητήρας λογισμού του ADT δεν είναι, στην παρούσα φάση, εφοδιασμένος με εργαλεία που θα αντιμετωπίσουν την ύπαρξη *ασαφειών, πλεονασμών, παραλείψεων και αντιθέσεων*, θεωρώντας ότι οι καταστάσεις αυτές έχουν αντιμετωπισθεί στο τμήμα URAM του Manager.

Η λειτουργία του ADT χωρίζεται σε δύο *φάσεις*, κάθε μία από τις οποίες αντιστοιχεί σε μια σειρά ενεργειών εννοιολογικά ή λειτουργικά συνδεδεμένων μεταξύ τους και ο τερματισμός μίας φάσης έχει ως αποτέλεσμα την έναρξη (πυροδότηση) κάποιας άλλης.

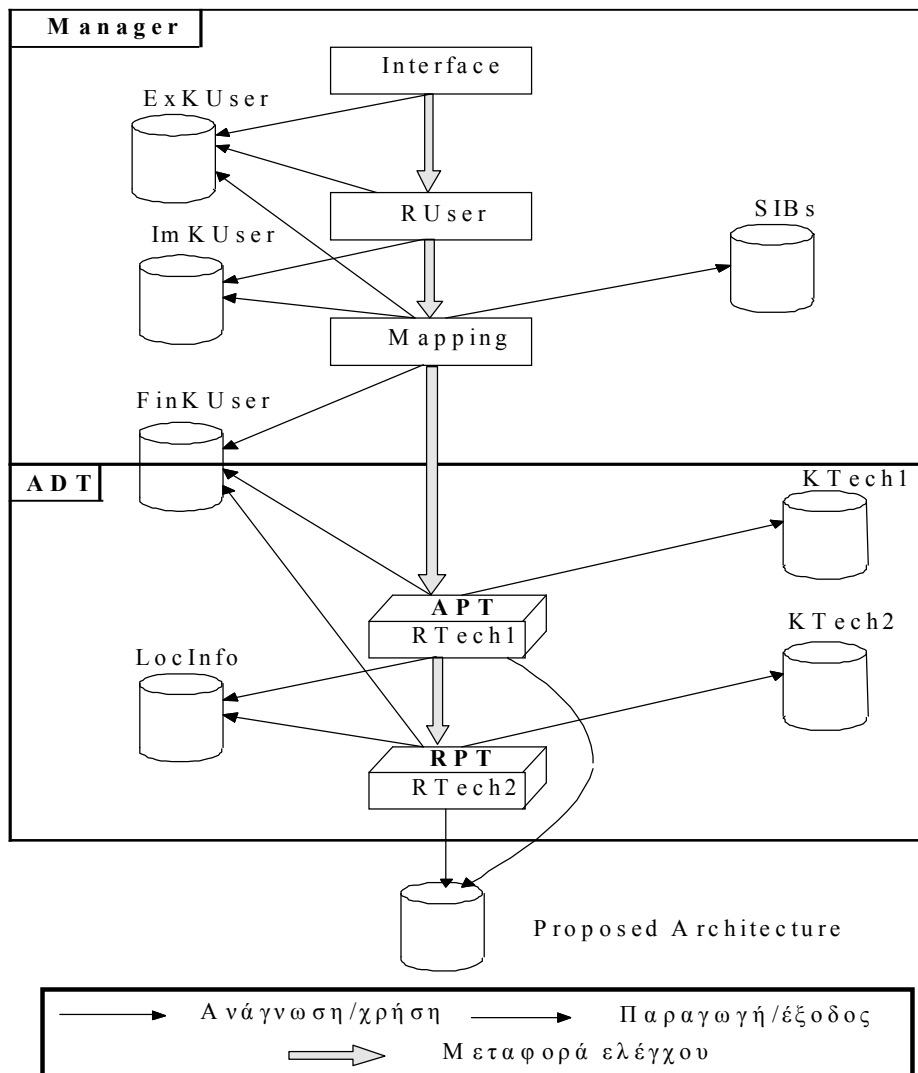
Η πρώτη φάση λειτουργίας του ADT συνίσταται στην ενεργοποίηση του υποσυστήματος APT, το οποίο είναι υπεύθυνο για την τοποθέτηση των εφαρμογών και των δεδομένων που οι εφαρμογές χρησιμοποιούν σε περιοχές. Η τοποθέτηση γίνεται κατά τρόπο ώστε η διακίνηση δεδομένων στο δίκτυο να ελαχιστοποιείται και τα αρχεία να βρίσκονται όσο το δυνατόν πλησιέστερα στις περιοχές στις οποίες χρησιμοποιούνται. Το υποσύστημα APT παραλαμβάνει τη *FinKUser*, όπως αυτή έχει διαμορφωθεί από το Manager και σύμφωνα με τις απαιτήσεις του χρήστη για τις *Front End* εφαρμογές καθορίζει και τοποθετεί τις *Back End* εφαρμογές και τα αντίστοιχα δεδομένα εφαρμόζοντας σ' αυτή τους κανόνες *RTech1*, λαμβάνοντας υπόψη τη γνώση που περιέχεται στη βάση γνώσεων *KTech1*, η οποία περιγράφει τα κριτήρια βάσει των οποίων θα ληφθούν οι αποφάσεις. Το APT αποφασίζει αν η *Back End* εφαρμογή θα είναι καταναμημένη ή όχι, σε ποια περιοχή θα τοποθετηθεί αυτή ή τα τμήματά της, καθώς και θα τοποθετηθούν τα αντίστοιχα δεδομένα, όπως επίσης αν θα χρησιμοποιηθούν ομοιότυπα ή αντίγραφα. Στην περίπτωση που η εφαρμογή είναι μια βάση δεδομένων το APT δέχεται τις προδιαγραφές του χρήστη για τα εργαλεία προσπέλασης σε αυτή και σχεδιάζει και τοποθετεί στις κατάλληλες περιοχές τον πυρήνα της βάσης. Επιπλέον τοποθετεί τα αντίστοιχα δεδομένα στην καταλληλότερη περιοχή.

Το αποτέλεσμα της εφαρμογής των κανόνων *RTech1* είναι η βάση γνώσης *LocInfo* (**L**ocation **I**nformation) η οποία περιέχει πληροφορίες για την τοποθεσία εφαρμογών και των αντίστοιχων δεδομένων στο καταναμημένο σύστημα. Οι πληροφορίες που αφορούν τις εφαρμογές έχουν λάβει πλέον την τελική τους μορφή, οπότε προωθούνται στην έξοδο του ADT, η οποία ονομάζεται *ProposedArchitecture*. Στην έξοδο του ADT προωθείται επίσης και η βάση γνώσης *LocInfo*, η οποία θα χρησιμοποιηθεί από το Manager για την εκτίμηση των αποτελεσμάτων του PET και την παρουσίαση της τελικής λύσης στο χρήστη.

Η δεύτερη φάση λειτουργίας του ADT, η εκτέλεση της οποίας αρχίζει αμέσως μετά την περάτωση της πρώτης, συνίσταται στην ενεργοποίηση του υποσυστήματος RPT, το οποίο είναι υπεύθυνο για την σχεδίαση του δικτυακού υποστρώματος λαμβάνοντας υπόψη τις βάσεις γνώσεων *FinKUser*, *LocInfo* και *KTech2*. Σε αυτό περιλαμβάνεται ο σχεδιασμός των υποσυστημάτων των περιοχών και των μεταξύ τους διασυνδέσεων, όπου αυτό κρίνεται απαραίτητο. Στον σχεδιασμό

των περιοχών καθορίζονται οι υπολογιστικοί κόμβοι και οι τερματικοί σταθμοί που διατίθενται, η αποθηκευτική δυνατότητα που προσφέρεται και τα επικοινωνιακά στοιχεία, που χρησιμοποιούν για να επικοινωνούν οι υπολογιστικοί κόμβοι και οι τερματικοί σταθμοί κάθε περιοχής (με την χρήση και τον συνδυασμό πρωτοκόλλων σύμφωνα με την στρωματοποιημένη αρχιτεκτονική ISO/OSI). Μετά τον σχεδιασμό των περιοχών το RPT προσπαθεί να τις διασυνδέσει με τον βέλτιστο δυνατό τρόπο. Οι τεχνικές και οι γνώσεις για την επιλογή πρωτοκόλλων και τοποθέτηση πόρων σε ένα καταναμημένο υπολογιστικό σύστημα φυλάσσεται στο σώμα κανόνων RTech2 και τη βάση γνώσεων KTech2 αντίστοιχα. Η επιλογή γίνεται έτσι ώστε να καλύπτεται το σύνολο των αναγκών του χρήστη, χωρίς να γίνεται σπατάλη υπολογιστικών ή επικοινωνιακών πόρων. Οι πληροφορίες που αφορούν την μορφή του δικτυακού υποστρώματος τοποθετούνται στην βάση γνώσεων *Proposed Architecture*, που αποτελεί την έξοδο του ADT και μαζί με τις πληροφορίες για τις καταναμημένες εφαρμογές προωθούνται στο PET.

Η συνολική αρχιτεκτονική του Manager/ADT που περιγράφηκε στις προηγούμενες παραγράφους, απεικονίζεται στο σχήμα 4.



Σχήμα 4 - Αρχιτεκτονική Manager και ADT

Γ. Performance Evaluation Tool (PET)

Το PET είναι υπεύθυνο για τον υπολογισμό των παραμέτρων, οι οποίες προσδιορίζουν την απόδοση της αρχιτεκτονικής που προτείνει το ADT για το κατανεμημένο σύστημα, όπως ο βαθμός χρησιμοποίησης των στατικών δομικών στοιχείων που συνθέτουν το κατανεμημένο σύστημα (utilisation) και η ανταπόκριση τους (response time) στις αιτήσεις για επεξεργασία, μεταφορά και ανάκτηση δεδομένων από μονάδες αποθήκευσης. Το PET τροφοδοτείται από το υποσύστημα επικοινωνίας (CM) του Manager με τα αποτελέσματα που παράγονται από το ADT. Σε αυτά περιλαμβάνονται:

- α. τα αποτελέσματα που προκύπτουν από το RPT για την περιγραφή επικοινωνιακών στοιχείων, υπολογιστικών κόμβων, μονάδων αποθήκευσης και των μεταξύ τους συσχετίσεων και παρέχουν μια στατική περιγραφή του δικτυακού υποστρώματος.
- β. η περιγραφή των εφαρμογών, που προδιαγράφονται από το χρήστη και ο συνδυασμός τους με το δικτυακό υπόστρωμα, όπως παράγεται από APT. Σε αυτή περιλαμβάνονται οι διάφορες **αιτήσεις χρηστών** για επεξεργασία δεδομένων από εφαρμογές, που εκτελούνται σε συγκεκριμένους κόμβους και οι αιτήσεις για προσπέλαση δεδομένων από συσκευές αποθήκευσης. Καθεμία από αυτές περιγράφεται από ποσοτικές και χρονικές παραμέτρους, όπως η ποσότητα των δεδομένων που επεξεργάζονται ή μεταφέρονται, οι χρόνοι έναρξης και περάτωσης των διαφόρων αιτήσεων καθώς και η κατανομή τους στο χρόνο εκτέλεσης του μοντέλου προσομοίωσης. Οι αιτήσεις, σε συνδυασμό με τις παραμέτρους τους, αντιστοιχούν στην δυναμική περιγραφή της λειτουργίας του κατανεμημένου συστήματος και καθορίζουν τις απαιτήσεις των εφαρμογών από κάθε στοιχείο του δικτυακού υποστρώματος και τις χρονικές στιγμές ενεργοποίησής του.

Με βάση τα στοιχεία που του παρέχονται, το PET πρέπει να διαμορφώσει ένα περιβάλλον προσομοίωσης για να εκτιμήσει την απόδοση της προτεινόμενης λύσης. Για να μπορέσει να σχηματίσει το PET το περιβάλλον αυτό χρησιμοποιεί μια σειρά από αναπαραστάσεις των δομικών στοιχείων του κατανεμημένου συστήματος, στα πλαίσια μίας αντικειμενοστραφούς (object-oriented) μεθοδολογίας. Στο PET ορίζεται ένα σύνολο από κλάσεις (classes) που αντιστοιχούν σε κάθε τύπο δομικού στοιχείου με σκοπό την περιγραφή των βασικών χαρακτηριστικών του. Η περιγραφή κάθε επιπέδου του OSI, καθώς και η περιγραφή των αιτήσεων χρηστών, των εφαρμογών, των υπολογιστικών κόμβων και των συσκευών αποθήκευσης αντιστοιχεί σε μια διαφορετική κλάση ομοιοτύπων. Κάθε κλάση μπορεί να έχει και υποκλάσεις, αν είναι απαραίτητο να ορισθούν υποκατηγορίες στην κατηγορία δομικών στοιχείων που αυτή η κλάση περιγράφει, π.χ. για τα πρωτόκολλα με και χωρίς σύνδεση (connection oriented και connectionless) ενός OSI επιπέδου. Τα πολύπλοκα δομικά στοιχεία, όπως τα επικοινωνιακά στοιχεία που αναπτύσσονται με βάση τη στρωματοποιημένη αρχιτεκτονική του OSI, περιγράφονται μέσω σύνθετων ομοιοτύπων (compound objects).

Εκτός από το στατικό μοντέλο, πρέπει να περιγραφεί και η δυναμική του συμπεριφορά, ο τρόπος δηλαδή με τον οποίο αλληλεπιδρούν τα δομικά στοιχεία. Τα δομικά στοιχεία αλληλεπιδρούν μέσω αποστολής *μηνυμάτων*, τα οποία αντιστοιχούν σε αιτήσεις εξυπηρέτησης και προκαλούν την ενεργοποίηση του παραλήπτη. Ο παραλήπτης οφείλει να εξυπηρετήσει την αίτηση, αποστέλλοντας ενδεχομένως μηνύματα σε άλλα δομικά στοιχεία. Την έναρξη της διαδικασίας αποστολής μηνυμάτων προκαλούν οι αιτήσεις χρηστών, με βάση τις χρονικές τους παραμέτρους.

Το PET υποδέχεται κάθε προτεινόμενη αρχιτεκτονική από το ADT σαν ένα σύνολο από δομικά στοιχεία με τις αντίστοιχες παραμέτρους μέσω του υποσυστήματος επικοινωνίας του Manager. Μετά την υποδοχή της προτεινόμενης λύσης ακολουθεί η γένεση των κατάλληλων

ομοιοτύπων, με τις αντίστοιχες τιμές στις παραμέτρους των ομοιοτύπων κάθε κλάσης. Αμέσως μετά αρχίζει η διαδικασία προσομοίωσης βασισμένη σε τεχνικές διακριτού χρόνου (discrete event).

Το PET υπολογίζει τα χαρακτηριστικά εκείνα που θα επιτρέψουν την εκτίμηση της απόδοσης της προτεινόμενης αρχιτεκτονικής. Τέτοια είναι ο βαθμός χρησιμοποίησης (utilisation) και ο μέσος χρόνος απόκρισης (mean response time) του κάθε δομικού στοιχείου του κατανεμημένου συστήματος καθώς και η μέση καθυστέρηση (average delay) και ο ρυθμός μετάδοσης (throughput) για την ανταλλαγή δεδομένων. Επίσης μπορούν να υπολογιστούν παράμετροι για την απόδοση κάθε περιοχής αλλά και η συνολική απόδοση του συστήματος, όπως ρυθμός μετάδοσης προς καθυστέρηση μετάδοσης (throughput/delay), φόρτος προς καθυστέρηση (load/delay) και χρόνος απόκρισης προς φόρτο (response time/load).

Δ. Εξασφάλιση επικοινωνίας μεταξύ ADT και PET

Όταν το ADT διαμορφώνει μία αρχιτεκτονική πρόταση αυτή θα πρέπει να τροφοδοτηθεί στο PET προκειμένου να εκτιμηθεί η απόδοσή της. Η τροφοδότηση αυτή δεν μπορεί να είναι άμεση, δηλαδή η πρόταση δεν μπορεί να εισαχθεί στο PET ως έχει, γιατί το ADT και το PET αναπαριστούν τη γνώση κατά διαφορετικό τρόπο. Οι διαφορετικές μορφές αναπαράστασης οφείλονται στα παρακάτω:

1. το ADT και το PET, λόγω ειδικών απαιτήσεων που παρουσιάζει το καθένα, αναπτύσσονται σε διαφορετικές γλώσσες προγραμματισμού και έτσι η εσωτερική αναπαράσταση και δομή των δεδομένων που περιγράφουν ένα αντικείμενο στο ADT δεν είναι κατ' ανάγκην ίδια με αυτή που θα υπάρχει στο PET. Το ADT αναπτύσσεται σε Prolog και το PET σε MODSIM ([24]) και είναι απαραίτητος, κατά συνέπεια, ένας μηχανισμός ο οποίος θα μεταφράζει τα κατηγορήματα (predicates) που παράγει η Prolog σε αντικείμενα (objects) τα οποία μπορεί να υποδεχθεί η MODSIM και θα μετασχηματίζει τις μορφές με τις οποίες η Prolog αναπαριστά τις πληροφορίες στις αντίστοιχες της MODSIM (π.χ. atoms σε strings, μετάφραση λιστών κ.λπ.)
2. το ADT και το PET ενδιαφέρονται για διαφορετικά χαρακτηριστικά των δομικών στοιχείων που συνθέτουν το κατανεμημένο σύστημα. Πιο συγκεκριμένα το ADT ενδιαφέρεται για εκείνα τα χαρακτηριστικά τα οποία:
 - θα του επιτρέψουν να επιλέξει τα καταλληλότερα δομικά στοιχεία για κάθε περίπτωση
 - θα παρουσιάσει στην έξοδό του ως αποτέλεσμα των αποφάσεων που έλαβε. Τα χαρακτηριστικά αυτά είτε θα περιγράφουν την τελική λύση, όπως αυτή θα παρουσιαστεί στο χρήστη, είτε θα δίνουν πληροφορίες στο PET για τη λειτουργική συμπεριφορά του δομικού στοιχείου.

Από την άλλη πλευρά το PET ενδιαφέρεται μόνο για τα χαρακτηριστικά εκείνα που θα του επιτρέψουν να εκτιμήσει τη συνολική απόδοση και το βαθμό χρησιμοποίησης του κάθε δομικού στοιχείου του κατανεμημένου συστήματος και για εκείνα τα χαρακτηριστικά τους, που αυτό θα πρέπει να υπολογίσει, όπως π.χ. ο βαθμός χρησιμοποίησης (utilisation) και ο μέσος χρόνος απόκρισης (mean response time). Γίνεται έτσι σαφές ότι ο μηχανισμός επικοινωνίας μεταξύ ADT και PET θα πρέπει να φροντίζει ώστε στο PET να φτάνουν μόνο τα χαρακτηριστικά των στοιχείων τα οποία είναι απαραίτητα, προκειμένου να αποφεύγονται λάθη από εσφαλμένη τροποποίηση χαρακτηριστικών στο PET και η λειτουργία του να είναι αποδοτικότερη, δεδομένου ότι θα επιφορτίζεται με μικρότερο όγκο δεδομένων.

Τέλος, ο μηχανισμός επικοινωνίας φροντίζει ώστε το PET να τροφοδοτείται κατάλληλα με τα δομικά εκείνα στοιχεία που περιγράφουν τη δυναμική λειτουργία του κατανεμημένου συστήματος,

δηλαδή τις διάφορες αιτήσεις για επεξεργασία, ανάγνωση δεδομένων από συσκευές αποθήκευσης και επικοινωνία μεταξύ περιοχών.

Κάθε μία από τις αιτήσεις αυτές θα συνοδεύεται από ποσοτικές και χρονικές παραμέτρους, οι οποίες θα καθορίζουν τις απαιτήσεις σε πόρους που προβάλλει η συγκεκριμένη αίτηση, καθώς και τις χρονικές στιγμές ενεργοποίησής της. Οι αιτήσεις αυτές αποτελούν τα γεγονότα τα οποία, μέσα στο μοντέλο προσομοίωσης, θα προκαλέσουν την ενεργοποίηση των δομικών στοιχείων που περιγράφουν την αρχιτεκτονική του κατανεμημένου συστήματος. Η τελική μορφή των αιτήσεων καθορίζεται από το APT, μια και το τμήμα αυτό του Έμπειρου Συμβουλευτικού Συστήματος επιλέγει τις τοποθεσίες στις οποίες θα τοποθετηθούν οι πόροι του κατανεμημένου συστήματος και οι εφαρμογές.

Ε. Αξιολόγηση των αποτελεσμάτων του PET

Το PET είναι αρμόδιο για την προσομοίωση των λύσεων που προτείνει το ADT και για την παραγωγή αποτελεσμάτων που προσδιορίζουν τον τρόπο και το βαθμό χρησιμοποίησης των στατικών δομικών στοιχείων που συνθέτουν το κατανεμημένο σύστημα, καθώς και την ανταπόκριση τους στις αιτήσεις για επεξεργασία, μεταφορά δεδομένων και ανάκτηση δεδομένων από μονάδες αποθήκευσης. Τα αποτελέσματα αυτά θα πρέπει να αξιολογηθούν, προκειμένου να αποφασισθεί αν το κατανεμημένο σύστημα ανταποκρίνεται ικανοποιητικά στις απαιτήσεις του χρήστη και αν κάποια στοιχεία αποτελούν σημεία συμφόρησης (bottlenecks) ή υποχρησιμοποιούνται. Η αξιολόγηση αυτή γίνεται από το κατάλληλο υποσύστημα του Manager, το οποίο χρησιμοποιώντας τη διαθέσιμη σ' αυτό γνώση για την τοποθέτηση των εφαρμογών και πόρων καθώς και τις επιλογές πρωτοκόλλων είναι σε θέση να αποφασίσει για τα σημεία του σχεδιασμού, που προκαλούν τελικά τις δυσλειτουργίες και να ζητήσει από το ADT να τροποποιήσει τις επιλογές του σε ό,τι αφορά τα σημεία αυτά. Για τον προσδιορισμό των σημείων του σχεδιασμού που απαιτούν τροποποίηση θα χρησιμοποιείται ένα σύνολο κανόνων, το *ModDesign*, που θα τροφοδοτείται με τις απαιτήσεις του χρήστη, την προτεινόμενη αρχιτεκτονική, πληροφορίες για την τοποθέτηση εφαρμογών και πόρων (βάση γνώσης *LocInfo*), καθώς και τα στοιχεία που προέκυψαν από την προσομοίωση. Χρησιμοποιείται επίσης η βάση γνώσης *DesInfo*, η οποία περιέχει πληροφορίες που χρησιμεύουν στο υποσύστημα εκτίμησης αποτελεσμάτων για να διαπιστωθεί η σταθερότητα του κατανεμημένου συστήματος κάτω από διάφορες συνθήκες λειτουργίας και να αποφασισθεί στη συνέχεια αν πρέπει να τροποποιηθεί ο σχεδιασμός δομικών στοιχείων του κατανεμημένου συστήματος.

Αν το υποσύστημα εκτίμησης αποτελεσμάτων του Manager κρίνει ότι δεν υπάρχει τροποποίηση του σχεδιασμού που να είναι σε θέση να καλύψει τις ανάγκες του χρήστη, αυτός θα ειδοποιηθεί ότι οι απαιτήσεις του δεν μπορούν να ικανοποιηθούν. Από το σημείο αυτό και μετά ο χρήστης θα πρέπει είτε να τροποποιήσει τις απαιτήσεις του με τη βοήθεια του Manager, είτε, με τη βοήθεια καταλλήλων εργαλείων, να εισάγει νέα γνώση στο σύστημα σχετικά με τεχνολογικές προόδους, έτσι ώστε να εξετασθεί εκ νέου αν είναι δυνατό να καλυφθούν οι ανάγκες του.

III. ΑΝΑΠΤΥΞΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

Το Έμπειρο Συμβουλευτικό Σύστημα διατηρεί τα βασικά χαρακτηριστικά ενός "έξυπνου" συστήματος. Η σχεδιάσή του δεν ακολουθεί ντετερμινιστικά πρότυπα αλλά βασίζεται στην αξιοποίηση εμπειρικών μεθόδων για επιλεκτική χρήση αλγορίθμων σχεδιασμού, με αποτέλεσμα την αναγκαιότητα αξιολόγησης των αποτελεσμάτων, όχι με τις κλασσικές τεχνικές επαλήθευσης γνώσης (knowledge validation) ([25], [26], [27]), αλλά με χρήση τεχνικών προσομοίωσης. Αυτό αποτελεί την πρώτη διαφοροποίηση από τα υπάρχοντα έμπειρα συστήματα.

Μία δεύτερη, ίσως κρισιμότερη, είναι η απουσία διαλόγου με το χρήστη στη διάρκεια της φάσης ελέγχου του σχεδιασμού, γεγονός που στερεί τον κινητήρα λογισμού από μια έξωθεν σημαντική βοήθεια. Το σύστημα δεν επωφελείται από την ενδεχόμενη ανάδραση (feedback), που θα προσέφερε ο χρήστης. Ο βρόχος μεταξύ χρήστη και έμπειρου συστήματος, υποκαθίσταται από τον κύκλο επικοινωνίας ανάμεσα στο ADT και το PET και επιτελείται εσωτερικά στο Έμπειρο Συμβουλευτικό Σύστημα. Η επανάληψη του κατευθύνεται από το Manager, που είναι υπεύθυνος για την ενεργοποίηση του ADT και του PET.

Ο χρόνος απόκρισης του Έμπειρου Συμβουλευτικού Συστήματος, δηλαδή ο χρόνος σχεδιασμού του κατανεμημένου συστήματος, δεν μπορεί να υπερβαίνει ορισμένα όρια. Αυτό αποτελεί ένα επιπλέον χαρακτηριστικό, που πρέπει να ληφθεί υπόψη στο σχεδιασμό τόσο του έμπειρου συστήματος όσο και του περιβάλλοντος προσομοίωσης.

Εξαιτίας της πολυπλοκότητας του Έμπειρου Συμβουλευτικού Συστήματος αποφασίστηκε η ανάπτυξη του να γίνει σε δύο στάδια. Το πρώτο που βρίσκεται τώρα σε εξέλιξη ([28]), περιλαμβάνει το σχεδιασμό και την ανάπτυξη ενός πρωτότυπου συστήματος για να προσφέρει γνώση για την περαιτέρω ανάπτυξη των τμημάτων του (Manager, ADT, PET) και να επιλύσει τεχνικά ζητήματα, όπως π.χ. η επικοινωνία ανάμεσα στο ADT και το PET. Στο δεύτερο στάδιο περιλαμβάνεται η ολοκλήρωση του συστήματος και ανάπτυξη μηχανισμών για την προσθήκη τεχνογνωσίας στο σύστημα.

Στα πλαίσια του πρωτοτύπου γίνονται παραδοχές για όλα τα τμήματα του Έμπειρου Συμβουλευτικού Συστήματος με σκοπό τον περιορισμό των προβλημάτων που εξετάζονται, ώστε να είναι δυνατή η ανάπτυξη των μεθοδολογιών που θα αποτελέσουν τη βάση για τη λειτουργία του συστήματος. Οι παραδοχές έχουν σκοπό να περιορίσουν σε έκταση τα εναλλακτικά σενάρια που πρέπει να εξεταστούν κατά τη σχεδίαση ενός υπολογιστικού συστήματος χωρίς να θίξουν τη γενικότητα των μεθοδολογιών που αναπτύσσονται. Οι παραδοχές αυτές συνίστανται στα παρακάτω:

- το πρωτότυπο σύστημα προσφέρει στο χρήστη βασικές υπηρεσίες στο περιβάλλον των δικτύων, όπως η μεταφορά και διαχείριση αρχείων ή το ηλεκτρονικό ταχυδρομείο. Οποιαδήποτε άλλη εφαρμογή προδιαγράφεται από το χρήστη. Βασικός στόχος κατά την ανάλυση των αναγκών του χρήστη είναι ο εντοπισμός των στοιχειωδών υπηρεσιών (SIBs), που είναι ικανές να περιγράψουν οποιαδήποτε υπηρεσία προσφέρεται από ένα κατανεμημένο σύστημα. Το περιβάλλον επικοινωνίας που αναπτύσσεται, υποχρεώνει το χρήστη να τροφοδοτήσει το Έμπειρο Συμβουλευτικό Σύστημα με πληροφορίες χωρίς να του παρέχει εναλλακτικούς τρόπους περιγραφής του σχεδιαζόμενου συστήματος.
- δεν προβλέπεται βελτιστοποιημένος σχεδιασμός των κατανεμημένων εφαρμογών από το APT. Ο χρήστης του συμβουλευτικού συστήματος επιλέγει ο ίδιος τις περιοχές όπου εκτελούνται οι εφαρμογές, καθώς και τις περιοχές στις οποίες αποθηκεύονται τα δεδομένα των εφαρμογών.
- στα πλαίσια του RPT βασικός στόχος είναι η ανάπτυξη των κανόνων που προδιαγράφουν το δικτυακό υπόστρωμα. Επειδή το σύνολο των αρχιτεκτονικών και των πρωτοκόλλων είναι εκτεταμένο, αποφασίστηκε η ανάπτυξη του RPT να βασιστεί σε πρωτόκολλα τοπικών δικτύων τόσο σε επίπεδο περιοχής, όσο και για τη διασύνδεση περιοχών. Τα επίπεδα εφαρμογής, παρουσίασης και συνόδου του OSI περιγράφονται από το πρωτόκολλο RPCs (CCITT X.410), που χρησιμοποιείται από τα περισσότερα κατανεμημένα συστήματα. Για την περιγραφή των χαμηλότερων επιπέδων χρησιμοποιούνται πρωτόκολλα της οικογένειας ARPANET και IEEE 802. Σκοπός μας είναι η επιλογή των κατάλληλων δομών αναπαράστασης γνώσης για την περιγραφή των δομικών στοιχείων του δικτυακού

υποστρώματος και η διαμόρφωση κανόνων για την περιγραφή βασικών αρχών στην επιλογή των δομικών στοιχείων, όπως π.χ. τύπων πρωτοκόλλων (με ή χωρίς σύνδεση, Ethernet ή token passing).

- βασικός στόχος στην υλοποίηση του PET είναι να αναπτυχθεί ένα περιβάλλον προσομοίωσης των δομικών στοιχείων και των χαρακτηριστικών τους και να υπολογίζεται η συνολική απόδοση του κατανεμημένου συστήματος κάτω από διαφορετικές συνθήκες φόρτου. Το περιβάλλον αυτό θα πρέπει να περιλαμβάνει μοντέλα, που να περιγράφουν τα δομικά στοιχεία του κατανεμημένου περιβάλλοντος, που υποστηρίζει το RPT. Η ανάπτυξη του PET πραγματοποιείται με τη γλώσσα προσομοίωσης MODSIM.

Η ανάπτυξη όλων των τμημάτων του Έμπειρου Συμβουλευτικού Συστήματος γίνεται αξιοποιώντας πόρους του δικτύου SUN του Τμήματος Πληροφορικής του Πανεπιστημίου Αθηνών.

IV. ΣΥΜΠΕΡΑΣΜΑΤΑ

Στην εργασία αυτή παρουσιάσαμε μία μεθοδική προσέγγιση στην ανάπτυξη λογισμικού για αυτοματοποιημένο σχεδιασμό δικτύων και κατανεμημένων εφαρμογών. Στόχος του Έμπειρου Συμβουλευτικού Συστήματος είναι η τοποθέτηση των εφαρμογών και των αρχείων που αυτές διαχειρίζονται στις κατάλληλες περιοχές, ενώ παράλληλα επιδιώκεται βέλτιστη κατανομή των υπολογιστικών, αποθηκευτικών και επικοινωνιακών πόρων. Η ανάλυση των αναγκών του χρήστη γίνεται κατά τρόπο που να επιτρέπει τη σαφή και πλήρη περιγραφή των απαιτήσεων από το κατανεμημένο σύστημα.

V. ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] G. F. Coulouris και J. Dollimore, "Distributed Systems: Concepts and Design", Addison Wesley Publishing Company, 1988.
- [2] CCITT, "Open Distributed Systems", CCITT X.900 series, Geneva, Απρίλιος, 1992.
- [3] A. S. Tanenbaum, "Modern Operating Systems", Prentice-Hall International Editions, 1992.
- [4] G. Popek και B. Walker, "The LOCUS Distributed System Architecture", MIT Press, 1985.
- [5] A. S. Tanenbaum, R. van Renesse, S. J. Mullender et al., "Experiences with the Amoeba Distributed Operating System", Communications of the ACM, Δεκέμβριος 1990.
- [6] R. F. Rashid, "From RIG to Accent to Mach: the evolution of a network operating system", Proceedings of the ACM/IEEE Computer Society Fall Joint Conference, ACM, Νοέμβριος 1986.
- [7] Sape Mullender, "Distributed Systems", ACM PressFrontier Series, 1989.
- [8] S. Ceri και L. Tanca, "Expert Design of Local Area Networks", IEEE Expert, Οκτώβριος 1990
- [9] K. S. Leung και M.H. Wong, "An Expert-System Shell Using Structured Knowledge - An Object-Oriented Approach", IEEE Computer, Μάρτιος 1990
- [10] F. Andre, A. Joubelt και S. Kuppaswami, "An Intelligent Simulation Environment for Evaluating Distributed Algorithms and Architectures", Artificial Intelligence, Expert Systems and Languages in Modeling and Simulation, Elsevier Science Publishers B.V. (North-Holland), 1988
- [11] CACI Products Company, "A Quick Look at COMNET II.5"
- [12] CACI Products Company, "A Quick Look at NETWORK II.5"
- [13] CACI Products Company, "A Quick Look at LANNET II.5"

- [14] R. L. Bagrodia και C. Shen, "MIDAS: Integrated Design and Simulation of Distributed Systems", IEEE Transactions on Software Engineering, τεύχος 17, αριθ. 10, Οκτώβριος 1991
- [15] COMDISCO Systems Inc, "BONeS Modeling Reference Guide", Ιανουάριος 1991
- [16] B. W. Wah, "File Placement in Distributed Computer Systems", IEEE Computer, Ιανουάριος 1984
- [17] P. M. Apers, "Data Allocation in Distributed Computer Systems", ACM Transaction on Database Systems, Σεπτέμβριος 1988
- [18] A. B. Gerbessiotis, J. G. Kollias και M. Hatzopoulos, "Towards an Optimal Allocation of Fragments of Relations", Research into Networks and Distributed Applications, Elsevier Science Publishers B.V. (North-Holland), 1988
- [19] B. Gavish και H. Pirkul, "Computer and Database Locations in Distributed Computer Systems", IEEE Transactions on Computers, τεύχος c-35, αριθ. 7, Ιούλιος 1986
- [20] B. Gavish, "Optimization Models for Configuring Distributed Computer Systems", IEEE Transactions on Computers, τεύχος c-36, αριθ. 7, Ιούλιος 1987
- [21] Κ. Βασιλάκης, Π. Γεωργιάδης, Δ. Λέλης, Δ. Μουζάκης και Μ. Νικολαΐδου, "Σχεδιασμός Κατανεμημένων Συστημάτων με Μεθόδους Τεχνητής Νοημοσύνης και Τεχνικές Προσομοίωσης- Μέρος Α'", Εσωτερική Τεχνική Έκθεση, σελ. 1-40, Πανεπιστήμιο Αθηνών-Τμήμα Πληροφορικής, Νοέμβριος 1992.
- [22] Study Group XI, Report R30 - CCITT Q.800 series, Geneva, Νοέμβριος 1990.
- [23] C. Riveline, "L'enseignement du dur et l'enseignement du mou", Gerer et comprendre, Δεκέμβριος 1986, σελ. 42-45.
- [24] CACI Products Company, "MODSIM II Reference Manual"
- [25] C. Assemat, A. Bonnet A. και Conceptualisation, "Coherence et validation des bases de connaissances", Huitieme journees internationales: Les systemes experts et leurs applications, Avignon 1988, σελ.299-318.
- [26] D. Fontaine, P. Le Beaux και A. Strauss., "Un systeme interactif pour le maintient de la coherence d'une base de regles", Huitiemes journees internationales: Les systemes experts et leurs applications, Avignon 1988, σελ.49-61.
- [27] A. Beauvieux, "Controler la coherence d'une base de connaissances", Huitiemes journees internationales: Les systemes experts et leurs applications, Avignon 1988, σελ.35-48.
- [28] Κ. Βασιλάκης, Π. Γεωργιάδης, Δ. Λέλης, Δ. Μουζάκης και Μ. Νικολαΐδου, "Σχεδιασμός Κατανεμημένων Συστημάτων με Μεθόδους Τεχνητής Νοημοσύνης και Τεχνικές Προσομοίωσης- Μέρος Β'", Εσωτερική Τεχνική Έκθεση, σελ. 40-200, Πανεπιστήμιο Αθηνών - Τμήμα Πληροφορικής, Φεβρουάριος 1993.