

ΠΕΡΙΕΧΟΜΕΝΑ

ΕΙΣΑΓΩΓΗ

1. CORBA (Common Object Request Broker Architecture - Αρχιτεκτονική Διαμεσολάβησης για Αιτήματα Κοινών Αντικειμένων)

- 1.1. Τι είναι η CORBA
- 1.2. CORBA αρχιτεκτονική διαχείρισης αντικειμένου (OMA), Marketing and Technical Reference Model
- 1.3. Τι παρέχει η CORBA. CORBA IDL (Interface Definition Language)
- 1.4. Κατανομή και αρχιτεκτονική
 - 1.4.1. Αντικειμενοστραφής αρχιτεκτονική
- 1.5. Διαχείριση Τύπων
- 1.6. Αναπαράσταση της OMG IDL στις γλώσσες προγραμματισμού
- 1.7. Μεσίτης Αιτούμενου Αντικειμένου (Object Request Broker, ORB)
 - 1.7.1. Παραδείγματα ORBs
- 1.8. Διαλειτουργικότητα
 - 1.8.1. Αρχιτεκτονική ORB διαλειτουργικότητας
 - 1.8.2. Υποστήριξη Inter-ORB Bridge
 - 1.8.3. General Inter-ORB Protocol (GIOP)
 - 1.8.4. Internet Inter-ORB Protocol (IIOP)
 - 1.8.5. Environment-Specific Inter-ORB Protocols (ESIOPs)
- 1.9. Το παράδειγμα της CORBA-CGI Gateway

2. DCOM (Distributed Component Object Model)

- 2.1. Εισαγωγή στο DCOM
- 2.2. Κατανεμημένες Εφαρμογές
- 2.3. Η DCOM αρχιτεκτονική
- 2.4. Συστατικά και επαναχρησιμοποίηση
- 2.5. Χαρακτηριστικά του DCOM
 - 2.5.1. Ανεξαρτησία τοποθεσίας

- 2.5.2. Ουδετερότητα γλώσσας
- 2.5.3. Διαχείριση σύνδεσης
- 2.5.4. Κλιμάκωση
- 2.5.5. Εκτέλεση
- 2.5.6. Εύρος ζώνης και εκκρεμότητα
- 2.5.7. Ασφάλεια
- 2.5.8. Εξισορρόπηση φόρτισης
- 2.5.9. Ανεκτικότητα λάθους
- 2.5.10. Ευκολία στην εκτέλεση
- 2.5.11. Ουδετερότητα πρωτοκόλλου
- 2.5.12. Ουδετερότητα πλατφόρμας
- 2.5.13. Ολοκλήρωση με άλλα Internet πρωτόκολλα
 - α. Το DCOM πάνω από εικονικά ιδιωτικά δίκτυα
 - β. Το DCOM πάνω από το Internet
 - γ. Ολοκληρώνοντας το HTML και τα κατακεμημένα υπολογιστικά συστήματα

3. DCOM και CORBA

4. DISMAN (Distributed Management)

- 4.1. Τι είναι DISMAN
- 4.2. Περιγραφή της DISMAN ομάδας εργασίας
- 4.3. Γιατί DISMAN

ΒΙΒΛΙΟΓΡΑΦΙΑ

CORBA, DCOM, DISMAN πλατφόρμες Διαχείρισης Δικτύων

Τσαγκαλίδης Ευάγγελος

MIS, Β' Εξάμηνο

Περίληψη

Η ανάγκη επικοινωνίας διαφορετικών υπολογιστικών συστημάτων και εφαρμογών με σκοπό τη συνδυασμένη εκτέλεση μιας εργασίας οδήγησε στην ανάπτυξη της αρχιτεκτονικής CORBA της OMG, του μοντέλου DCOM του ActiveX® Consortium και στον ορισμό της ομάδας εργασίας DISMAN της IETF της Internet Society ISOC. Η CORBA και το DCOM είναι δύο ανταγωνιστικά πρότυπα στο χώρο των κατανεμημένων αντικειμένων που ακολουθούν την αρχιτεκτονική πελάτη-εξυπηρετή. Τα συστήματα διαχείρισης στον παγκόσμιο ιστό άρχισαν να βασίζονται σ' αυτά τα μοντέλα αντικειμένου σε συνδυασμό με τα εργαλεία υπέρ-μέσων και επεκτάσιμων γλωσσών σήμανσης. Έτσι τα συστήματα αυτά πλεονεκτούν και για ευρυζωνικά και για πολυμεσικά δίκτυα όπου τίθεται η κατανεμημένη επεξεργασία και οι εφαρμογές πραγματικού χρόνου. Η ομάδα εργασίας DISMAN περιορίζει τη δουλειά της στο αντικείμενο των εφαρμογών κατανεμημένης δικτυακής διαχείρισης, όπου ο κύριος μηχανισμός επικοινωνίας για τη λειτουργία και τον έλεγχο είναι το SNMP. Προσπάθειά της είναι η υιοθέτηση μιας γενικής λύσης για την αξιόπιστη διαχείριση λαθών από το πρωτόκολλο SNMP, ώστε να αποφεύγεται η κάθε εφαρμογή και η κάθε συσκευή να παρέχουν τη δική τους λύση. Οι εργασίες της ομάδας βρίσκονται σε εξέλιξη.

CORBA, DCOM, DISMAN platforms for Network Management

Abstract

The need to communicate different computer systems and applications in order to cooperate in a work implementation led in developing the architecture CORBA of OMG, the model DCOM of ActiveX® Consortium and defining the working group DISMAN of IETF of Internet Society ISOC. CORBA and DCOM are the competing standards for distributed object computing which follow the client-server architecture. Web-based management systems are now starting to be based upon these object models in conjunction with hyper-media and extensible mark-up language tools. These systems have benefits for both broadband and multimedia networks where distributed processing and time-critical application are common place. The work of DISMAN working group is limited to distributed network management applications where the main communication mechanism for monitoring and control is SNMP. It intends to define a generic solution for more reliability from SNMP fault management to avoid defining for each application and device its own solution. Its work is in progress.

ΕΙΣΑΓΩΓΗ

Η αυξανόμενη πολυπλοκότητα των Η/Υ και των τηλεπικοινωνιακών δικτύων απαιτεί και την ταυτόχρονη αύξηση, τόσο της ισχύος όσο και της φιλικότητας στο χρήστη, των εργαλείων *Διαχείρισης Δικτύων* (Stallings, 1993). Διαχείριση Δικτύων σημαίνει ανάπτυξη και συντονισμός των πόρων με σκοπό την προσχεδίαση, λειτουργία, διοίκηση, ανάλυση, αξιολόγηση, σχεδιασμό και επεκτασιμότητα των επικοινωνιακών

δικτύων ώστε να επιτυγχάνονται πάντα τα επιδιωκόμενα επίπεδα υπηρεσιών, με ανεκτό κόστος και με βέλτιστη δυνατότητα (Terplan, 1999).

Το διαδίκτυο, οι προσωπικοί Η/Υ και τα τοπικά δίκτυα (LAN) οδηγούν στην κατανεμημένη λειτουργία των υπολογιστικών συστημάτων. Συστήματα τα οποία συνήθως είναι ετερογενή τόσο στο hardware όσο και στο λογισμικό. Η επικοινωνία αυτών των διαφορετικών συστημάτων και εφαρμογών με σκοπό τη συνδυασμένη εκτέλεση μιας εργασίας εισάγει αρκετές δυσκολίες. Περίπου το ένα-τρίτο των προγραμμάτων λογισμικού ακυρώνονται, ενώ μόνο το ένα-έκτο είναι επιτυχημένο. Το κόστος τους είναι δύο με τρεις φορές περισσότερο από το προβλεπόμενο και προϋπολογιζόμενο. Συστήματα μεταφοράς συχνά αποτυγχάνουν να παρέχουν όλα τα επιθυμητά χαρακτηριστικά, ενώ είναι δύσκολη και υψηλού κόστους η αλλαγή τους. Τα βασικά στοιχεία που χρειάζονται ώστε να λυθούν τα προβλήματα αυτά είναι η σταθερή υποδομή, οι αρχιτεκτονικές λογισμικού και η εξελιγμένη καθοδήγηση και προσανατολισμός (Mowbray,1997).

1. CORBA (Common Object Request Broker Architecture - Αρχιτεκτονική Διαμεσολάβησης για Αιτήματα Κοινών Αντικειμένων)

1.1. Τι είναι η CORBA

Είναι ένα σύγχρονο και δημοφιλές πρωτόκολλο για την επικοινωνία εφαρμογών, ανεξάρτητα από τον τρόπο κατασκευής τους και τον υπολογιστή στον οποίο είναι εγκατεστημένες. Θεσπίστηκε από την ομάδα OMG (Object Manager Group) μια μη κερδοσκοπική εταιρεία που ιδρύθηκε το 1989 από οκτώ μεγάλες εταιρείες (3Com Corporation, American Airlines, Canon Inc, Data General, Hewlett-Packard, Phillips Telecommunications N. V, Sun Microsystems και Unisys Corporation). Υποστηρίζεται από μια πληθώρα εκδόσεων του UNIX, αλλά εφαρμογές του υπάρχουν και σε εφαρμογές σχεδιασμένες για την οικογένεια λειτουργικών Microsoft Windows.

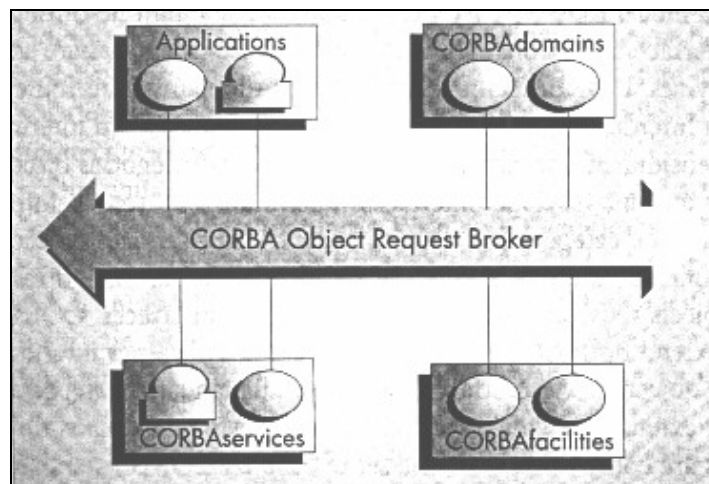
Η CORBA ακολουθεί την αρχιτεκτονική πελάτη-εξυπηρετητή και σημαντικό ρόλο σ' αυτή παίζει ο *Μεσίτης Αιτήσεων για Αντικείμενα* (Object Request Broker - ORB). Χρησιμοποιώντας την ενδιάμεση εφαρμογή ORB η εφαρμογή-πελάτης μπορεί να καλέσει μία μέθοδο (συνάρτηση) από την εφαρμογή-εξυπηρετητή η οποία μπορεί να βρίσκεται είτε στον ίδιο υπολογιστή είτε σε κάποιον άλλο που συνδέεται μέσω δικτύου. Ο ORB δέχεται την κλήση και είναι υπεύθυνος να βρει μια εφαρμογή-εξυπηρετητή που

να μπορεί να ανταποκριθεί σε αυτήν την κλήση, να καλέσει τη μέθοδο που ζήτησε ο πελάτης περνώντας τις κατάλληλες παραμέτρους και να προσφέρει το αποτέλεσμα της κλήσης στον πελάτη. Η εφαρμογή-πελάτης δεν χρειάζεται να γνωρίζει που είναι εγκατεστημένη η εφαρμογή-εξυπηρετητής, με ποια γλώσσα προγραμματισμού έχει κατασκευαστεί και με ποιο λειτουργικό σύστημα λειτουργεί.

Η CORBA έχει πολλά πλεονεκτήματα στα ετερογενή καταναμημένα υπολογιστικά συστήματα, όπως η διαπεραστικότητα θέσης και ενέργειας, η ανεξαρτησία γλώσσας και η ουδετερότητα της πλατφόρμας. Είναι αντικειμενοστραφής με πολλές δυνατότητες όπως η επαναχρησιμοποίηση. Η Interface Definition Language (IDL) της CORBA είναι ένα διεθνές πρότυπο που έγινε διεθνής γραφή στις διεπαφές λογισμικού που αναπτύχθηκε για να υλοποιήσει την αρχιτεκτονική CORBA. Καθιερώθηκε ως η de facto υποδομή για συστήματα παλιάς τεχνολογίας, εμπορικά λογισμικά και τεχνολογίες αντικειμένων.

1.2. CORBA αρχιτεκτονική διαχείρισης αντικειμένου (OMA), Marketing and Technical Reference Model

Στο σχήμα 1 απεικονίζεται η αρχιτεκτονική διαχείρισης αντικειμένου (OMA) της OMG, το βιομηχανικό πρότυπο Μοντέλου Αναφοράς για την τεχνολογία αντικειμένου. Η OMA προσδιορίζει 4 κατηγορίες διεπαφών λογισμικού για προτυποποίηση που είναι η CORBA Object Request Broker (ORB), η CORBA services, η CORBA facilities και η CORBA domains. Το κεντρικό συστατικό της αρχιτεκτονικής είναι το ORB μέσω του οποίου όλα τα αντικείμενα επικοινωνούν.



Σχήμα 1.1: CORBA αρχιτεκτονική διαχείρισης αντικειμένου (OMA), Marketing Reference Model

Το ORB λειτουργεί ως υποδομή επικοινωνίας, αναμεταδίδοντας διαφανώς τα αιτήματα για αντικείμενα μεταξύ κατανεμημένων ετερογενών υπολογιστικών περιβαλλόντων.

Η CORBA services είναι θεμελιώδης, δίνοντας τη δυνατότητα διεπαφών που είναι πλήρως εφαρμόσιμες για την ισχυροποίηση της ανάπτυξης εφαρμογών. Ένα δείγμα της CORBA services περιέχει τον κύκλο ζωής του αντικειμένου, τη γραφή συμβάντος και τις κατανεμημένες συνδιαλλαγές. Ως υπηρεσία βρίσκεται στο χαμηλότερο επίπεδο και παρέχει τη βασική, στο επίπεδο του αντικειμένου, λειτουργικότητα για τις εφαρμογές του πελάτη. Όλα τα άλλα πρότυπα διεπαφών αντικειμένου είναι είτε CORBA facilities είτε CORBA domains.

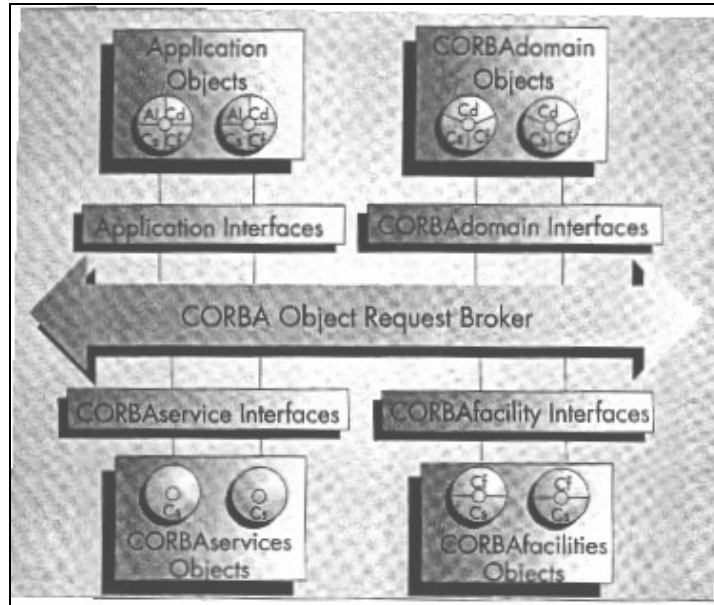
Η CORBA facilities περιλαμβάνει εκείνες τις διεπαφές που είναι διαμοιρασμένες από την πολλαπλή CORBA domains. Η CORBA facilities είναι το αμέσως υψηλότερο επίπεδο μετά τη CORBA services που παρέχει ένα σύνολο υπηρεσιών οριοθέτησης των εργασιών σε ένα κατανεμημένο περιβάλλον. Είναι πρότυπα οριζόντιας διεπαφής εστιασμένα στην διαλειτουργικότητα εφαρμογής και είναι οριζοντίως εφαρμόσιμα διαμέσου των domains. Μερικά από αυτά μπορεί να βασίζονται κατά την εκτέλεσή τους σε μια ή περισσότερες υπηρεσίες της CORBA services. Παραδείγματα των υπηρεσιών της είναι οι IDL ορισμοί στην περιοχή διεπαφής του χρήστη, τα πολυσύνθετα έγγραφα, η διαχείριση συστήματος και η ανταλλαγή δεδομένων.

Η CORBA domains περιλαμβάνει κάθετες υπηρεσίες σε εξειδικευμένες περιοχές της αγοράς, όπως υγείας και οικονομικές υπηρεσίες. Σκοπός της CORBA domains είναι να σταθεροποιήσει τους ορισμούς διεπαφής, έτσι ώστε οι πελάτες να μπορούν να επιλέξουν μεταξύ διαφόρων εκτελέσεων με τις μικρότερες αλλαγές στις εφαρμογές.

Τέλος, οι Διεπαφές Εφαρμογής συμπεριλαμβάνουν όλες τις εναπομένουσες διεπαφές λογισμικού που αναπτύσσονται με τη χρήση της IDL, όπως εμπορικές και διεπαφές παλιάς τεχνολογίας.

Για την περιγραφή του OMG Μοντέλου Αναφοράς σχεδιάστηκαν δύο διαγράμματα, όπως απεικονίζονται στα σχήματα 1 και 2 αντίστοιχα, με σκοπό να διακριθούν σε δύο επίπεδα οι εμπλεκόμενες λεπτομέρειες. Το σχήμα 1 που αντιστοιχεί στο Marketing Reference Model, ορίζει το μοντέλο από τη σκοπιά του τελικού χρήστη ή της διαχείρισης. Οι όροι σε αυτό χρησιμοποιούνται συνοπτικά χωρίς τεχνικές λεπτομέρειες για να περιγράψουν κάποιες γενικές κατηγορίες προτύπων.

Στο σχήμα 2, όπου απεικονίζεται το Technical Reference Model, γίνεται η διάκριση μεταξύ αντικειμένων και διεπαφών και δείχνεται η επαναχρησιμοποίηση των προτύπων διεπαφής μεταξύ των κατηγοριών.



Σχήμα 1.2: CORBA αρχιτεκτονική διαχείρισης αντικειμένου (OMA), Technical Reference Model

Στο Technical Reference Model όλα τα αντικείμενα επικοινωνούν διαμέσου της CORBA με τους ορισμούς διεπαφής. Τρεις από τις κατηγορίες διεπαφής, η CORBA services, η CORBA facilities και η CORBA domains προτυποποιούνται. Ενώ η τέταρτη, οι Διεπαφές Εφαρμογής, ίσως να περιέχουν μη-πρότυπες διεπαφές που ορίζονται από το χρήστη και οι οποίες ίσως να είναι επεκτάσεις των προτύπων που ορίστηκαν σε άλλες κατηγορίες ή συστήματα ή διεπαφές ειδικών προϊόντων. Κάθε κατηγορία επαναχρησιμοποιεί διεπαφές και χτίζει με ένα λογικό τρόπο πάνω σε άλλες. Τα CORBA services αντικείμενα χρησιμοποιούν πρωταρχικά θεμελιώδεις διεπαφές χωρίς να βασίζονται σε άλλες κατηγορίες. Τα CORBA facilities αντικείμενα επαναχρησιμοποιούν και επεκτείνουν τις CORBA services διεπαφές, ενώ τα CORBA domains επαναχρησιμοποιούν και επεκτείνουν και τις CORBA services και τις CORBA facilities διεπαφές. Τέλος, τα Αντικείμενα Εφαρμογής έχουν διεπαφές που επαναχρησιμοποιούν όλες τις πρότυπες κατηγορίες και ίσως προσθέτουν συστηματικές επεκτάσεις. Το Technical Reference Model εξηγεί πως δομούνται τα πλαίσια των διεπαφών από τα διάφορα επίπεδα των προτύπων της CORBA.

1.3. Τι παρέχει η CORBA. CORBA IDL (Interface Definition Language)

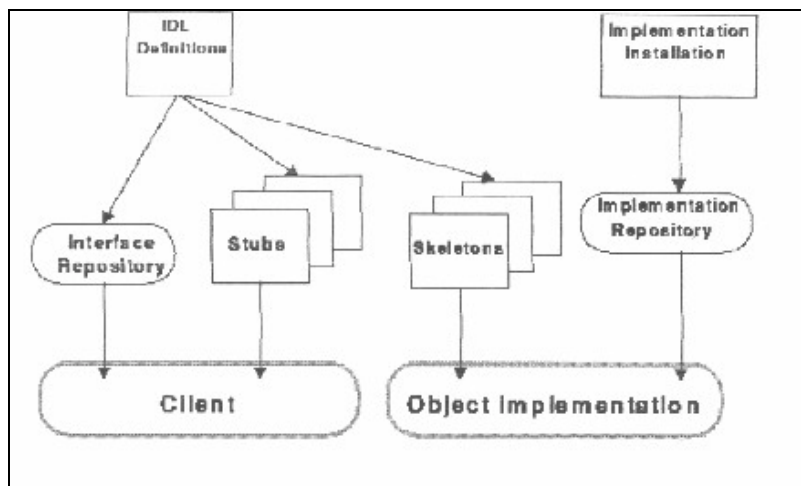
Η CORBA απλοποιεί τα καταναμημένα συστήματα με πολλούς τρόπους. Το καταναμημένο περιβάλλον ορίζεται χρησιμοποιώντας ένα αντικειμενοστραφές παράδειγμα το οποίο κρύβει όλες τις διαφορές μεταξύ των γλωσσών προγραμματισμού, των λειτουργικών συστημάτων και των τοποθεσιών όπου γίνονται οι διαδικασίες. Η αντικειμενοστραφής προσέγγιση επιτρέπει διαφορετικούς τύπους εφαρμογών να διαλειτουργούν στο ίδιο επίπεδο, κρύβοντας τις λεπτομέρειες εκτέλεσης και υποστηρίζοντας την επαναχρησιμοποίηση. Η CORBA ορίζει μια πολύ χρήσιμη γραφή για τον ορισμό διεπαφών λογισμικού που ονομάζεται Interface Definition Language (IDL). Η CORBA services ορίστηκε στην OMG IDL με διπλό ρόλο: να παρέχει υπηρεσίες προμηθευτή και εφαρμογής. Οι δημιουργοί ενθαρρύνονται να επαναχρησιμοποιούν και να επεκτείνουν τα πρότυπα διεπαφών.

Η CORBA IDL είναι η πρότυπη γραφή για τον ορισμό διεπαφών λογισμικού. Αυτή η γραφή μπορεί να χρησιμοποιηθεί με τα CORBA προϊόντα ή ανεξάρτητα. Η CORBA IDL αποτελεί πλέον μια ευρέως χρησιμοποιούμενη γραφή αναγνωρισμένη από οργανισμούς προτύπων, όπως ISO, European Computer Manufacturers Association (ECMA), OMG, X/Open κ.ά. Ως τέτοια είναι ευρέως αποδεκτή και πολύ σταθερή. Υπάρχουν πρότυπες επεκτάσεις σε γλώσσες προγραμματισμού, όπως η C, C++, Java, Smalltalk, Cobol, Ada95. Επιπρόσθετα είναι χρήσιμη για τον καθορισμό των ορίων του λογισμικού στα συστήματα τελικού χρήστη. Παρέχει τον ξεκάθαρο διαχωρισμό των διεπαφών από τις λεπτομέρειες της εκτέλεσης. Αυτή η οριοθέτηση του λογισμικού είναι βασική στην προσαρμοστικότητα των αρχιτεκτονικών λογισμικού και των εκτελέσεων.

Η σύνταξη της CORBA IDL χρησιμοποιείται για να καθορίσει τα χαρακτηριστικά και τη λειτουργία των διεπαφών των αντικειμένων. Με ένα IDL μεταφραστή για κάθε μια από τις ειδικές επεκτάσεις στις γλώσσες προγραμματισμού, οι περιγραφές διεπαφών μπορούν να μεταφραστούν σε γλώσσα ειδικών αρχείων stubs, skeletons, και αρχείων ορισμού. Προγράμματα πελάτη σε αντικείμενα ορισμένα με IDL αναπτύχθηκαν χρησιμοποιώντας τον ορισμό αντικειμένου και τα stub αρχεία, τα οποία παρέχουν επικοινωνία διαμέσου του ORB στην εκτέλεση του αντικειμένου. Ο εκτελεστής τροποποιεί τα skeleton αρχεία γεμίζοντας πλήρως τις μεθόδους με τον κώδικα εφαρμογής που χρειάζεται για να εξυπηρετήσει το αίτημα του πελάτη. Ο πελάτης ζητά το αντικείμενο χρησιμοποιώντας τη stub ρουτίνα, μια παραγόμενη τοπική κλήση λειτουργίας, η οποία διαφανώς επικοινωνεί με τον ORB. Ο τελευταίος αναλαμβάνει να θέσει μια εκτέλεση η οποία θα εξυπηρετήσει το αίτημα και παρέχει

οποιαδήποτε υποστήριξη που είναι απαραίτητη, όπως έλεγχος δεδομένων για κωδικοποίηση και αποκωδικοποίηση των παραμέτρων της λειτουργίας σε φόρμες επικοινωνίας κατάλληλες για διαβίβαση. Όταν το αντικείμενο εκτέλεσης ολοκληρώσει το αίτημα, ο ORB στέλνει τα αποτελέσματα πίσω στον πελάτη. Αν παρουσιαστεί κάποιο πρόβλημα στέλνει το κατάλληλο μήνυμα λάθους (exception) που παράγεται είτε από το αντικείμενο εκτέλεσης είτε από τον ίδιο τον ORB.

Οι διεπαφές στα χαρακτηριστικά και στις μεθόδους των αντικειμένων ορίζονται με την IDL και μεταφράζονται σε συστατικά της υποδομής που καλούνται client stubs και server skeletons. Τα client stubs χρησιμοποιούνται για να θέτουν αιτήματα και τα server skeletons χρησιμοποιούνται από τον εκτελεστή για την εκτέλεση της μεθόδου του αντικειμένου. Ένας άλλος εναλλακτικός ή επιπρόσθετος τρόπος ορισμού των διεπαφών είναι η εναπόθεση των διεπαφών σε μια υπηρεσία Αποθήκης Διεπαφών (Repository Interface). Σε κάθε εκτέλεση αντικειμένου και οι δύο τρόποι ορισμού των διεπαφών είναι ισοδύναμες για ανταπόκριση. Στο σχήμα 1.3 παρουσιάζεται πως οι διεπαφές και οι πληροφορίες εκτέλεσης γίνονται διαθέσιμες στους πελάτες και στα αντικείμενα εκτέλεσης. Οι διεπαφές ορίζονται με την IDL και/ή στις Αποθήκες Διεπαφής. Οι ορισμοί χρησιμοποιούνται για την παραγωγή των client stubs και τα skeleton αντικείμενα εκτέλεσης. Οι πληροφορίες του αντικειμένου εκτέλεσης παρέχονται κατά την εγκατάσταση και αποθηκεύονται στην Αποθήκη Εκτέλεσης για χρήση κατά την αίτηση μεταφοράς.

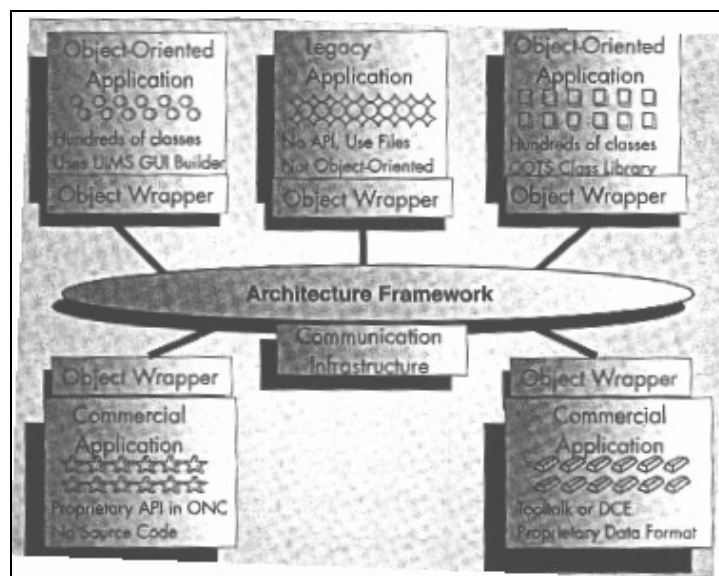


Σχήμα 1.3: Αποθήκες Διεπαφής και Εκτέλεσης

Σύμφωνα με την αποστολή του OMG, η CORBA επινοήθηκε για να παρέχει μια νέα δικτυακή υποδομή με σκοπό την επίλυση των προβλημάτων διαλειτουργικότητας του λογισμικού. Αντικαθιστά τις υπάρχουσες υποδομές Διαδικασιών Κλήσης από Απόσταση (Remote Procedure Call, RPC) και τις προγενέστερες τεχνολογίες παρέχοντας μια αντικειμενοστραφή διεπαφή δικτύου η οποία απλοποιεί τα καταναμημένα υπολογιστικά συστήματα.

1.4. Κατανομή και αρχιτεκτονική

Το πλεονέκτημα το οποίο παρέχει η CORBA στην ολοκλήρωση λογισμικού είναι η δυνατότητα ενθυλάκωσης εφαρμογών λογισμικού και η υποδομή επικοινωνιών με τον ORB. Ο οποίος επιτρέπει στον πελάτη να προβεί στις εφαρμογές μέσω της ενθυλάκωσης αντικειμένου. Το επίπεδο της ενθυλάκωσης παρέχει μια αντικειμενοστραφή εμφάνιση στις εφαρμογές του πελάτη ο οποίος χρειάζεται μόνο να νοιαστεί για τις αιτούμενες μεθόδους στη διεπαφή του αντικειμένου. Ο ορισμός αυτής της διεπαφής είναι τελείως ξεχωριστός από τις λεπτομέρειες οι οποίες περιγράφουν πως εκτελείται το αντικείμενο. Είναι ευθύνη του εκτελεστή του αντικειμένου να αναπαραστήσει από την αντικειμενοστραφή διεπαφή τις τελικές εφαρμογές.



Σχήμα 1.4: Η ανάγκη για αντικειμενοστραφή αρχιτεκτονική

Με μια τέτοια προσέγγιση κάθε τρόπος εφαρμογής μπορεί να παρέχει μια αντικειμενοστραφή εμφάνιση στις εφαρμογές του πελάτη χωρίς να απαιτείται να ξαναγραφούν οι υποκείμενες εφαρμογές. Βεβαίως, κάποια κωδικοποίηση απαιτείται

κατά την αναπαράσταση των τελικών εφαρμογών από την αντικειμενοστραφή διεπαφή κι αν αυτές είναι αντικειμενοστραφείς τότε ο κώδικας είναι τετριμμένος, αλλά και στην περίπτωση των παλιών εφαρμογών ο απαιτούμενος κώδικας είναι πολύ λιγότερος απ' ό τι θα απαιτηθεί αν ξαναγραφεί από την αρχή ως αντικειμενοστραφής. Αυτή η πολυπλοκότητα είναι τελείως διαφανής στην εφαρμογή του πελάτη, η οποία μπορεί πάντα να προσπελάσει τη λειτουργικότητα του υποκείμενου συστήματος με ένα ομοιόμορφο τρόπο, ακόμη κι όταν η τελική εφαρμογή αντικαθίσταται με ένα άλλο σύστημα που υποστηρίζει την ίδια διεπαφή. Όταν ολοκληρώνουμε παλιά λογισμικά, ο εκτελεστής λογισμικού είναι υπεύθυνος για να κάνει την αντιστοίχιση από τον server στην λειτουργικότητα της υποκείμενης εφαρμογής. Αυτό ονομάζεται *συσκευασία αντικειμένου (object wrapper)*. Η IDL δεν εμπλέκεται με το πως οι μέθοδοι εκτελούνται και δεν θέτει περιορισμούς για το πως θα μεταφερθούν τα δεδομένα προς και από το skeleton στην υποκείμενη εφαρμογή. Γι' αυτό ο εκτελεστής μπορεί να αλληλεπιδρά με την υποκείμενη εφαρμογή χρησιμοποιώντας Προγράμματα Διεπαφής Εφαρμογών (API, Application Program Interface) και φόρμες αρχείων έως ότου ο κώδικας συσκευασίας (wrapping code) μεταποιεί τα αποτελέσματα πίσω σε ένα κατάλληλο IDL τύπο ώστε να τα περάσει πίσω στην εφαρμογή του πελάτη. Ο πελάτης έχει μόνο να νοιαστεί για την αλληλεπίδραση με τη διεπαφή και δεν ξέρει τίποτα για τις μετατροπές και τις μεταφορές που γίνονται κατά την εκτέλεση του αντικειμένου.

1.4.1. Αντικειμενοστραφής αρχιτεκτονική

Η αντικειμενοστραφής αρχιτεκτονική αποτελείται από ένα σύνολο κανόνων, οδηγιών, διεπαφών και συμβάσεων που χρησιμοποιούνται για να καθορίσουν τον τρόπο με τον οποίο οι εφαρμογές θα επικοινωνούν και θα διαλειτουργούν μεταξύ τους. Ο σχεδιασμός μια αντικειμενοστραφούς αρχιτεκτονικής εστιάζεται στον καθορισμό της υποδομής και των διεπαφών μεταξύ των συστατικών λογισμικού κι όχι στη δημιουργία αυτών των συστατικών. Ο στόχος είναι να παραχθεί ένα συντηρήσιμο, κλιμακωτό και αξιόπιστο περιβάλλον, όπου οι εφαρμογές θα μπορούν να διαλειτουργούν και να ανταποκριθούν στις ανάγκες του χρήστη καθόσον το σύστημα εξελίσσεται. Για να επιτευχθεί αυτό ο αρχιτέκτονας του συστήματος θα πρέπει να ορίσει ένα ισχυρό σύνολο αποσπασμάτων που θα διαχειρίζονται την πολυπλοκότητα, τις αλλαγές και άλλες επιδράσεις. Επιπρόσθετα, θα πρέπει να νοιαστεί για το μερισμό του λογισμικού, τη ροή του ελέγχου μέσα σε ένα σύστημα λογισμικού, τη ροή πληροφοριών, το χρονισμό, τη

στρωματοποίηση των διεπαφών και τα πρότυπα πρωτοκόλλων, την hardware/software κατανομή και τη διαχείριση λαθών.

1.5. Διαχείριση Τύπων

Τύπος είναι μια αναγνωρίσιμη οντότητα με ένα σχετιζόμενο εξαγόμενο ορισμένο σε τιμές. Η τιμή ικανοποιεί τον τύπο εάν το εξαγόμενο είναι αληθινό για αυτή την τιμή. Η τιμή που ικανοποιεί ένα τύπο ονομάζεται *μέλος του τύπου*. Οι τύποι χρησιμοποιούνται στον καθορισμό των προδιαγραφών μιας λειτουργίας για να θέσουν τα όρια μιας πιθανής παραμέτρου ή να χαρακτηρίσουν ένα πιθανό αποτέλεσμα.

Σε ένα τυπικό καταναμημένο υπολογιστικό σύστημα υπάρχουν πολλαπλά προγράμματα εφαρμογής, καθένα από τα οποία έχει το δικό του μοναδικό τύπο συστήματος. Κάποιες εφαρμογές δεν έχουν τη δυνατότητα να υποστηρίξουν ένα περιβάλλον αντικειμενοστραφούς τύπου και είναι μη ρεαλιστικό και συχνά αδύνατο, ο εξαναγκασμός ενός απλού-τύπου συστήματος για όλες τις εφαρμογές. Γι αυτό κρίνονται απαραίτητες κάποιες μέθοδοι διαχείρισης αντικειμένων. Στην περίπτωση της CORBA, η IDL χρησιμοποιείται για να καθορίσει τη διαχείριση όλων των τύπων πληροφορίας σε ένα περιορισμένο περιβάλλον. Αυτοί οι τύποι δεδομένων και αντικειμένων μπορούν να χρησιμοποιηθούν ως ενδιάμεσοι, οι οποίοι απεικονίζονται για τους ειδικούς τύπους εφαρμογών στα παραγόμενα server skeletons. Τα αποτελέσματα από τις εφαρμογές απεικονίζονται προς τα πίσω στους IDL τύπους και διαβιβάζονται στις εφαρμογές πελάτη.

Αυτή η βασική μέθοδος συσκευασίας αντικειμένου (object wrapping) είναι ικανοποιητική για μια χαμηλού επιπέδου ολοκλήρωση σε ένα απλό σύστημα. Αλλά σε ένα καταναμημένο σύστημα, όπου ένας τύπος πιθανόν να μετακινηθεί μεταξύ δύο διαφορετικών εφαρμογών, χρειάζονται πιο ευέλικτοι μηχανισμοί για να λύσουν τα μεγάλα θέματα της διαλειτουργικότητας. Η CORBA για να λύσει το πρόβλημα της λήψης ενός άγνωστου τύπου παρέχει υπηρεσίες διαχείρισης απαιτώντας όλοι οι τύποι να έχουν ένα κωδικοποιημένο τύπο (TypeCode). Ο TypeCode είναι μια διαφανής αναφορά που περιγράφει τους τύπους CORBA. Περιέχει μια περιγραφή των περιεχομένων καθώς και την οργάνωση ενός τύπου ορισμένου από το χρήστη και μπορεί να προσπελαστεί χρησιμοποιώντας τις πρότυπες CORBA διεπαφές. Με ένα συνοδευόμενο TypeCode, ένας τύπος ορισμένος από το χρήστη μπορεί να διαχειριστεί μόνο εάν η εφαρμογή πελάτη γνωρίζει τον ορισμό του τύπου και πως να τον διαχειριστεί. Εντούτοις, εάν δεδομένα ορισμένα από το χρήστη μεταφέρονται με ένα

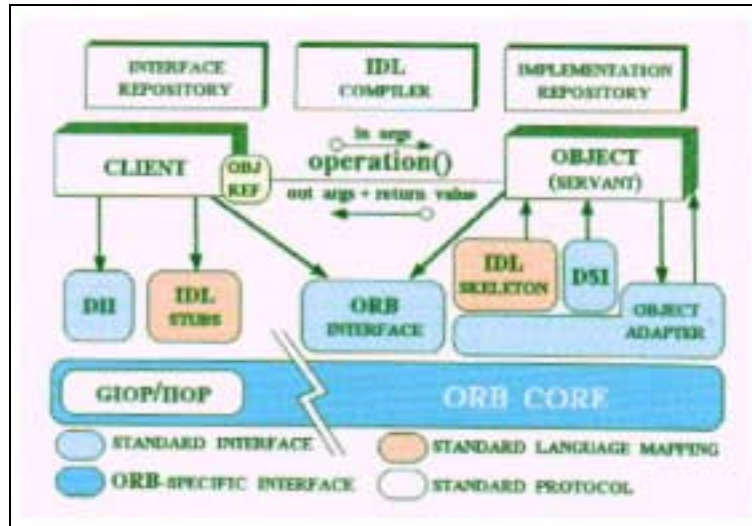
TypeCode, ο πελάτης ίσως χρησιμοποιήσει τον TypeCode για να φανερώσει πληροφορίες του τύπου.

1.6. Αναπαράσταση της OMG IDL στις γλώσσες προγραμματισμού

Διαφορετικές αντικειμενοστραφείς ή μη γλώσσες προγραμματισμού μερικές φορές προτιμούν να προσπελάσουν τα CORBA αντικείμενα με διαφορετικούς τρόπους. Όσον αφορά τις αντικειμενοστραφείς γλώσσες είναι επιθυμητό να βλέπουν τα CORBA αντικείμενα ως αντικείμενα γλωσσών προγραμματισμού. Ακόμη και για τις μη αντικειμενοστραφείς είναι καλό να κρυφτεί η ακριβής ORB παρουσίαση της αναφοράς αντικειμένου, τα ονόματα των μεθόδων κλπ. Μια συγκεκριμένη αναπαράσταση της OMG IDL σε μια γλώσσα προγραμματισμού πρέπει να είναι ίδια για όλες τις ORB εκτελέσεις. Η αναπαράσταση στις γλώσσες περιλαμβάνει ορισμούς των ειδικών στη γλώσσα τύπων δεδομένων και διαδικασίες διεπαφών για την προσπέλαση αντικειμένων διαμέσου του ORB. Αυτή περιλαμβάνει τη δομή της διεπαφής του client stub (δεν απαιτείται για αντικειμενοστραφείς γλώσσες), τη δυναμική διεπαφή αίτησης, την skeleton εκτέλεση, τους προσαρμοστές αντικειμένου και την άμεση ORB διεπαφή. Η αναπαράσταση στις γλώσσες ορίζει την αλληλεπίδραση μεταξύ των αιτημάτων αντικειμένων και τους κινδύνους του ελέγχου στον πελάτη ή στην εκτέλεση. Η πιο κοινές αναπαραστάσεις παρέχουν σύγχρονες κλήσεις, στις επιστροφές των ρουτινών όταν ολοκληρώνεται η λειτουργία του αντικειμένου. Επιπρόσθετες αναπαραστάσεις μπορεί να παρέχονται ώστε να επιτρέψουν σε μια κλήση να ξεκινήσει και ο έλεγχος να επιστραφεί στο πρόγραμμα. Σε τέτοιες περιπτώσεις, επιπρόσθετες ρουτίνες, ειδικές κάθε γλώσσας, μπορεί να παρέχονται για το συγχρονισμό των κινδύνων του ελέγχου των προγραμμάτων με την αίτηση του αντικειμένου.

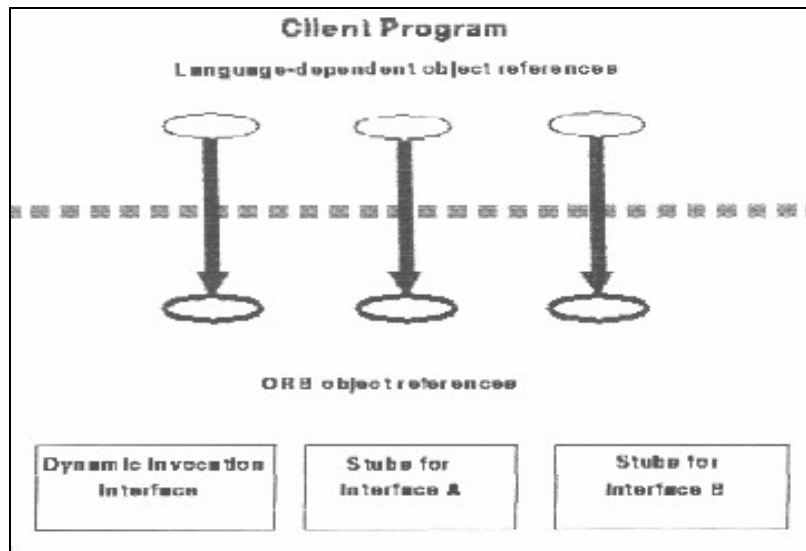
1.7. Μεσίτης Αιτούμενου Αντικειμένου (Object Request Broker, ORB)

Ο ORB υποστηρίζει την εκτέλεση μιας ευέλικτης αρχιτεκτονικής επιτρέποντας τη δυναμική εναπόθεση αίτησης ενός συστατικού του πελάτη στο αντικείμενο που θα πραγματοποιήσει την εκτέλεση.

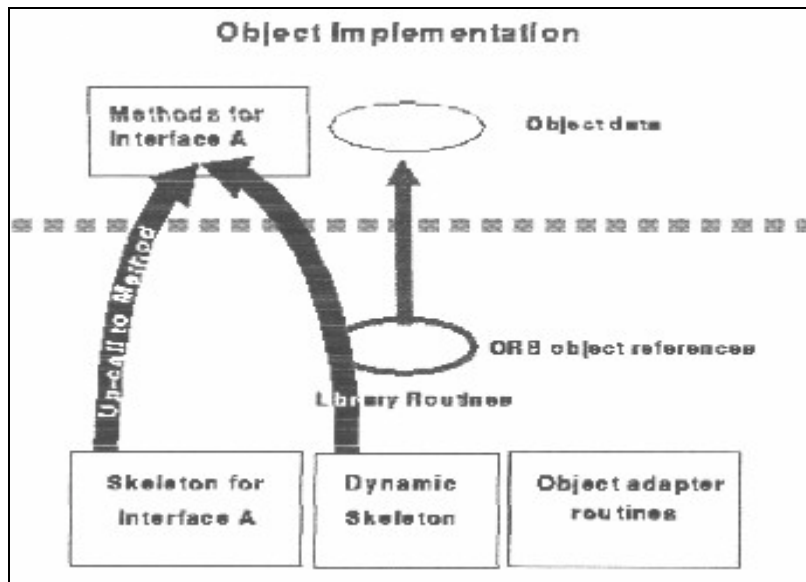


Σχήμα 1.5: Η δομή των διεπαφών του Μεσίτη Αιτούμενου Αντικειμένου

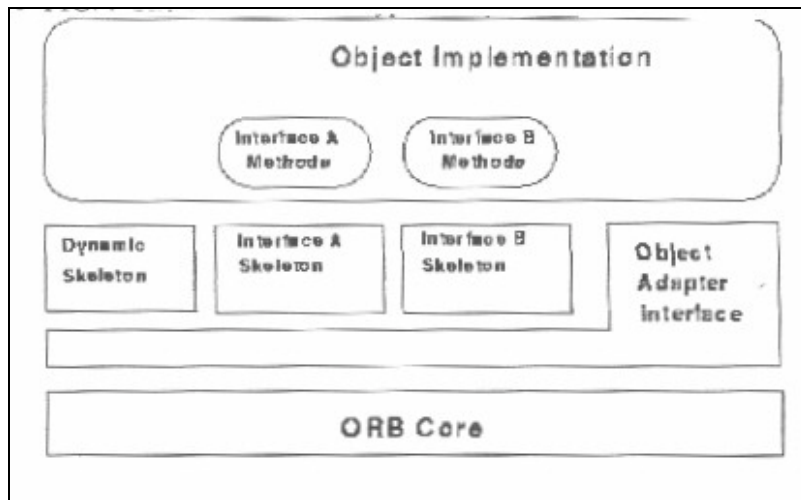
Ο ORB δέχεται μια αίτηση από τον πελάτη διαμέσου του ειδικού IDL stub που εξαρτάται από τη διεπαφή του αντικειμένου-στόχου ή μιας δυναμικά παραγόμενης αίτησης χρησιμοποιώντας τη διεπαφή-πελάτη DII (Dynamic Invocation Interface) που είναι η ίδια διεπαφή ανεξάρτητη από τη διεπαφή του αντικειμένου-στόχου και τακτοποιεί τα δεδομένα σε μια πλατφόρμα με μορφή που είναι ανεξάρτητη γλώσσας. Για κάποιες λειτουργίες ο πελάτης έχει τη δυνατότητα να αλληλεπιδρά άμεσα με τον ORB. Στη συνέχεια, θέτει μια υπηρεσία που υποστηρίζει την αιτούμενη διεπαφή ή δημιουργεί μια αν δεν είναι κάποια διαθέσιμη. Έπειτα, η αίτηση μεταβιβάζεται στο αντικείμενο εκτέλεσης που τη δέχεται ως μια "προς τα πάνω κλήση" (up-call) είτε από το IDL skeleton είτε διαμέσου ενός δυναμικού skeleton, με τον ORB να παρέχει κάθε απαιτούμενες μετατροπές προς το server skeleton (DSI, Dynamic Skeleton Interface). Το αντικείμενο εκτέλεσης, καθώς επεξεργάζεται μια αίτηση ή σε άλλες χρονικές στιγμές, ίσως να καλέσει μέσω του κατάλληλου κατά περίπτωση προσαρμοστή αντικειμένου (Object Adapter) κάποιες υπηρεσίες από τον ORB. Αυτή η διαδικασία ακολουθείται με την αντίστροφη πορεία μεταβιβάζοντας το αποτέλεσμα της εκτέλεσης και τον έλεγχο πίσω στον πελάτη. Στα σχήματα 1.6, 1.7 και 1.8 παριστάνονται η δομή ενός τυπικού πελάτη, μιας τυπικής εκτέλεσης αντικειμένου και ενός τυπικού προσαρμοστή αντικειμένου αντίστοιχα.



Σχήμα 1.6. Δομή ενός τυπικού πελάτη



Σχήμα 1.7. Δομή μιας τυπικής εκτέλεσης αντικειμένου



Σχήμα 1.8. Δομή ενός τυπικού προσαρμοστή αντικειμένου

Στην αρχιτεκτονική, ο ORB δεν απαιτείται να εκτελεστεί ως ένα απλό συστατικό, αλλά περισσότερο ορίζεται από τις διεπαφές του. Κάθε ORB εκτέλεση που παρέχει την κατάλληλη διεπαφή είναι αποδεκτή. Η διεπαφή οργανώνεται σε τρεις κατηγορίες:

Λειτουργίες που είναι ίδιες για όλες τις ORB εκτελέσεις.

Λειτουργίες που είναι ειδικές για συγκεκριμένους τύπους αντικειμένων.

Λειτουργίες που είναι ειδικές για συγκεκριμένες μορφές των αντικειμένων εκτέλεσης.

Διαφορετικά ORBs ίσως να κάνουν εντελώς διαφορετικές επιλογές εκτέλεσης και μαζί με τους IDL μεταφραστές, αποθήκες και τους διάφορους προσαρμοστές αντικειμένων παρέχουν ένα σύνολο υπηρεσιών στους πελάτες και στις εκτελέσεις αντικειμένων που έχουν διαφορετικές ιδιότητες και ποιότητες. Ίσως να υπάρχουν πολλαπλές ORB εκτελέσεις, οι οποίες έχουν διαφορετικές παρουσιάσεις για αναφορές αντικειμένων και διαφορετικές σημασίες των εκτελούμενων αιτημάτων. Ίσως να είναι πιθανό για ένα πελάτη ταυτόχρονα να έχει πρόσβαση σε δύο αναφορές αντικειμένων διαχειριζόμενες από διαφορετικές ORB εκτελέσεις. Όταν δύο ORBs πρόκειται να εργαστούν μαζί, τότε πρέπει να είναι σε θέση να ξεχωρίσουν τις αναφορές αντικειμένων τους και αυτό δεν είναι ευθύνη το πελάτη. Το ORB Core είναι το τμήμα του ORB που παρέχει τη βασική παρουσίαση των αντικειμένων και την επικοινωνία των αιτημάτων. Η CORBA σχεδιάστηκε για να υποστηρίξει διαφορετικούς μηχανισμούς αντικειμένων και το επιτυγχάνει κατασκευάζοντας τον ORB με συστατικά πάνω από τον ORB Core, τα οποία παρέχουν διεπαφές που μπορούν να καλύψουν τις διαφορές μεταξύ των ORB Cores.

Ο ORB υποστηρίζει τη διαπεραστικότητα στην πλευρά του πελάτη, η οποία επιτρέπει στον δημιουργό να κάνει αλλαγές στην εκτέλεση της εφαρμογής χωρίς να επηρεάζει τους πελάτες που χρησιμοποιούν τις υπηρεσίες εκτέλεσης. Ο πελάτης δεν γνωρίζει εάν η υπηρεσία εκτελείται, πότε έγινε η αίτηση ή εάν ενεργοποιείται και έγινε διαθέσιμη και φυσικά ούτε πως εκτελείται (ποιος προσαρμοστής αντικειμένου χρησιμοποιείται, ποιο ORB).

1.7.1. Παραδείγματα ORBs

Υπάρχουν διάφορες ORB εκτελέσεις στην κοινή ORB αρχιτεκτονική. Θα πρέπει να σημειωθεί ότι ένας συγκεκριμένος ORB μπορεί να υποστηρίζει πολλαπλές επιλογές και πρωτόκολλα για επικοινωνία.

ORB στους πελάτες και στις εκτελέσεις:

Εάν υπάρχει ένας κατάλληλος επικοινωνιακός μηχανισμός ο ORB μπορεί να εκτελεστεί σε ρουτίνες που βρίσκονται στους πελάτες και στις εκτελέσεις. Τα stubs στον πελάτη είτε χρησιμοποιούν ένα τοπικό διαφανή IPC μηχανισμό ή άμεσα προσπελούν μια τοπική υπηρεσία ώστε να δημιουργήσουν επικοινωνία με τις εκτελέσεις. Ο κώδικας που συνδέεται με τις εκτελέσεις είναι υπεύθυνος για την εύρεση κατάλληλων Βάσεων Δεδομένων για τη χρήση τους από τον πελάτη.

ORB βασισμένη στον εξυπηρετητή:

Για την κεντροποίηση της διαχείρισης του ORB όλοι οι πελάτες και οι εκτελέσεις μπορούν να επικοινωνούν με ένα ή περισσότερους εξυπηρετητές των οποίων η δουλειά είναι να δρομολογούν αιτήματα από τους πελάτες στις εκτελέσεις. Ο ORB θα μπορούσε να είναι ένα κανονικό πρόγραμμα και μια κανονική IPC θα μπορούσε να χρησιμοποιηθεί για την επικοινωνία με τον ORB.

ORB βασισμένη στο σύστημα:

Για την ενίσχυση της ασφάλειας, της ακμαιότητας και της εκτέλεσης ο ORB θα μπορούσε να παρέχεται ως μια βασική υπηρεσία του υποκείμενου λειτουργικού συστήματος. Εκμεταλλευόμενοι το πλεονέκτημα ότι το λειτουργικό σύστημα γνωρίζει τις θέσεις και τη δομή των πελατών και των εκτελέσεων θα μπορούσαν να αποφευχθούν διάφορες λειτουργίες όταν και οι δύο ανήκουν στην ίδια μηχανή.

ORB βασισμένη σε βιβλιοθήκη:

Για αντικείμενα μικρού βάρους και των οποίων οι εκτελέσεις μπορούν να είναι σε κοινή χρήση, η εκτέλεση θα μπορούσε να είναι σε βιβλιοθήκη. Σ' αυτήν την περίπτωση τα stubs θα μπορούσαν να είναι οι πραγματικές μέθοδοι. Αυτό σημαίνει ότι είναι

δυνατόν το πρόγραμμα του πελάτη να προσπελάσει στα δεδομένα για τα αντικείμενα και η εκτέλεση να εμπιστεύεται τον πελάτη ότι δεν θα καταστρέψει τα δεδομένα.

1.8. Διαλειτουργικότητα

Η ORB διαλειτουργικότητα καθορίζει μια περιεκτική, ευέλικτη προσέγγιση για την υποστήριξη δικτύων αντικειμένων που είναι καταναμημένα και διαχειρίζονται από πολλαπλούς ετερογενείς CORBA-ORBs. Η προσέγγιση αυτή είναι παγκόσμια γιατί τα στοιχεία της μπορούν να συνδυαστούν με πολλούς τρόπους ώστε να ικανοποιήσουν ένα ευρύ φάσμα αναγκών.

Τα στοιχεία της διαλειτουργικότητας είναι:

- Αρχιτεκτονική ORB διαλειτουργικότητας
- Υποστήριξη Inter-ORB γέφυρας
- General και Internet Inter-ORB πρωτόκολλα (GIOPs και IIOPs)

Επιπρόσθετα, η αρχιτεκτονική εξυπηρετεί Environment-Specific Inter-ORB πρωτόκολλα (ESIOPs) για συγκεκριμένα περιβάλλοντα όπως το DCE.

1.8.1. Αρχιτεκτονική ORB διαλειτουργικότητας

Η αρχιτεκτονική ORB διαλειτουργικότητας παρέχει ένα εννοιολογικό πλαίσιο για τον καθορισμό των στοιχείων της διαλειτουργικότητας και για να προσδιορίσει τα σημεία συμμόρφωσης της. Επίσης, χαρακτηρίζει νέους μηχανισμούς και καθορίζει συνθήκες απαραίτητες ώστε να επιτευχθεί η διαλειτουργικότητα μεταξύ ανεξάρτητα παραγόμενων ORBs. Ειδικότερα, η αρχιτεκτονική εισάγει τις έννοιες της immediate και mediated bridging της ORB domains. Το Internet Inter-ORB πρωτόκολλο (IIOP) σχηματίζει την κοινή βάση για broad-scope mediated bridging. Η υποστήριξη Inter-ORB bridge μπορεί να χρησιμοποιηθεί για την εκτέλεση και των immediate bridges και για να κατασκευάσει "half-bridges" στα mediated bridge domains.

Με τη χρήση των τεχνικών γεφύρωσης, οι ORBs μπορούν να διαλειτουργούν χωρίς να γνωρίζουν οποιαδήποτε λεπτομέρεια εκτέλεσης των ORBs, όπως κάποια συγκεκριμένα IPC ή πρωτόκολλα (όπως τα ESIOPs) χρησιμοποιούνται για την εκτέλεση των CORBA εξειδικεύσεων.

Το IIOP μπορεί να χρησιμοποιηθεί για να γεφυρώσει δύο ή περισσότερες ORBs εκτελώντας "half bridges" οι οποίες επικοινωνούν χρησιμοποιώντας το IIOP. Αυτή η προσέγγιση δουλεύει και για απομονωμένες ORBs και για δικτυωμένες που χρησιμοποιούν ESIOP.

Το ΠΙΟΡ μπορεί ακόμη να χρησιμοποιηθεί για την εκτέλεση ενός εσωτερικού μηνύματος ενός ORB. Εφόσον οι ORBs δεν απαιτούνται να χρησιμοποιούν εσωτερικά το ΠΙΟΡ, το κέρδος της μη απαιτούμενης εκ των προτέρων γνώσης της μεταξύ τους εκτέλεσης είναι πλήρως ικανοποιητικό.

1.8.2. Υποστήριξη Inter-ORB Bridge

Η αρχιτεκτονική διαλειτουργικότητας καθορίζει επακριβώς το ρόλο των διαφόρων ειδών domains για ειδικές ORB πληροφορίες. Τα domains μπορούν να περιλαμβάνουν domains αναφοράς αντικειμένου, τύπου, ασφάλειας, το domain συνδιαλλαγής κ.ά.

Όταν δύο ORBs είναι στο ίδιο domain μπορούν να επικοινωνούν άμεσα. Σε πολλές περιπτώσεις αυτή είναι η επιθυμητή προσέγγιση. Αυτό δεν συμβαίνει πάντα, αφού οι οργανισμοί αναγκάζονται συχνά να καθιερώσουν domains τοπικού ελέγχου. Όταν η πληροφορία σε μια αίτηση πρέπει να αφήσει το domain της, η αίτηση πρέπει να διατρέξει μία γέφυρα. Ο ρόλος της γέφυρας είναι να σιγουρέψει ότι το περιεχόμενο και τα εννοιολογικά θα αναπαρασταθούν κατάλληλα από τον έναν ORB στον άλλο, ενώ ο χρήστης ενός ORB βλέπει μόνο το κατάλληλο περιεχόμενο και τα εννοιολογικά.

Το στοιχείο της Υποστήριξης Inter-ORB bridge καθορίζει ORB APIs και συνθήκες για να καταστήσει δυνατή την εύκολη κατασκευή των γεφυρών διαλειτουργικότητας μεταξύ ORB domains.

Η Υποστήριξη Inter-ORB bridge μπορεί ακόμη να χρησιμοποιηθεί για την παροχή διαλειτουργικότητας με μη CORBA συστήματα, όπως το Component Object Model (COM) της Microsoft. Η ευκολία για να πραγματοποιηθεί αυτό εξαρτάται από το πόσο αυτά τα συστήματα προσαρμόζουν στο CORBA μοντέλο αντικειμένου.

1.8.3. General Inter-ORB Protocol (GIOP)

Το στοιχείο του Γενικού Inter-ORB πρωτοκόλλου (GIOP) καθορίζει μια τυποποιημένη σύνταξη μεταφοράς και ένα σύνολο μορφών μηνυμάτων για επικοινωνίες μεταξύ των ORBs. Το GIOP χτίστηκε ειδικά για ORB με ORB αλληλεπιδράσεις και σχεδιάστηκε ώστε να δουλέψει άμεσα πάνω από κάθε προσανατολισμένο-προς-σύνδεση πρωτόκολλο μεταφοράς το οποίο έχει ένα ελάχιστο σύνολο υποθέσεων. Αυτό δεν απαιτεί ή δεν βασίζεται στη χρήση υψηλότερου επιπέδου RPC (Remote Procedure Call) μηχανισμών. Το πρωτόκολλο είναι όσο γίνεται απλό, κλιμακωτό και σχετικά εύκολο στην εκτέλεση. Σχεδιάστηκε για να επιτρέπει φορητές εκτελέσεις με ελάχιστες εξαρτήσεις στο υποστηριζόμενο λογισμικό, διαφορετικό από το υποκείμενο επίπεδο μεταφοράς. Ενώ οι εκδόσεις του GIOP τρέχουν σε διαφορετικές

μεταφορές δεν ήταν άμεσα ικανές για διαλειτουργικότητα, η κοινόχρηστη ιδιότητά τους επέτρεψε την εύκολη και αποτελεσματική γεφύρωση μεταξύ τέτοιων δικτυακών domains.

1.8.4. Internet Inter-ORB Protocol (IIOP)

Το στοιχείο του Internet Inter-ORB πρωτοκόλλου (IIOP) καθορίζει πως τα GIOP μηνύματα ανταλλάσσονται χρησιμοποιώντας TCP/IP συνδέσεις. Το IIOP καθορίζει ένα τυποποιημένο πρωτόκολλο διαλειτουργικότητας για το Internet, παρέχοντας "έξω από το κουτί" διαλειτουργία με άλλους συμβατούς ORBs βασισμένο στο περισσότερο δημοφιλές προϊόν -και προμηθευτής- ουδέτερο επίπεδο μεταφοράς. Μπορεί ακόμη να χρησιμοποιηθεί ως το πρωτόκολλο μεταξύ half-bridges.

Το πρωτόκολλο σχεδιάστηκε ώστε να είναι κατάλληλο για χρήση από κάθε ORB για να διαλειτουργεί σε Internet πρωτόκολλο domains εκτός κι αν ένα εναλλακτικό πρωτόκολλο απαιτείται από το ειδικό κέντρο σχεδιασμού ή από το λειτουργικό περιβάλλον του ORB. Μ' αυτήν την έννοια αντιπροσωπεύει το βασικό Inter-ORB πρωτόκολλο για TCP/IP περιβάλλοντα, ένα περισσότερο διαισθανόμενο επίπεδο μεταφοράς. Η σχέση του IIOP με το GIOP είναι όμοια με αυτήν μιας γλώσσας προγραμματισμού με την OMG IDL. Το GIOP μπορεί να αναπαριστάται σε ένα αριθμό διαφορετικών μεταφορών και καθορίζει τα στοιχεία του πρωτοκόλλου που είναι κοινά σε όλες αυτές τις αναπαραστάσεις. Το GIOP, εντούτοις, από μόνο του δεν παρέχει πλήρη διαλειτουργικότητα, όπως η IDL δεν μπορεί να χρησιμοποιηθεί για την δημιουργία ολοκληρωμένων προγραμμάτων. Το IIOP και άλλες παρόμοιες αναπαραστάσεις σε διαφορετικές μεταφορές, είναι συγκεκριμένες πραγματικότητες των συνοπτικών ορισμών του GIOP.

1.8.5. Environment-Specific Inter-ORB Protocols (ESIOPs)

Αυτή η εξειδίκευση έγινε για ένα σύνολο ανοικτού τέλους ESIOPs πρωτοκόλλων. Τέτοια πρωτόκολλα μπορούν να χρησιμοποιηθούν για "έξω από το κουτί" διαλειτουργικότητα στην πλευρά του χρήστη, όπου μια συγκεκριμένη δικτύωση ή μια υποδομή κατανεμημένου υπολογιστικού συστήματος είναι ήδη σε γενική χρήση. Εξαιτίας των δυνατοτήτων δημιουργίας ευκολιών από το ειδικό περιβάλλον, τα ESIOPs μπορούν να υποστηρίξουν ειδικές δυνατότητες, όπως αυτές που σχετίζονται με την ασφάλεια και τη διοίκηση. Ενώ τα ESIOPs είναι κατάλληλα για συγκεκριμένα περιβάλλοντα, όλες οι ESIOPs εξειδικεύσεις θα αναμένεται να συμμορφωθούν στις συμφωνίες της αρχιτεκτονικής της γενικής ORB διαλειτουργικότητας ώστε να επιτυγχάνεται με ευκολία η γεφύρωση. Η Υποστήριξη Inter-ORB bridge δίνει τη

δυνατότητα να κατασκευαστούν γέφυρες μεταξύ των ORB domains που χρησιμοποιούν το ΙΟΡ και τα ORB domains που κι αυτά χρησιμοποιούν ένα συγκεκριμένο ΕΣΙΟΡ.

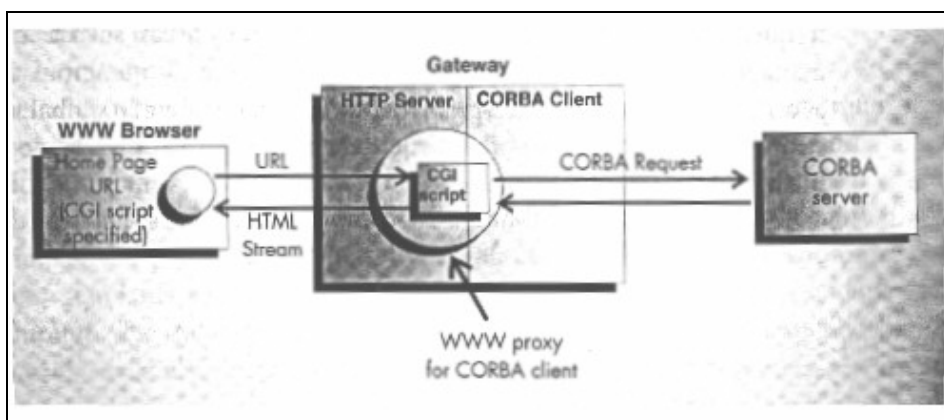
1.9. Το παράδειγμα της CORBA-CGI Gateway

Ως ένα παράδειγμα θα αναφερθεί η λύση λογισμικού με το όνομα CORBA-CGI Gateway με σκοπό τη χρήση της CORBA ως πηγής δεδομένων για τη δημιουργία HTML σελίδων (Mowbray,1997).

Τυπικά κατά την αναζήτηση πληροφοριών στο διαδίκτυο, ο χρήστης εισάγει ή επιλέγει μια σύνδεση σ' ένα browser ή εισάγει δεδομένα σε μια HTML φόρμα και ενεργοποιεί μια διεύθυνση HTTP URL. Ο browser τεμαχίζει την πληροφορία σε σε πακέτα αντιστοιχώντας τους ένα αύξοντα αριθμό και τα στέλνει στον HTTP server που υποδεικνύεται από την URL διεύθυνση. Η διεύθυνση URL αντιστοιχεί σ' ένα HTML έγγραφο στον server, ένα αρχείο ή ένα script που πρέπει να εκτελεστούν. Το HTTP πακετάρει τα δεδομένα στο HTML αρχείο ή στην έξοδο του προγράμματος στην περίπτωση ενός script και μεταφέρει τα δεδομένα πίσω στον browser. Ο μηχανισμός είναι απλός και ισχυρός, εντούτοις στο μεγαλύτερο τμήμα, η διεπαφή χρησιμοποιείται για την έκθεση στατικής πληροφορίας αποθηκευμένη σε συνταγμένα από πριν HTML αρχεία. Κάποια scripts για τη δημιουργία δυναμικών σελίδων είναι διαθέσιμα, αλλά αυτές γενικώς είναι hard-coded scripts που είτε είναι τοποθετημένες σε HTTP μηχανές ή παρέχουν αναφορές σε συγκεκριμένες πηγές δεδομένων, περιορίζοντας την ευελιξία του συστήματος. Με την συνεργασία των CORBA πελατών ως ο ικανός μηχανισμός για τη δημιουργία δυναμικών σελίδων, οι σελίδες μπορούν να έχουν μια ειδική εκτέλεση και μπορούν να προσπελάσουν υπηρεσίες διαμέσου μιας περισσότερο ευέλικτης, μετακινούμενης και συντηρήσιμης διεπαφής.

Η Common Gateway Interface (CGI) είναι μέρος των HTTP πρωτοκόλλων και επιτρέπει το τρέξιμο εξωτερικών προγραμμάτων και scripts σ' ένα server. Πρώτα δημιουργεί μια αναπαράσταση από μια διεύθυνση URL που είναι μια εκ των προτέρων προγραμματιζόμενη αίτηση πληροφορίας, υπάρχουσα σε μια στατική HTML σελίδα προς σ' ένα CGI script ώστε να εκτελεστεί στον HTTP server. Αυτό το CGI script μπορεί να είναι ένας CORBA πελάτης με τον αντίστοιχο ORB, στην ίδια μηχανή με τον HTTP server. Ορίζοντας έναν CORBA πελάτη να αναφέρεται διαμέσου CGI διεπαφών, ο CGI script/CORBA πελάτης μπορεί να επεξεργαστεί την αίτηση με την προσπέλαση των υπηρεσιών CORBA services. Αυτός ο πελάτης κερδίζει από την CORBA διαπεραστικότητα τοποθεσιών, τη δυναμική επισύναψη σε εκτελέσεις υπηρεσιών,

αυτόματη ενεργοποίηση του server κ.ά. Ενώ η εφαρμογή του πελάτη πρέπει να είναι τοποθετημένη στην ίδια μηχανή με το HTTP, οι CORBA services μπορούν να είναι καταναμημένες και αυτό είναι το σημαντικό. Μετά την επεξεργασία της CORBA αίτησης, ο CORBA πελάτης συναρμολογεί τα δεδομένα στον HTTP server. Επειδή η έξοδος ενός CGI script πρέπει να είναι μια HTML σελίδα, ο CORBA πελάτης είναι υπεύθυνος να δημιουργήσει μια σελίδα δυναμικά ή παρέχοντας την τοποθεσία μιας σελίδας ως έξοδο στον HTTP server. Δημιουργεί δυναμικά ένα HTML έγγραφο ή δυναμικά ένα Java applet ώστε να μεταφερθεί πίσω στον πελάτη. Τέλος, ο HTTP server στέλνει το HTML έγγραφο στον πελάτη ως ανταπόκριση στην αρχική αίτηση.



Σχήμα 1.9: Η HTTP CGI διεπαφή μπορεί να χρησιμοποιηθεί για τη μεταβίβαση αιτήματος στους CORBA πελάτες, οι οποίοι ανταποκρίνονται χρησιμοποιώντας τις CORBA services για την δυναμική κατασκευή μιας HTML σελίδας.

Αυτό το σχέδιο μπορεί να χρησιμοποιηθεί είτε για να παρέχει μια πύλη με το Internet και κάποιες άλλες τεχνολογίες είτε ως ένα οδηγός για την περιτύλιξη της λειτουργικότητας του CORBA πελάτη με μια υπάρχουσα τεχνολογία.

Τι κερδίζουμε

Χρησιμοποιώντας μ' αυτόν τον τρόπο την CORBA, αυτή συνεργάζεται με το μοντέλο αντικειμένου του διαδικτύου.

Ο browser χρήστης δεν εκτίθεται στην CORBA ή στις αναφορές αντικειμένου.

Εργάζεται μαζί με το μοντέλο αντικειμένου του διαδικτύου, παρά εναντίον. Κρατά σαφείς διαχωριστικές γραμμές μεταξύ των περιβαλλόντων του διαδικτύου και της CORBA με τα όρια ενθυλακωμένα μέσα στους CGI/CORBA πελάτες.

Άλλες συνέπειες

Δεν υπάρχει πραγματική άμεση προσπέλαση σε CORBA αντικείμενα.

Ο CORBA πελάτης πρέπει να τρέξει στην ίδια μηχανή με τον HTTP server (μολονότι ένα script μπορεί να προσπελάσει άλλες μηχανές διαμέσου άλλου μηχανισμού, όπως rsh, ftp κ.ά.).

Είτε ο CORBA πελάτης ή ένας εξειδικευμένος HTTP server χρειάζεται για τη δυναμική δημιουργία μιας HTML σελίδας χρησιμοποιώντας τα CORBA δεδομένα.

2. DCOM (Distributed Component Object Model)

2.1. Εισαγωγή στο DCOM

Το DistributedCOM είναι η επέκταση του COM (Component Object Model) της εταιρείας Microsoft για την υποστήριξη της επικοινωνίας μεταξύ αντικειμένων διαφορετικών υπολογιστών σ' ένα LAN, WAN ή ακόμη και στο Internet. Έτσι, με το DCOM αναβαθμίζονται οι επενδύσεις σε εφαρμογές που βασίζονται στην τεχνολογία COM, με την είσοδό τους στον κόσμο των βασισμένων σε πρότυπα κατανεμημένων υπολογιστικών συστημάτων. Τα COM και DCOM δεν είναι πλέον ιδιοκτησία της Microsoft, αλλά διευθύνεται από το ανεξάρτητο ActiveX® Consortium.

Το DCOM μπορεί να βοηθήσει στην επίλυση διαφόρων προβλημάτων τόσο σε υπάρχουσες εφαρμογές όσο σε εντελώς νέους σχεδιασμούς, όπως η ανεκτικότητα λάθους σε περίπτωση αποτυχίας του υλικού, η εγκατάσταση σ' ένα περιβάλλον μικρής ομάδας εργασίας αλλά και εξίσου καλά σε κλίμακα ολόκληρης επιχείρησης, η ακμαιότητα σε περιπτώσεις δικτυακών αποτυχιών, η εξυπηρέτηση ενός ευρέως φάσματος μηχανών πελατών με διαφορετικές δυνατότητες ή σε διαφορετικές γεωγραφικές περιοχές, η αποτελεσματικότητα με την έννοια φόρτισης του δικτύου.

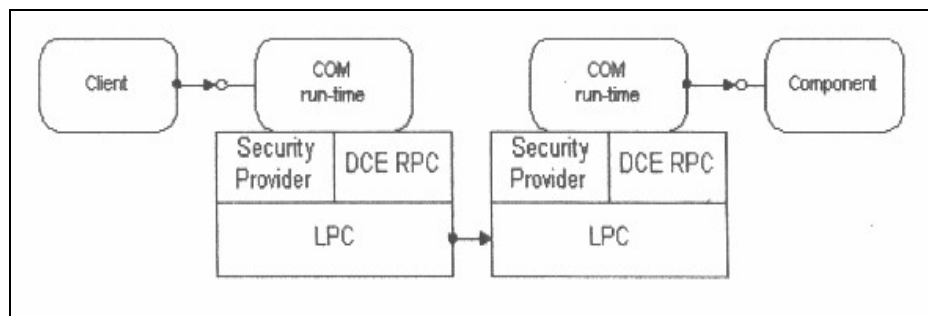
2.2. Κατανεμημένες Εφαρμογές

Οι κατανεμημένες εφαρμογές εισάγουν ένα τελείως διαφορετικό είδος σχεδίασης και εκτέλεσης των διαφόρων θεμάτων. Η επιπρόσθετη πολυπλοκότητα που εισάγεται εξαιτίας της κατανομής καλύπτεται από το σημαντικό κέρδος που απολαμβάνει η εκτέλεση της εφαρμογής με την κατανεμημένη αρχιτεκτονική. Κάποιες εφαρμογές είναι κληρονομικά κατανεμημένες, όπως τα παιχνίδια με πολλαπλούς χρήστες, οι εφαρμογές συνομιλίας και τηλεδιάσκεψης, ενώ κάποιες άλλες είναι επίσης κατανεμημένες με την έννοια ότι έχουν τουλάχιστον δύο συστατικά (components) που τρέχουν σε διαφορετικές μηχανές. Αλλά επειδή οι τελευταίες δεν σχεδιάστηκαν ώστε να είναι κατανεμημένες, γι' αυτό είναι περιορισμένες ως προς την κλιμάκωση και την ευκολία εκτέλεσης. Κάθε είδος ομαδικής εφαρμογής, οι περισσότερες εφαρμογές πελάτη/εξυπρέτη και ακόμη κάποιες εφαρμογές παραγωγικότητας επιφάνειας εργασίας στοιχειωδώς ελέγχουν τον τρόπο με τον οποίο επικοινωνούν και συνεργάζονται οι χρήστες τους. Θεωρώντας αυτές τις εφαρμογές ως κατανεμημένες και τρέχοντας τα σωστά συστατικά στις σωστές θέσεις κερδίζει ο χρήστης και

βελτιστοποιεί τη χρήση του δικτύου και των πόρων του υπολογιστικού συστήματος. Η εφαρμογή που σχεδιάστηκε έχοντας υπόψη την κατανομή μπορεί να εξυπηρετήσει διαφορετικούς πελάτες με διαφορετικές δυνατότητες, τρέχοντας όταν είναι δυνατόν συστατικά στην πλευρά του πελάτη και στην πλευρά του εξυπηρετή όταν απαιτείται. Με το σχεδιασμό κατανεμημένων εφαρμογών η διεύθυνση του συστήματος είναι περισσότερο ευέλικτη στην εκτέλεση.

2.3. Η DCOM αρχιτεκτονική

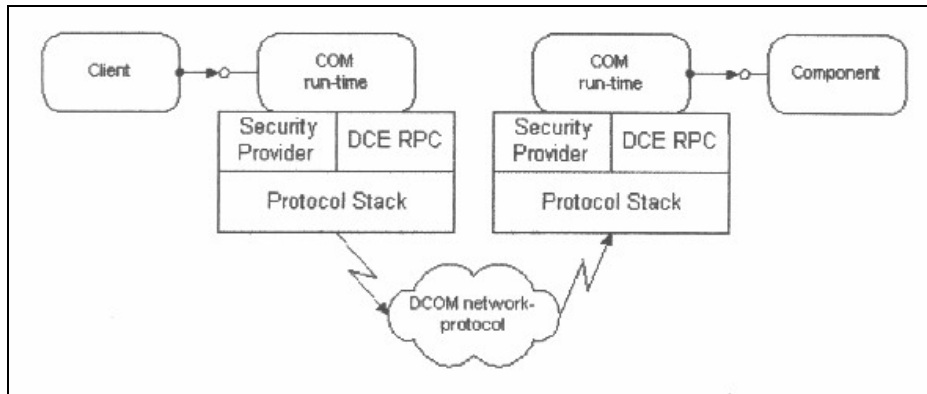
Το COM ορίζει πως αλληλεπιδρούν τα συστατικά και οι πελάτες τους. Αυτή η αλληλεπίδραση ορίζεται ως η δυνατότητα σύνδεσης πελάτη και συστατικού χωρίς την ανάγκη διαμεσολάβησης κάποιου άλλου συστατικού. Ο πελάτης καλεί μεθόδους στο συστατικό χωρίς να μεσολαβεί ή να βασίζεται σε κάτι άλλο. Στα σημερινά λειτουργικά συστήματα οι διαδικασίες είναι θωρακισμένες μεταξύ τους. Ένας πελάτης που χρειάζεται να επικοινωνήσει μ' ένα συστατικό σε μια άλλη διαδικασία δεν μπορεί να καλέσει άμεσα το συστατικό, αλλά πρέπει να χρησιμοποιήσει κάποια μορφή επικοινωνίας που παρέχεται από το λειτουργικό σύστημα. Το COM παρέχει αυτήν την επικοινωνία μ' ένα τελείως διαφανή τρόπο. Δέχεται τις κλήσεις από τον πελάτη και τις προωθεί στο συστατικό σε μια άλλη διαδικασία. Στο σχήμα 2.1 παρουσιάζεται πως οι COM/DCOM τρέχουσες βιβλιοθήκες παρέχουν τη σύνδεση μεταξύ πελάτη και συστατικού.



Σχήμα 2.1: COM συστατικά σε διαφορετικές διαδικασίες

Όταν πελάτης και συστατικό βρίσκονται σε διαφορετικές μηχανές, το DCOM απλώς αντικαθιστά την τοπική διαδικασία επικοινωνίας μ' ένα πρωτόκολλο δικτύου. Ούτε ο πελάτης ούτε το συστατικό γνωρίζουν ότι το καλώδιο που τους συνδέει είναι έγινε λίγο μακρύτερο. Το σχήμα 2.2 δείχνει την DCOM αρχιτεκτονική. Το COM run-time παρέχει αντικειμενοστραφείς υπηρεσίες σε πελάτες και συστατικά και

χρησιμοποιεί RPC (Remote Procedure Calls) από απόσταση διαδικαστικές κλήσεις και τον παροχέα ασφάλειας για τη δημιουργία πρότυπων δικτυακών πακέτων που προσαρμόζονται στο DCOM wire-protocol πρότυπο.



Σχήμα 2.2: DCOM/COM συστατικά σε διαφορετικά μηχανήματα

2.4. Συστατικά και επαναχρησιμοποίηση

Η υπάρχουσα υποδομή μηχανικού (h/w), το λογισμικό, τα συστατικά, καθώς επίσης και τα εργαλεία χρειάζεται να ολοκληρωθούν και να ισχυροποιηθούν ώστε να μειωθεί ο χρόνος και το κόστος ανάπτυξης και εκτέλεσης. Το DCOM άμεσα και διαφανώς δίνει πλεονέκτημα σε κάθε υπάρχουσα επένδυση COM συστατικών και εργαλείων. Οι δημιουργοί που είναι εξοικειωμένοι με το COM μπορούν εύκολα να εφαρμόσουν τις γνώσεις τους στις DCOM βασισμένες κατανεμημένες εφαρμογές.

Κάθε συστατικό που αναπτύχθηκε ως τμήμα μιας κατανεμημένης εφαρμογής είναι υποψήφιο για μελλοντική επαναχρησιμοποίηση. Οργανώνοντας τη διαδικασία ανάπτυξης γύρω από το παράδειγμα του συστατικού εννοείται η συνεχής άνοδος του επιπέδου λειτουργικότητας σε νέες εφαρμογές, καθώς επίσης και ο χρόνος παραγωγής καθώς χρησιμοποιείται προηγούμενη δουλειά.

2.5. Χαρακτηριστικά του DCOM

2.5.1. Ανεξαρτησία τοποθεσίας

Όταν ξεκινά η εκτέλεση μιας κατανεμημένης εφαρμογής σ' ένα πραγματικό δίκτυο εμφανίζονται αρκετοί αντικρουόμενοι περιορισμοί στη σχεδίαση, όπως τα συστατικά που αλληλεπιδρούν περισσότερο πρέπει να είναι πιο "κοντά" μεταξύ τους, κάποια συστατικά μπορούν να τρέξουν μόνο σε συγκεκριμένα μηχανήματα ή σε συγκεκριμένες τοποθεσίες, μικρότερα συστατικά αυξάνουν την ευελιξία της εκτέλεσης

αλλά επίσης αυξάνουν την κυκλοφορία στο δίκτυο, ενώ το αντίθετο συμβαίνει με τα μεγάλα συστατικά. Με το DCOM αυτοί οι κρίσιμοι σχεδιαστικοί περιορισμοί μπορούν εύκολα να αντιμετωπιστούν γιατί οι λεπτομέρειες της εκτέλεσης δεν καθορίζονται στον πηγαίο κώδικα. Το DCOM κρύβει την τοποθεσία του συστατικού είτε είναι στην ίδια διαδικασία όπως ο πελάτης ή σ' ένα μηχάνημα πολύ μακριά. Σ' όλες τις περιπτώσεις, ο τρόπος με τον οποίο ο πελάτης συνδέεται με το συστατικό και καλεί τις μεθόδους του συστατικού είναι ταυτόσημος. Όχι μόνο το DCOM δεν απαιτεί αλλαγές στον πηγαίο κώδικα, αλλά ούτε την επανάληψη της μετάφρασης του προγράμματος. Απλά, χρειάζονται κάποιες αλλαγές μορφοποίησης του τρόπου με τον οποίο συνδέονται τα συστατικά μεταξύ τους.

2.5.2. Ουδετερότητα γλώσσας

Ένα κοινό θέμα κατά τη διάρκεια και εκτέλεση μιας καταναμημένης εφαρμογής είναι η επιλογή της γλώσσας ή του εργαλείου για ένα δεδομένο συστατικό. Το DCOM ως επέκταση του COM είναι τελείως ανεξάρτητο της γλώσσας. Εικονικά κάθε γλώσσα μπορεί να χρησιμοποιηθεί για να δημιουργήσει COM συστατικά κι αυτά τα συστατικά μπορούν να χρησιμοποιηθούν από περισσότερες γλώσσες και εργαλεία. Οι Java, Microsoft Visual C++®, Microsoft Visual Basic®, Delphi, PowerBuilder and Micro Focus COBOL όλες αλληλεπιδρούν καλά με το DCOM.

2.5.3. Διαχείριση σύνδεσης

Οι δικτυακές συνδέσεις είναι περισσότερο εύθραυστες από τις συνδέσεις μέσα σ' ένα μηχάνημα. Τα συστατικά σε μια καταναμημένη εφαρμογή πρέπει να σημειώνονται αν ο πελάτης δεν είναι πλέον ενεργός, ακόμη και στην περίπτωση δικτυακής ή μηχανικής αποτυχίας. Το DCOM παρέχει ένα ισχυρό καταναμημένο μηχανισμό συλλογής απορριμμάτων που είναι τελείως διαφανής στην εφαρμογή. Είναι ένα κληρονομικά συμμετρικό δικτυακό πρωτόκολλο και μοντέλο προγραμματισμού. Όχι μόνο προσφέρει την παραδοσιακή μιας κατεύθυνσης πελάτη-εξυπηρετή αλληλεπίδραση, αλλά ακόμη παρέχει πλούσια αλληλεπιδραστική επικοινωνία μεταξύ πελατών και εξυπηρετών και μεταξύ ζευγών.

2.5.4. Κλιμάκωση

Ένας κρίσιμος παράγοντας για μια καταναμημένη εφαρμογή είναι η ικανότητά της να αυξάνεται με τον αριθμό των χρηστών, τη ποσότητα των δεδομένων και την

απαιτούμενη λειτουργικότητα. Η εφαρμογή πρέπει να είναι μικρή και γρήγορη όταν οι απαιτήσεις είναι ελάχιστες, αλλά πρέπει να είναι ικανή να διαχειριστεί επιπρόσθετες αιτήσεις χωρίς να θυσιάζονται η εκτέλεση ή η αξιοπιστία. Το DCOM παρέχει ένα αριθμό από χαρακτηριστικά που ενισχύουν την κλιμάκωση των εφαρμογών, όπως η παράλληλη εκτέλεση, η απομόνωση κρίσιμων συστατικών και η τεχνική pipelining των διαδοχικών διεργασιών.

Το DCOM δίνει το πλεονέκτημα στα Windows NT με την πολυεπεξεργασία. Οι DCOM εφαρμογές μπορούν εύκολα να κλιμακωθούν από τα μικρά απλά μηχανήματα επεξεργασίας στα μεγάλα συστήματα με δυνατότητες πολυεπεξεργασίας (*Συμμετρική Πολυεπεξεργασία, SMP, Symmetric Multiprocessing*).

Καθώς η φόρτωση σε μια εφαρμογή αυξάνεται, ακόμη και το γρηγορότερο μηχανήμα με πολυεπεξεργασία πιθανό να μην είναι σε θέση να ικανοποιήσει τη ζήτηση. Η ανεξαρτησία τοποθεσίας του DCOM επιτρέπει την εύκολη κατανομή συστατικών σε άλλους Η/Υ, χωρίς να απαιτούνται αλλαγές στον κώδικα, προσφέροντας ένα ευκολότερο και λιγότερο δαπανηρό τρόπο για κλιμάκωση.

Το DCOM παρέχει ευέλικτους μηχανισμούς εξέλιξης για τους πελάτες και τα συστατικά. Με το COM και το DCOM οι πελάτες μπορούν δυναμικά να ερωτούν τη λειτουργικότητα του συστατικού. Αντί να εκθέτει τη λειτουργικότητα ως ένα απλό, μονολιθικό σύνολο μεθόδων και ιδιοτήτων, το COM συστατικό μπορεί να εμφανιστεί διαφορετικά σε διαφορετικούς πελάτες. Ένας πελάτης που χρησιμοποιεί ένα συγκεκριμένο χαρακτηριστικό χρειάζεται πρόσβαση μόνο σε μεθόδους και ιδιότητες που χρησιμοποιεί. Οι πελάτες μπορούν επίσης να χρησιμοποιούν περισσότερο από ένα χαρακτηριστικό του συστατικού ταυτόχρονα. Εάν άλλα χαρακτηριστικά προστεθούν στο συστατικό δεν θα επηρεάσουν ένα παλιό πελάτη που δεν είναι ενημερωμένος μ' αυτά.

2.5.5. Εκτέλεση

Η κλιμάκωση δεν θα μπορεί να προσφέρει αν η αρχική εκτέλεση δεν είναι ικανοποιητική. Στο COM και DCOM ο πελάτης δεν βλέπει ποτέ το αντικείμενο εξυπηρέτη, αλλά κι ο πελάτης ποτέ δεν διαχωρίζεται από τον εξυπηρέτη από ένα συστατικό συστήματος εκτός κι αν αυτό είναι απαραίτητο. Αυτή η διαφάνεια επιτυγχάνεται με μια απλή ιδέα. Ο μόνος τρόπος με τον οποίο ένας πελάτης μπορεί να μιλήσει στο συστατικό είναι διαμέσου κλήσεων μεθόδων. Ο πελάτης φανερώνει τις διευθύνσεις αυτών των μεθόδων από ένα πίνακα που περιέχει αυτές τις διευθύνσεις.

Όταν ο πελάτης θέλει να καλέσει μια μέθοδο σ' ένα συστατικό φανερώνει τη διεύθυνση της μεθόδου και την καλεί. Όταν πελάτης και συστατικό βρίσκονται πολύ μακριά μεταξύ τους, σε διαφορετικά και απομακρυσμένα μηχανήματα, τότε το COM θέτει τον κώδικα υποδομής της δικής του απομακρυσμένης διαδικασίας κλήσης (Remote Procedure Call, RPC) στον πίνακα με τις διευθύνσεις. Έπειτα πακετάρει κάθε κλήση μεθόδου σε μια πρότυπη ενδιάμεση παρουσίαση, τη στέλνει στην πλευρά του συστατικού, εκεί την επαναφέρει στην αρχική μορφή κι έτσι γίνεται η κλήση της μεθόδου στο απομακρυσμένο συστατικό.

2.5.6. Εύρος ζώνης και εκκρεμότητα

Οι κατανεμημένες εφαρμογές δίνουν το πλεονέκτημα σ' ένα δίκτυο δένοντας τα συστατικά μεταξύ τους. Θεωρητικά, το DCOM κρύβει το γεγονός ότι τα συστατικά τρέχουν σε διαφορετικούς Η/Υ. Εν τούτοις στην πράξη, οι εφαρμογές χρειάζονται να λαμβάνουν υπόψη τους δύο πρωταρχικούς περιορισμούς μιας δικτυακής σύνδεσης. Κι αυτοί είναι το *εύρος ζώνης*, δηλαδή, το μέγεθος των παραμέτρων που περνούν σε μια κλήση μεθόδου άμεσα επηρεάζει το χρόνο που χρειάζεται για να ολοκληρώσει την κλήση και η *εκκρεμότητα*, δηλαδή, η φυσική απόσταση και ο αριθμός των εμπλεκόμενων δικτυακών στοιχείων (όπως οι δρομολογητές και οι επικοινωνιακές γραμμές) καθυστερούν σημαντικά ακόμη και τα μικρότερα πακέτα δεδομένων. Στην περίπτωση ενός παγκόσμιου δικτύου, όπως το Internet, αυτές οι καθυστερήσεις μπορεί να είναι της τάξης των δευτερολέπτων. Το DCOM ελαχιστοποιεί τις διαδρομές στο δίκτυο, όποτε αυτό είναι δυνατό, ώστε να αποφευχθεί η επίδραση της δικτυακής εκκρεμότητας. Το προτιμώμενο πρωτόκολλο μεταφοράς από το DCOM είναι το προσανατολισμένο χωρίς σύνδεση UDP υποσύνολο του TCP/IP πρωτοκόλλου. Η χωρίς σύνδεση φύση αυτού του πρωτοκόλλου επιτρέπει στο DCOM να εκτελεί αρκετές βελτιστοποιήσεις, συγχωνεύοντας αρκετά χαμηλού επιπέδου αναγνωρίσιμα πακέτα με πραγματικά δεδομένα και μηνύματα ελέγχου (pinging). Ακόμη, τρέχοντας και με προσανατολισμένα προς σύνδεση πρωτόκολλα, το DCOM ακόμη προσφέρει σημαντικά πλεονεκτήματα πάνω σε εξειδικευμένα για εφαρμογές συνήθη πρωτόκολλα. Ένα παράδειγμα ελέγχου είναι η ανάγκη του συστατικού να σημειώνει όταν το μηχανήμα του πελάτη υποστεί μηχανική βλάβη ή η δικτυακή σύνδεση μεταξύ πελάτη και συστατικού διακοπεί για ένα ορισμένο χρονικό διάστημα. Γι αυτό στέλνει σε τακτά χρονικά διαστήματα μηνύματα (pinging) κι αν ο εξυπηρέτης δεν δεχτεί για κάποιο συγκεκριμένο διάστημα ένα ping μήνυμα τότε δηλώνεται ως "νεκρός" ο πελάτης.

2.5.7. Ασφάλεια

Στις καταναμημένες εφαρμογές με τη χρήση δικτύου προκύπτει το θέμα της ασφάλειας, αφού μπορεί κάποιος να έχει πρόσβαση στο δίκτυο και να παρεμβληθεί στη λειτουργία πελάτη-συστατικού. Χωρίς την υποστήριξη ασφάλειας από την πλατφόρμα ανάπτυξης των καταναμημένων εφαρμογών, η καθεμιά θα έπρεπε να εκτελεί τους δικούς της μηχανισμούς ασφάλειας. Ένας τέτοιος τυπικός μηχανισμός περιλαμβάνει το πέρασμα του ονόματος χρήστη και του κωδικού πρόσβασης. Μια καταναμημένη πλατφόρμα υποχρεούται να παρέχει ένα πλαίσιο ασφάλειας ώστε να ξεχωρίζει τους διαφορετικούς πελάτες ή ομάδες πελατών, ώστε το σύστημα ή η εφαρμογή να έχει τον τρόπο να γνωρίζει ποιος προσπαθεί να εκτελέσει μια λειτουργία σ' ένα συστατικό. Το DCOM χρησιμοποιεί το επεκτάσιμο πλαίσιο ασφάλειας που παρέχεται από τα Windows NT. Ένα κεντρικό τμήμα του πλαισίου ασφάλειας είναι ένας φάκελος χρήστη ο οποίος αποθηκεύει τις απαραίτητες πληροφορίες αναγνώρισης ενός χρήστη. Επίσης και στις μη Windows NT πλατφόρμες παρέχεται ένας παρόμοιος μηχανισμός.

Κι ένας άλλος μηχανισμός ασφάλειας παρέχεται από το DCOM στις καταναμημένες εφαρμογές χωρίς κωδικούς πρόσβασης, με την αποθήκευση από το DCOM των καταλόγων ελέγχου πρόσβασης (ACLs, Access Control Lists) για τα συστατικά. Αυτοί οι κατάλογοι περιέχουν τα ονόματα των χρηστών που μπορούν να έχουν πρόσβαση σ' ένα συστατικό και μπορούν εύκολα να μορφοποιηθούν με χρήση του DCOM εργαλείου μορφοποίησης ή με πρόγραμμα χρησιμοποιώντας τις λειτουργίες Windows NT registry και Win32® security.

Ένα ευρύ φάσμα διαφορετικών παροχέων ασφάλειας μπορεί να χρησιμοποιηθεί από τις καταναμημένες εφαρμογές που βασίζονται στο DCOM χωρίς να απαιτείται καμιά αλλαγή ακόμη και στις ευαίσθητες σε ασφάλεια εφαρμογές.

2.5.8. Εξισορρόπηση φόρτισης

Μια καταναμημένη εφαρμογή είναι περισσότερο επιτυχημένη, όσο μεγαλύτερη μπορεί να είναι η φόρτιση που μπορεί να προκληθεί σε όλα τα συστατικά της εφαρμογής όταν αυξημένος αριθμός πελατών ζητά να εξυπηρετηθεί κι αν ακόμη το γρηγορότερο μηχανικό (h/w) δεν είναι αρκετό για την αυξημένη ζήτηση. Μια τεχνική για την εξισορρόπηση της φόρτισης είναι η μόνιμη ανάθεση συγκεκριμένων χρηστών σε συγκεκριμένους εξυπηρετητές που τρέχουν την ίδια εφαρμογή. Αυτή η τεχνική ονομάζεται *στατική εξισορρόπηση φόρτισης* επειδή η ανάθεση δεν αλλάζει με τις

συνθήκες που δημιουργούνται στο δίκτυο ή από άλλους παράγοντες. Αυτή η τεχνική είναι καλή όσο αυξάνεται η ζήτηση από τους χρήστες, αλλά απαιτεί την εμπλοκή του διευθύνοντα και δουλεύει καλά μόνο για αναμενόμενες φορτίσεις.

Η ιδέα του αναφερόμενου συστατικού μπορεί να χρησιμοποιηθεί για την παροχή ευφύτερης εξισορρόπησης φόρτισης. Αντί να βασίζεται η επιλογή του εξυπηρέτη από τον κωδικό του χρήστη, το αναφερόμενο συστατικό μπορεί να χρησιμοποιήσει πληροφορίες σχετικά με τη φόρτιση του εξυπηρέτη, την τοπολογία του δικτύου μεταξύ πελάτη και διαθέσιμων εξυπηρετών και στατιστικά στοιχεία που σχετίζονται με τις προηγούμενες αιτήσεις ενός συγκεκριμένου χρήστη. Κάθε φορά που ένας πελάτης συνδέεται μ' ένα συστατικό, το αναφερόμενο συστατικό μπορεί να το αναθέσει στον καταλληλότερο διαθέσιμο εξυπηρέτη εκείνη τη στιγμή, χωρίς ο πελάτης να το γνωρίζει. Αυτή η τεχνική ονομάζεται *δυναμική εξισορρόπηση φόρτισης*.

2.5.9. Ανεκτικότητα λάθους

Το DCOM παρέχει βασική υποστήριξη για ανεκτικότητα σε περιπτώσεις λάθους στο επίπεδο πρωτοκόλλου. Ένας εξελιγμένος ringing μηχανισμός αναγνωρίζει τα μηχανικά προβλήματα στο δίκτυο ή στην πλευρά του πελάτη. Αν το δίκτυο επανέλθει πριν το πέρας του χρόνου ανεκτικότητας, τότε το DCOM επανεγκαθιστά τη σύνδεση αυτόματα.

Σε μια άλλη περίπτωση, όταν οι πελάτες αναγνωρίζουν κάποιο πρόβλημα στο συστατικό, τότε επανασυνδέονται στο ίδιο αναφερόμενο συστατικό που εγκατέστησε την πρώτη σύνδεση. Αυτό γνωρίζει ποιοι εξυπηρέτες δεν είναι πλέον διαθέσιμοι και αυτόματα βρίσκει συστατικό σ' άλλο μηχάνημα για να εξυπηρετήσει τον πελάτη. Κι όλα αυτά γίνονται χωρίς να απασχολούν τον πελάτη κρατώντας την διαφάνεια στο σύστημα.

2.5.10. Ευκολία στην εκτέλεση

Η καλύτερη εφαρμογή θα είναι άχρηστη αν δεν μπορεί εύκολα να εγκατασταθεί και να διαχειριστεί. Όσον αφορά την απλοποίηση της εγκατάστασης στην πλευρά του πελάτη η κοινή προσέγγιση συνοψίζεται στην έννοια "λεπτός πελάτης". Βασίζεται στο ότι όσο μικρότερη λειτουργικότητα απαιτείται τόσο λιγότερα προβλήματα εγκατάστασης και συντήρησης θα προκύπτουν. Από την άλλη όμως μεριά είναι λιγότερο φιλική στο χρήστη όλη η εφαρμογή και ακόμη γίνονται περισσότερες αιτήσεις στο δίκτυο και στον εξυπηρέτη. Γι αυτό θα πρέπει να επιλεγθεί το κατάλληλο επίπεδο

"λεπτότητας" ώστε να συγκεραστούν όλα τα παραπάνω κι αυτή είναι μια κρίσιμη απόφαση στην ανάπτυξη μιας κατανεμημένης εφαρμογής.

Ακόμη, στις κατανεμημένες εφαρμογές είναι κρίσιμη η δυνατότητα κεντροποίησης της διαχείρισης, με ικανότητες αναγνώρισης πιθανόν αποτυχιών όσο γίνεται ταχύτερα και ιδιαίτερα πριν προκληθεί κάποια καταστροφή. Μέρος της εγκατάστασης και αναβάθμισης των συστατικών πελάτη είναι η μορφοποίηση των συστατικών και η συντήρηση της μορφοποίησης. Όσον αφορά το DCOM, η πιο απλή σημαντική μορφοποίηση πληροφορίας είναι το μηχανήμα του εξυπηρέτη που τρέχει τα συστατικά που αιτούνται από τον πελάτη. Κι αυτό μπορεί να διαχειριστεί από μια κεντρική θέση. Έτσι μπορεί να ενημερώνει όλους τους πελάτες διαφανώς, όταν γίνει μια απλή αλλαγή στην πληροφορία μορφοποίησης και στα πακέτα εγκατάστασης και το ίδιο συμβαίνει και στην περίπτωση του αναφερόμενου συστατικού. Για να επιτραπεί η ίδια DCOM υποδομή να χρησιμοποιηθεί για μορφοποίηση και λειτουργία των συστατικών, αυτά μπορούν να εκθέτουν επιπρόσθετες DCOM διεπαφές που είναι ορατές και προσβάσιμες μόνο στους διαχειριστές.

2.5.11. Ουδετερότητα πρωτοκόλλου

Πολλές κατανεμημένες εφαρμογές πρέπει να ενταχθούν στην δικτυακή υποδομή ενός πελάτη ή μιας επιχείρησης. Απαιτώντας ένα ειδικό δικτυακό πρωτόκολλο ίσως να απαιτηθεί μια αναβάθμιση όλων των δυνητικών χρηστών που δεν είναι συνήθως εφικτό. Γι αυτό θα πρέπει οι δημιουργοί των εφαρμογών να καταβάλλουν προσπάθεια ώστε η εφαρμογή να είναι όσο το δυνατόν ανεξάρτητη από την υποκείμενη υποδομή δικτύου. Το DCOM μπορεί να χρησιμοποιήσει κάθε πρωτόκολλο μεταφοράς, όπως το TCP/IP, UDP, IPX/SPX και NetBIOS. Το DCOM παρέχει ένα πλαίσιο ασφάλειας σ' όλα αυτά τα πρωτόκολλα, περιλαμβάνοντας και τα προσανατολισμένα προς σύνδεση και τα προσανατολισμένα χωρίς σύνδεση πρωτόκολλα. Οι δημιουργοί θα πρέπει να χρησιμοποιούν τα χαρακτηριστικά του DCOM και να σιγουρεύονται ότι οι εφαρμογές τους είναι τελείως ουδέτερες πρωτοκόλλου.

2.5.12. Ουδετερότητα πλατφόρμας

Μια κατανεμημένη εφαρμογή συχνά πρέπει να προσαρμόσει διαφορετικές πλατφόρμες τόσο στην πλευρά του πελάτη όσο και του εξυπηρέτη. Έτσι, οι δημιουργοί έρχονται αντιμέτωποι με σημαντικές διαφορές στη μορφή αυτών των πλατφόρμων, όπως διαφορά στη φιλοσοφία διεπαφής του χρήστη, στις υπηρεσίες του συστήματος και

ακόμη το σύνολο των διαθέσιμων δικτυακών πρωτοκόλλων δυσκολεύει στην προσαρμογή τους. Μια προσέγγιση για την αντιμετώπιση αυτού του προβλήματος είναι η επιλογή του μικρότερου κοινού παρανομαστή απ' όλες τις πλατφόρμες και η χρήση ενός αποσπασματικού επιπέδου για να κρατείται μια μικρή βάση κώδικα για όλες τις πλατφόρμες. Κι αυτό γίνεται από πολλά πλαίσια ανάπτυξης πλατφόρμων και ακόμη από εικονικά μηχανικά περιβάλλοντα, όπως για παράδειγμα η Java. Από την άλλη μεριά αυτό το αποσπασματικό επίπεδο εισάγει επιπρόσθετο βάρος και εμποδίζει τη χρήση ισχυρών εξειδικευμένων υπηρεσιών και βελτιστοποιήσεων. Για συστατικά διεπαφών χρήστη, αυτή η λύση συχνά σημαίνει φτώχη οπτική ολοκλήρωση με άλλες εφαρμογές, προκαλώντας μεγαλύτερες δυσκολίες στη χρήση και κατά συνέπεια αυξημένο κόστος εκπαίδευσης. Όσον αφορά τα συστατικά του εξυπηρέτη, η προσέγγιση αυτή θυσιάζει τη δυνατότητα να συντονιστεί η εκτέλεση των κρίσιμων συστατικών για κάθε πλατφόρμα.

Το DCOM είναι ανοικτό σε όλες τις προσεγγίσεις ανάπτυξης διασταυρούμενων πλατφόρμων, έχοντας την ίδια μορφή του συστήματος υπηρεσιών και χωρίς να εμποδίζει τη χρήση εξειδικευμένων σε μια πλατφόρμα υπηρεσιών ή βελτιστοποιήσεων. Η αρχιτεκτονική του DCOM επιτρέπει την ολοκλήρωση των πλαισίων ανάπτυξης ουδέτερων πλατφόρμων και εικονικών μηχανικών περιβαλλόντων για υψηλής εκτέλεσης, βελτιστοποιημένης πλατφόρμας συνήθων συστατικών σε μια απλή καταναεμημένη εφαρμογή.

Το DCOM καλωδιακό πρωτόκολλο βασίζεται στο DCE RPC, έτσι διευκολύνεται η εκτέλεση του DCOM σε πλατφόρμες για τις οποίες το DCE RPC είναι ήδη διαθέσιμο. Το DCE RPC ορίζει ένα αποδεδειγμένο πρότυπο για τη μετατροπή δομών δεδομένων και παραμέτρων μέσα στη μνήμη σε δικτυακά πακέτα. Η παρουσίαση των δεδομένων δικτύου (NDR, Network Data Representation) του είναι ουδέτερη πλατφόρμας και παρέχει ένα πλούσιο σύνολο φορητών τύπων δεδομένων. Τα COM και DCOM δανείζονται τη γραφή των παγκοσμίως μοναδικών ταυτοτικών (GUIDs, Globally Unique Identifiers) από το DCE RPC. Τα GUIDs παρέχουν ελευθερία συγκρούσεων, μη διαχειριζόμενη ονοματοδοσία των αντικειμένων και των διεπαφών και είναι η βάση των ανθεκτικών εκδόσεων του COM.

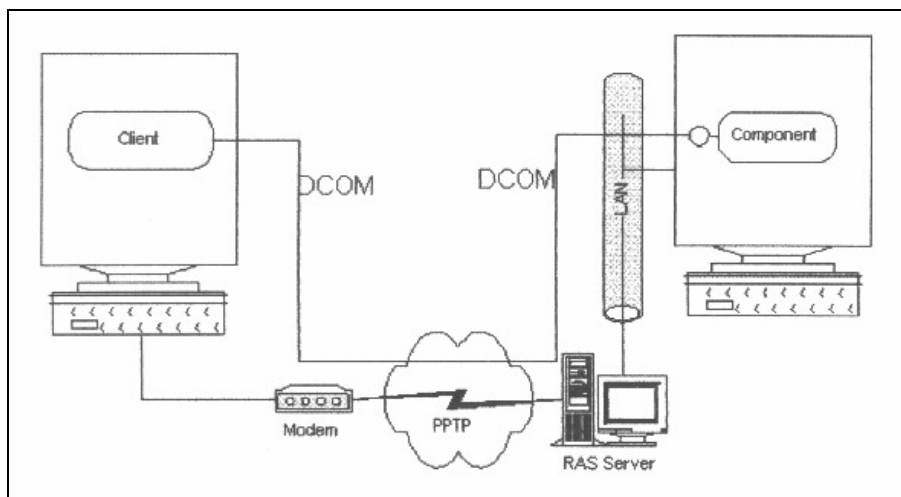
2.5.13. Ολοκλήρωση με άλλα Internet πρωτόκολλα

Στον πυρήνα του, το Internet είναι ένα παγκόσμιο, αποκεντροποιημένα διαχειριζόμενο και κοινής χρήσης TCP/IP δίκτυο που μετατρέπει την παγκόσμια

συνδεσιμότητα σ' ένα είδος. Η εφαρμογή που προκάλεσε τη δυνατότητα συγκέντρωσης τόσων χρηστών είναι μια απλή προτυποποιημένη γλώσσα περιγραφής σελίδας (HTML), μαζί μ' ένα απλό πρωτόκολλο φόρτωσης εγγράφων (HTTP). Οι καταναεμημένες εφαρμογές δίνουν πλεονέκτημα στο Internet με πολλούς διαφορετικούς τρόπους.

α. Το DCOM πάνω από εικονικά ιδιωτικά δίκτυα

Στο χαμηλότερο επίπεδο, χρησιμοποιώντας το Internet ως ένα φθινό, παγκόσμιο TCP/IP δίκτυο ανοίγονται νέες δυνατότητες για τις εταιρείες, ώστε να επιτευχθεί η σύνδεση απομακρυσμένων τμημάτων και μεμονωμένων χρηστών. Εικονικά ιδιωτικά δίκτυα, όπως το Windows NT 4.0 Point-to-Point Tunneling Protocol (PPTP), είναι ένας τρόπος χρήσης του δικτύου για τον ασφαλή καναλισμό ιδιωτικών πληροφοριών πάνω από Internet. Οι εφαρμογές που βασίζονται στο DCOM μπορούν με διαφάνεια να ισχυροποιήσουν τέτοια εικονικά ιδιωτικά δίκτυα.



Σχήμα 2.3. Το DCOM πάνω από εικονικά ιδιωτικά δίκτυα

β. Το DCOM πάνω από το Internet

Επειδή το DCOM είναι ένα κληρονομικά ασφαλή πρωτόκολλο μπορεί να χρησιμοποιηθεί χωρίς να ενθλακωθεί σ' ένα εικονικό ιδιωτικό δίκτυο. Οι DCOM εφαρμογές μπορούν απλά να χρησιμοποιήσουν τη φθινό, παγκόσμιο TCP/IP δίκτυο. Οι περισσότερες εταιρείες δεν παρέχουν άμεση πρόσβαση των υπολογιστών της στο Internet. Όλα και κάποια μηχανήματα-εξυπηρετή κρύβονται πίσω από ένα τοίχο ασφάλειας (Firewall), ο οποίος τυπικά αποτελείται από τα φίλτρα του επιπέδου πρωτοκόλλου (port-based) και του επιπέδου εφαρμογής (proxy servers).

Το DCOM μπορεί να δουλέψει καλά και με τις δύο κλάσεις των firewalls. Χρησιμοποιεί μια απλή θύρα για την έναρξη συνδέσεων και καταχωρεί ένα ευρύ φάσμα θυρών με δυνατότητα μορφοποίησης στα πραγματικά συστατικά που τρέχουν σ' ένα μηχανήμα (μία θύρα για κάθε διαδικασία).

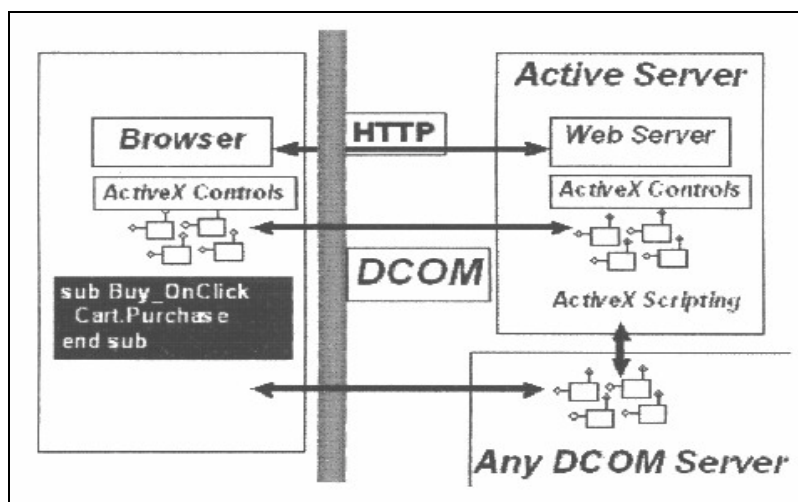
Τα επιπέδου εφαρμογής proxies μπορούν εύκολα να δημιουργηθούν. Μπορεί να είναι είτε γενικά είτε ειδικά στις εφαρμογές.

Οι διαχειριστές εξυπηρέτη μπορούν επίσης να επιλέξουν να καναλίσουν το DCOM διαμέσου του HTTP, παρακάμπτοντας αποτελεσματικά τα περισσότερα σημερινά firewalls.

Μ' αυτές τις δυνατότητες επιλογών, οι εφαρμογές που βασίζονται στο DCOM μπορούν να χρησιμοποιήσουν το Internet για ιδιωτικές συνδέσεις στο εσωτερικό μιας εταιρείας, ιδιωτική επικοινωνία με εξωτερικούς πελάτες και συνεργάτες και μαζικές συνδέσεις με κάθε πελάτη σ' όλο τον κόσμο. Σε κάθε περίπτωση το DCOM παρέχει ευέλικτη ασφάλεια όποτε χρειαστεί.

γ. Ολοκληρώνοντας το HTML και τα κατακεμημένα υπολογιστικά συστήματα

Οι κατακεμημένες εφαρμογές, χρησιμοποιώντας το Internet ως φθινό, παγκόσμιο TCP/IP δίκτυο, μπορούν ακόμη να δώσουν πλεονέκτημα στα υπάρχοντα προτυποποιημένα πρωτόκολλα. Για τις μη αλληλεπιδραστικές κειμένου ή απλών γραφικών πληροφοριών, οι σελίδες HTML είναι ένα σπουδαίο όχημα που παρέχει ένα εύκολο και κατανοητό τρόπο για τους χρήστες, ώστε να έχουν πρόσβαση στις πληροφορίες που ζητούν.



Σχήμα 2.4. Ολοκλήρωση HTML και κατακεμημένων υπολογιστικών συστημάτων

Για πιο πολύπλοκες, δομημένες και αλληλεπιδραστικές πληροφορίες τα συνήθη συστατικά μπορούν να επεκτείνουν την HTML σελίδα για να εκτελέσουν τις πραγματικές εργασίες των κατανεμημένων εφαρμογών μ' ένα φιλικό στο χρήστη, ασφαλή και αποτελεσματικό τρόπο. Απλοί κανόνες της επιχείρησης μπορούν να εφαρμοστούν στην πλευρά του πελάτη, παρέχοντας άμεση ανατροφοδότηση στο χρήστη. Περισσότερο πολύπλοκοι κανόνες της επιχείρησης μπορούν με διαφάνεια να εμπλέξουν τα συστατικά του εξυπηρέτη πάνω από το DCOM. Με την ουδετερότητα του DCOM ως προς τη γλώσσα, αυτά τα συστατικά μπορούν να εκτελεστούν εικονικά σε κάθε γλώσσα, όπως C++, Java, Visual Basic, Cobol. Τα υπάρχοντα που δεν κυκλοφορούν πλέον στο εμπόριο συστατικά (ActiveX controls) μπορούν να δεθούν με τα συνήθη συστατικά στην πλευρά του πελάτη, αλλά και στου εξυπηρέτη, χρησιμοποιώντας Visual Basic Script ή Jscript. Ανεξάρτητα με το εάν ο δημιουργός πλουτίζει μια κατανεμημένη εφαρμογή με HTML στοιχεία ή μια εφαρμογή βασισμένη σε HTML με στοιχεία κατανεμημένου υπολογιστικού συστήματος, το DCOM παρέχει την αναγκαία κόλλα συστατικού για να δέσει τους διαφορετικούς κόσμους.

3. DCOM και CORBA

Η CORBA ως ένα ανταγωνιστικό πρότυπο για το DCOM ορίζει ένα αποσπασματικό μοντέλο αντικειμένου που περιγράφει τα συστατικά και τις διεπαφές τους. Παρέχει επίσης πρότυπες αναπαραστάσεις από το αποσπασματικό ορισμό αντικειμένου σε σταθερές γλώσσες προγραμματισμού, αλλά δεν ορίζει ένα δυαδικό πρότυπο σε κάθε περίπτωση. Διαφορετικές ORB εκτελέσεις που προσχωρούν στο πρότυπο μπορούν να πετύχουν συμβατότητα σε επίπεδο πηγαίου κώδικα, αλλά όχι τη δυνατότητα ανταλλαγής των δυαδικών συστατικών κι αυτός είναι ένας από τους λόγους για τους οποίους δεν υπάρχει και πιθανό να μην υπάρχει μια αγορά για επαναχρησιμοποιήσιμα συστατικά που δεν κυκλοφορούν πλέον στο εμπόριο. Οι προμηθευτές των συστατικών πρέπει να παρέχουν πηγαίο κώδικα για τα συστατικά τους ή μετάφραση και έλεγχο για κάθε πλατφόρμα και για κάθε ORB εκτέλεση. Επίσης, η CORBA ορίζει ένα πρότυπο για Inter-ORB επικοινωνία που επιτρέπει δύο ORB εκτελέσεις να αιτούν μεθόδους σε αντικείμενα που βρίσκονται και στα δύο μηχανήματα. Οι ORB εκτελέσεις όπως IBM's DSOM και Iona's Orbix παρέχουν κοινά κατάλληλες επεκτάσεις στο μοντέλο αντικειμένου, τα επισυναπτόμενα γλώσσας και το Inter-ORB πρωτόκολλο. Οι δημιουργοί, για να έχουν πλήρως πλεονέκτημα για μια δεδομένη πλατφόρμα, πρέπει να θυσιάσουν τη μεταξύ των ORB διαλειτουργικότητα και δυνατότητα μεταφοράς. Επιπλέον, η CORBA ορίζει το IIOP που είναι ένα ξεχωριστό πρωτόκολλο Inter-ORB επικοινωνίας για το Internet και το οποίο εξυπηρετεί τη διασταύρωση όλων των Inter-ORB πρωτοκόλλων. Όπως ήδη αναφέρθηκε, η CORBA επινοήθηκε για να παρέχει μια νέα δικτυακή υποδομή με σκοπό την επίλυση των προβλημάτων διαλειτουργικότητας του λογισμικού αντικαθιστώντας τις υπάρχουσες υποδομές Διαδικασιών Κλήσης από Απόσταση (Remote Procedure Call, RPC) και τις προγενέστερες τεχνολογίες παρέχοντας μια αντικειμενοστραφή διεπαφή δικτύου η οποία απλοποιεί τα κατανεμημένα υπολογιστικά συστήματα. Ίσως να φαίνεται πιο σύνθετη και πολύπλοκη η δομή της σε σύγκριση με τις άλλες τεχνολογίες κατανεμημένων συστατικών αντικειμένων αλλά αυτό φαίνεται ότι της δίνει και το πλεονέκτημα της ολοκλήρωσης σ' ένα ευρύτερο πεδίο πλατφόρμων.

Από την άλλη μεριά η αρχιτεκτονική του DCOM φαίνεται πιο συμπαγής και απλή, παρέχοντας σ' όλες τις περιπτώσεις τον ίδιο τρόπο σύνδεσης και κλήσης των μεθόδων του συστατικού από τον πελάτη. Το DCOM καλωδιακό πρωτόκολλο βασίζεται στο DCE RPC, το οποίο ορίζει ένα αποδεδειγμένο πρότυπο για τη μετατροπή δομών δεδομένων και παραμέτρων μέσα στη μνήμη σε δικτυακά πακέτα.

Η CORBA και το DCOM είναι δύο ανταγωνιστικά πρότυπα στο χώρο των καταναμημένων αντικειμένων. Τα συστήματα διαχείρισης στον παγκόσμιο ιστό άρχισαν να βασίζονται σ' αυτά τα μοντέλα αντικειμένου σε συνδυασμό με τα εργαλεία υπέρ-μέσων και επεκτάσιμων γλωσσών σήμανσης. Έτσι τα συστήματα αυτά πλεονεκτούν και για ευρυζωνικά και για πολυμεσικά δίκτυα όπου τίθεται η καταναμημένη επεξεργασία και οι εφαρμογές πραγματικού χρόνου. Ίσως να μην έχει τόση σημασία ποιο από τα δύο είναι το καλύτερο, αφού και τα δύο δίνουν σημαντικό πλεονέκτημα στα καταναμημένα υπολογιστικά συστήματα και μπορούν να συνεχίσουν να βελτιώνονται ως προς την αρχιτεκτονική τους, ενώ το θέμα της αποτελεσματικότητας των εφαρμογών των υπολογιστικών συστημάτων θα παραμένει πέρα από τη δική τους εξέλιξη.

4. DISMAN (Distributed Management)

4.1. Τι είναι DISMAN

Ο Internet Society ISOC είναι ένας μη κερδοσκοπικός, μη κυβερνητικός, διεθνής, οργανισμός επαγγελματιών στελεχών. Μέλη του είναι πάνω από 175 οργανισμοί και 8600 πρόσωπα σε περισσότερες από 170 χώρες σ' όλο τον κόσμο. Το αντικείμενό του εστιάζεται σε 4 πυλώνες: πρότυπα, πολιτική κοινού, εκπαίδευση και εξάσκηση, στελέχωση. Όσον αφορά τα πρότυπα, μ' αυτά ασχολούνται οι Internet Engineering Task Force (IETF), Internet Architecture Board (IAB), Internet Engineering Steering Group (IESG) και Internet Research Task Force (IRTF).

Το IETF είναι μια μεγάλη ανοικτή διεθνής κοινότητα σχεδιαστών δικτύων, λειτουργών, προμηθευτών και ερευνητών που ενδιαφέρονται για την εξέλιξη της αρχιτεκτονικής του Internet και της ομαλής του λειτουργίας. Το IETF είναι ανοικτό για οποιονδήποτε ενδιαφέρεται να ενταχθεί σ' αυτό. Η τεχνική δουλειά γίνεται από τις ομάδες εργασίας που είναι οργανωμένες με βάση το θέμα τους σε περιοχές, όπως Εφαρμογών, Internet, Λειτουργίας και Διαχείρισης, Δρομολόγησης, Ασφάλειας, Μεταφοράς, Υπηρεσιών Χρήστη. Μία από τις ομάδες εργασίας της περιοχής Λειτουργίας και Διαχείρισης είναι και αυτή με αντικείμενο Distributed Management (DISMAN).

4.2. Περιγραφή της DISMAN ομάδας εργασίας

Η ομάδα εργασίας με αντικείμενο την Καταναμημένη Διαχείριση (Distributed Management, DISMAN) συγκροτήθηκε για να ορίσει ένα αρχικό σύνολο

διαχειριζόμενων αντικειμένων για εξειδικευμένες εφαρμογές κατανεμημένης δικτυακής διαχείρισης και ένα πλαίσιο, στο οποίο αυτές οι εφαρμογές και άλλες μπορούν συνεχώς να αναπτυχθούν και να εκτελεστούν. Ένας κατανεμημένος δικτυακός διαχειριστής είναι μια εφαρμογή που δρα με ρόλο διαχειριστή εκτελώντας λειτουργίες διαχείρισης και με ρόλο πράκτορα, έτσι ώστε να μπορεί να ελεγχθεί και να παρατηρηθεί από απόσταση.

Η ομάδα εργασίας περιορίζει τη δουλειά της στο αντικείμενο των εφαρμογών κατανεμημένης δικτυακής διαχείρισης, όπου ο κύριος μηχανισμός επικοινωνίας για τη λειτουργία και τον έλεγχο είναι το SNMP (Simple Network Management Protocol). Μελλοντική δουλειά (και άλλες ομάδες εργασίας) πιθανόν να αποτελέσει η έρευνα άλλων κατανεμημένων εφαρμογών, όπως η CORBA ή το HTTP. Τα αντικείμενα που ορίζονται από την ομάδα εργασίας θα είναι σύμφωνα με το SNMP πλαίσιο. Η ομάδα εργασίας, καθώς θα ορίζει τη διεπαφή στην κατανεμημένη διαχείριση θα λαμβάνει ειδικά υπόψη της την ασφάλεια. Ειδικότερα, η ομάδα εργασίας πρόκειται να ολοκληρώσει τις εξής εργασίες, όπως ο ορισμός μιας γραφής, ενός προγράμματος και ενός καταωφλίου λειτουργίας MIB (Management Information Base).

Το αντικείμενο λοιπόν αυτής της ομάδας εργασίας είναι πολύ στενό όσον αφορά την ευρύτητα του πεδίου της κατανεμημένης δικτυακής διαχείρισης. Κι αυτό γίνεται έτσι ώστε να αυξηθεί η πιθανότητα να παραχθεί χρήσιμο και ποιοτικό έργο σε συγκεκριμένο χρονικό διάστημα. Εν τούτοις, η ομάδα αυτή έχει κατά νου και σκοπεύει ως μελλοντική δουλειά τα ακόλουθα θέματα: Γεγονός και συναγερμός logging και κατανομή, Συλλογή ιστορικών δεδομένων, Τοπολογική εξερεύνηση.

4.3. Γιατί DISMAN

Η βιομηχανία απαιτεί περισσότερη αξιόπιστη διαχείριση λαθών από το πρωτόκολλο SNMP. Και γι αυτό πρέπει να υιοθετηθεί μια γενική λύση, ώστε να αποφεύγεται η κάθε εφαρμογή και η κάθε συσκευή να παρέχουν τη δική τους λύση. Εισερχόμενοι στο domain της κατανεμημένης διαχείρισης, υπάρχει η ανάγκη παροχής μιας γενικής υποδομής όσον αφορά την κατανεμημένη διαχείριση ενεργών συναγερμών και συμπληρωματικές δηλώσεις Log MIB για τη δημιουργία μια ολοκληρωμένης λύσης στην κατανεμημένη διαχείριση λαθών.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- Bapat Subodh, 1994, *Object Oriented Networks: Models for Architecture, Operations and Management*, PTR Prentice Hall
- Chisholm S., 2000, *Distributed Management of Active Alarms*, DISMAN Working Group 48th IETF
- Distributed Management (disman) Charter, *Description of Working Group*, <http://www.ietf.org/html.charters/disman-charter.html>
- Digital Equipment Corporation, 1994, *Digital Distributed Computing Environment (DCE) for Windows NT*
- Eddon G. and H. Eddon, 1998, *Understanding the DCOM Wire Protocol by Analyzing Network Data Packets*, <http://msdn.microsoft.com/library/periodic/period98/dcom.htm>
- Internet Engineering Task Force (IETF), *Overview of the IETF*, <http://www.ietf.cnri.reston.va.us/overview.html>
- Kavasseri R., 2000, *Event MIB*, Network Working Group, <http://www.ietf.cnri.reston.va.us/rfc/rfc2981.txt>
- King A. and R. Hunt, 2000, *Protocols and architecture for managing TCP/IP network infrastructures*, Computer Communications 23 (2000) 1558-1572
- Microsoft Corporation, 1996, *DCOM Technical Overview*, <http://msdn.microsoft.com/library/backgrnd/>
- Microsoft Corporation, 1997, *DCOM A Business Overview*, <http://msdn.microsoft.com/library/backgrnd/>
- Microsoft Corporation, 1996, *What is Active X?*, <http://msdn.microsoft.com/library/backgrnd/>
- Minton G., *IIOF Specification: A Closer Look*, <http://www.blackmagic.com/people/gabe/iiof.html>
- Mowbray J. T. and R. C. Malveau, 1997, *CORBA Design Patterns*, Wiley Computer Publishing
- Object Management Group, 1995, *The Common Object Request Broker: Architecture and Specification*, <http://www.infosys.tuwien.ac.at/Research/Corba/OMG/cover.htm>
- Οικονόμου Γ. Θ. κ.ά., 1999, *Εφαρμογές Λογισμικού*, Εκδόσεις Κορυφή Α.Ε.
- Schmidt C. D., *Developing Distributed Object Computing Applications with CORBA*, <http://www.eng.uci.edu/~schmidt>
- Schmidt C. D., *A High-Performance Architecture for Distributed Object Computing*, <http://www.cs.wustl.edu/~schmidt>
- Stallings W., 1993, *Network Management*, IEEE Computer Society Press
- Terplan K., 1999, *Web-based Systems & Network Management*, CRC Press