

ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ
ΟΙΚΟΝΟΜΙΚΩΝ & ΚΟΙΝΩΝΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΠΜΣ ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ

Εργασία για το μάθημα:

Πληροφορικά Συστήματα για
Μάρκετινγκ και Δημοσκοπήσεις

Θέμα: Σύγκριση των Mobile Agents
Theme: Comparison of Mobile Agents



Υπεύθυνος Καθηγητής: Οικονομίδης Αναστάσιος
Εισηγήτρια: Μάλαμα Αφροδίτη

ΘΕΣΣΑΛΟΝΙΚΗ
ΜΑΙΟΣ 2002

Contents

Abstract (English – Greek)	pag. 1
1. Introduction	pag. 2
2. Experimentation environment of mobile agent	pag. 4
3. Representative mobile agent systems	pag. 6
3.1 Multiple-language systems	pag. 6
3.2 Java based systems	pag. 9
3.3 Other systems	pag. 11
4. Similarities and differences	pag. 13
5. Conclusion	pag. 16
6. References	pag. 18

Περιεχόμενα

Περίληψη (Αγγλικά - Ελληνικά)	σελ. 1
1. Εισαγωγή	σελ. 2
2. Περιβάλλον ανάπτυξης του mobile agent	σελ. 4
3. Αντιπροσωπευτικά mobile agent συστήματα	σελ. 6
3.1 Συστήματα πολλαπλών γλωσσών προγραμματισμού	σελ. 6
3.2 Συστήματα βασισμένα στη Java	σελ. 9
3.3 Άλλα συστήματα	σελ. 11
4. Ομοιότητες και διαφορές	σελ. 13
5. Συμπεράσματα	σελ. 16
6. Αναφορές	σελ. 18

Abstract

A mobile agent is an executing program that can migrate, at times of its own choosing, from machine to machine in a heterogeneous network. On each machine, the agent interacts with other agents and other resources to accomplish its task. There are many mobile agent systems and they can be categorized in several ways. In the present paper are examining and categorizing ten representative mobile agent systems, according to the programming languages that they support and then discuss their similarities and differences.

Περίληψη

Ο mobile agent είναι εκτελέσιμο πρόγραμμα που μπορεί να μετακινηθεί, με δική του απόφαση, από μηχάνημα σε μηχάνημα σε ένα ετερογενές δίκτυο. Σε κάθε μηχάνημα, ο agent αλληλεπιδρά με άλλους agents αλλά και πόρους για να πραγματοποιήσει το σκοπό του. Υπάρχουν πολλά mobile agent συστήματα και μπορούν να ταξινομηθούν με πολλούς τρόπους. Στην παρούσα εργασία, εξετάζονται και ταξινομούνται δέκα αντιπροσωπευτικά συστήματα βάση των γλωσσών προγραμματισμού που μπορούν να υποστηρίξουν και στο τέλος παραθέτονται οι ομοιότητες και οι διαφορές τους.

1. Εισαγωγή

Η λέξη 'πράκτορας' (agent) προέρχεται από την ελληνική λέξη 'άγω' (άγειν/agein) που σημαίνει οδηγώ ή καθοδηγώ. Σήμερα ο όρος δηλώνει κάτι που παράγει ή είναι ικανό να παράγει αποτέλεσμα. Μπορεί να έχει δράση βιολογική, φυσική ή χημική. Από αυτή τη σκοπιά η λέξη 'πράκτορας' έχει νόημα και περιεχόμενο και στην ανθρώπινη φύση, για παράδειγμα ταξιδιωτικός πράκτορας. Με την συνεκδοχή του οδηγητή ή καθοδηγητή σε κάτι, ο όρος είναι κατάλληλος για να περιγράψει τις σημερινές τάσεις στην επιστήμη των υπολογιστών που δεν αντιμετωπίζουν τον Η/Υ σαν ένα παθητικό εργαλείο αλλά σαν ένα πραγματικά δυναμικό όργανο. Για να ξεχωρίσουν οι πράκτορες αυτοί με τα υπόλοιπα είδη πρακτόρων αναφέρονται ως πράκτορες λογισμικού. Οι πράκτορες λογισμικού ορίζονται από τρεις διαστάσεις: ευφυΐα, πρακτόρευση και κινητικότητα.

Η πρώτη διάσταση, η ευφυΐα, έχει τις ρίζες της στην έρευνα της Τεχνητής Νοημοσύνης και παραπέμπει στη δεκαετία του '50 όπου ο όρος 'πράκτορας' χρησιμοποιήθηκε για να ονομάσει ένα 'soft robot' που ζούσε και δρούσε μέσα σε υπολογιστή.

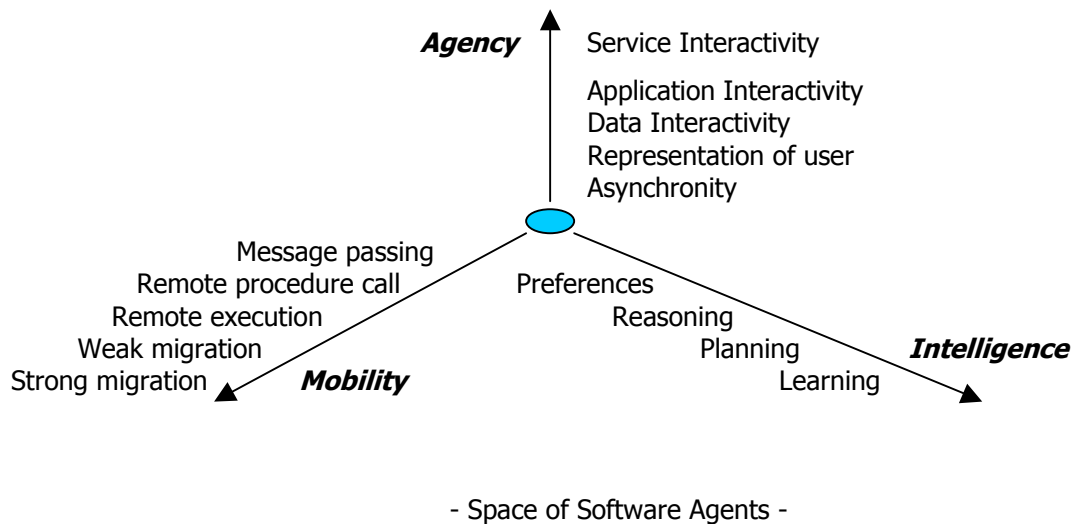
Η δεύτερη διάσταση, η πρακτόρευση, αναφέρεται στο βαθμό της αυτονομίας και δικαιοδοσίας που έχει ο πράκτορας και μπορεί να μετρηθεί, τουλάχιστον ποσοτικά, από τη φύση της αλληλεπίδρασης ανάμεσα στον πράκτορα και στις άλλες οντότητες του συστήματος.

Η τρίτη διάσταση, η κινητικότητα, εμφανίστηκε τη δεκαετία του '90 από την αλματώδη ανάπτυξη στο περιβάλλον της δικτύωσης των Η/Υ και της ανάγκης των τεχνικών για την εκμετάλλευσή τους. Η έρευνα αυτή απόφερε την απομακρυσμένη δράση και κινητικότητα των δεδομένων. Οι κινητοί πράκτορες μπορούν να διαχωριστούν, ανάλογα με την κινητικότητά τους, σε επιτραπέζιους (desktop) ή στατικούς πράκτορες, σε intranet πράκτορες, σε internet πράκτορες ή δικτυακούς πράκτορες. Αλλά τί ακριβώς είναι οι κινητοί πράκτορες;

«Οι κινητοί πράκτορες είναι οντότητες λογισμικού που αναλαμβάνουν ένα σύνολο λειτουργιών εκ μέρους του χρήστη ή κάποιου προγράμματος με κάποιο βαθμό ανεξαρτησίας ή αυτονομίας και καθώς προχωρούν αποκτούν γνώση ή αναπαράσταση των επιθυμιών και σκοπών του χρήστη» [IBM].

«Κινητοί πράκτορες είναι υπολογιστικά συστήματα που κατοικούν σε πολύπλοκα δυναμικά περιβάλλοντα, αισθάνονται και δρουν αυτόνομα σε αυτά τα περιβάλλοντα και μέσω της αυτονομίας πετυχαίνουν το σκοπό για τον οποίο σχεδιάστηκαν» [Pattie Maes ερευνήτρια στο MIT's Media Lab].

«Κινητοί πράκτορες είναι επίμονες οντότητες λογισμικού που είναι αφιερωμένοι σε συγκεκριμένο σκοπό. Ο χαρακτηρισμός 'επίμονες', διαχωρίζει τους πράκτορες από τις υπορουτίνες: οι πράκτορες έχουν τις δικές τους ιδέες για το πώς θα πραγματοποιήσουν ένα θέμα, τις δικές τους ατζέντες. Ο 'συγκεκριμένος σκοπός' τους διαχωρίζει από όλες τις πολυλειτουργικές εφαρμογές. Οι πράκτορες είναι πολύ μικρότεροι» [Smith, Cypher and Spohrer 1994, ερευνητές της Apple].



Έτσι ο κινητός πράκτορας αποτελεί οντότητα λογισμικού που:

- Είναι αυτόνομος και προσανατολισμένος στο σκοπό
- Μπορεί να δράσει δια λογαριασμό άλλου πράκτορα
- Αντιδράει στα εξωτερικά γεγονότα

Η ανάπτυξη του Internet έχει οδηγήσει τους ερευνητές στη μελέτη της κατασκευής κατακευμασμένων εφαρμογών στα large-scale συστήματα. Το κόστος επικοινωνίας σε αυτά τα συστήματα είναι κατά πολύ υψηλότερο των LANs και συχνά απρόβλεπτο. Επιπλέον, τα τμήματα και οι συνδέσεις των δικτύων είναι όλο και περισσότερο συχνές. Οι παράγοντες αυτοί οδήγησαν τους ερευνητές στον πειραματισμό νέων παραδειγμάτων για το κτίσιμο κατακευμασμένων εφαρμογών σε τέτοια δίκτυα. Σκοπός είναι η μείωση στα κόστη επικοινωνίας και να καταστούν δυνατές οι προβληματικές λειτουργίες.

Οι Mobile agents αποτελούν λύση σε αυτά τα προβλήματα. Ο mobile agent έχει τη δυνατότητα να κινεί τον κώδικά του και το περιεχόμενο εκτέλεσης από ένα μηχάνημα σε άλλο. Έτσι, μειώνονται τα επικοινωνιακά κόστη αφού μεταφέρονται οι υπολογιστικές διαδικασίες στο μηχάνημα όπου βρίσκονται τα δεδομένα-στόχοι. Οι προβληματικές λειτουργίες και συνδέσεις μπορούν να ξεπεραστούν αφού ένας πράκτορας μπορεί να μετακινείται και να επισκέπτεται διάφορους hosts χωρίς να κρατάει τη σύνδεση στον αρχικό host.

Οι κινητοί πράκτορες είναι πράκτορες που μπορούν να μεταναστεύσουν μεταξύ φυσικών κόμβων. Το βασικό χαρακτηριστικό τους είναι η κινητικότητα κι όχι η ευφυΐα. Η κινητικότητά τους δανείζεται αρκετά στοιχεία από τη διαδικασία μετανάστευσης. Αποτελούν ένα συναρπαστικό παράδειγμα για σχεδίαση και εφαρμογή των δυναμικών κατανεμημένων συστημάτων. Είναι προγράμματα ικανά για να αποφασίζουν την μετανάστευσή τους σε άλλους hosts. Κατά τη διάρκεια της μετακίνησης ο πράκτορας καθώς επίσης και ο κώδικάς του πρέπει να διαβιβαστούν. Ο υψηλός βαθμός ευελιξίας και προσαρμοστικότητας κάνει τους κινητούς πράκτορες ένα ελπιδοφόρο πρότυπο λογισμικού για τη νέα γενιά των ισχυρών κινητών πλατφόρμων με την ενσωμάτωση της ικανότητάς τους για τη μετανάστευση μέσα σε ένα δίκτυο με την ικανότητα της πλατφόρμας να συνοδεύει τον τελικό χρήστη στην καθημερινή ζωή.

2. Περιβάλλον ανάπτυξης του mobile-agent

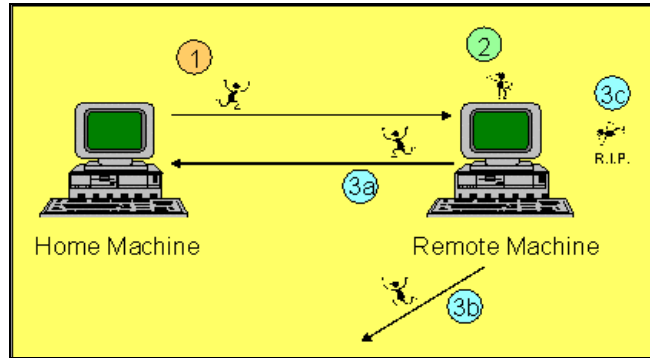
Ένας πράκτορας αποτελείται από τρία μέρη:

- τον κώδικα, ο οποίος αντιστοιχεί σε ορισμένο αλγόριθμο
- την thread εκτέλεση (με μια στοίβα εκτέλεσης)
- τα δεδομένα που αντιστοιχούν στις τιμές όλων των μεταβλητών του πράκτορα

Όλα αυτά τα μέρη κινούνται όποτε κινείται ο πράκτορας. Γενικά, προκειμένου να κινηθεί προς μια μηχανή, ένας πράκτορας πρέπει ρητά να καλέσει μια 'αρχή' κίνησης (μηχανή).

Η μετακίνηση ενός πράκτορα σε έναν server αποτελείται από τα εξής στάδια:

- αναφορά/δημοσίευση του πράκτορα
- δημιουργία μηνύματος από τον κώδικα και μέρη δεδομένων του πράκτορα
- αποστολή του μηνύματος στον εξυπηρετητή προορισμού
- καταστροφή του πράκτορα στον αρχικό εξυπηρετητή
- αναφορά/δημοσίευση του πράκτορα στον εξυπηρετητή προορισμού
- φόρτωση του κώδικα του εισερχόμενου πράκτορα
- δημιουργία νέου νήματος (thread) για την εκτέλεση του πράκτορα
- ξεκίνημα του πράκτορα (με το βασικό σημείο εισόδου)
- ο πράκτορας ψάχνει για σημείο εισόδου στον νέο εξυπηρετητή



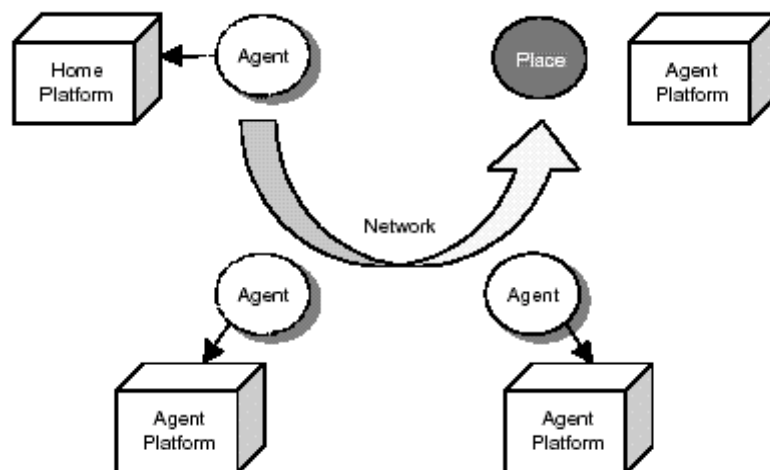
- Mobile Agents -

Όταν ένας πράκτορας κινείται από μια μηχανή προς άλλη, πρέπει να συνεχίσει την εκτέλεσή του στον κόμβο προορισμού σύμφωνα με την οδηγία που ακολουθεί αμέσως μετά από την επίκληση της 'αρχής' κίνησης. Στην πρότυπη Java Virtual Machine όμως, δεν παρέχεται η μετανάστευση των threads και έτσι τα περισσότερα συστήματα κινητών πρακτόρων, που εφαρμόζονται στην κορυφή της Java, παρέχουν ελλιπή μετανάστευση [Baumann97], δηλ. ένας πράκτορας κινείται με τον κώδικα και τα στοιχεία του χωρίς τη στοίβα του thread state. Αυτό υπονοεί ότι ένας πράκτορας επανεκκινεί την εκτέλεση από την αρχή (βασικό σημείο εισόδου) κάθε φορά που κινείται προς μια μηχανή. Συνεπώς, ο προγραμματιστής πρέπει να περιλάβει μέσα στον πράκτορα κάποιες πληροφορίες στοιχείων που του επιτρέπουν να ανιχνεύσει και επισημάνει το στάδιο εκτέλεσης. Κατόπιν, όταν φθάσει ο πράκτορας σε μια μηχανή προορισμού, ο αλγόριθμος του πράκτορα πρέπει να ελέγξει αυτό το στάδιο εκτέλεσης έτσι ώστε το κατάλληλο μέρος του κώδικα να εκτελεστεί σε εκείνη την μηχανή.

Γενικά, ένα σύστημα κινητών πρακτόρων παρέχει τις βασικές αρχές που επιτρέπουν στους κινητούς πράκτορες να επικοινωνήσουν μεταξύ τους αλλά και με τους εξυπηρετητές στις μηχανές που επισκέπτονται. Αυτές οι βασικές αρχές επικοινωνίας χρησιμοποιούν τη μεταβίβαση μηνυμάτων ή/και τις κλήσεις διαδικασιών.

3. Αντιπροσωπευτικά mobile-agent συστήματα

Η παρούσα και μελλοντική δυνατότητα των κινητών πρακτόρων έχει οδηγήσει σε μια αναταραχή των πρόσφατων εργασιών εφαρμογής. Σε αυτό το τμήμα, θα εξεταστούν αρχικά δέκα αντιπροσωπευτικά συστήματα κινητών πρακτόρων και έπειτα οι ομοιότητες και οι διαφορές τους.



- Agent System model -

Τα συστήματα κινητών πρακτόρων μπορούν να ταξινομηθούν με διάφορους τρόπους. Εδώ θα ταξινομηθούν σύμφωνα με τις γλώσσες προγραμματισμού που υποστηρίζουν. Μερικά συστήματα επιτρέπουν στους πράκτορες να γραφτούν σε πολλαπλές γλώσσες προγραμματισμού, άλλα επιτρέπουν στους πράκτορες να γραφτούν μόνο στη Java, η οποία είναι η δημοφιλέστερη γλώσσα πρακτόρων και άλλοι επιτρέπουν στους πράκτορες να γραφτούν σε κάποια γλώσσα προγραμματισμού διαφορετικής της Java.

3.1 Συστήματα πολλαπλών γλωσσών προγραμματισμού

Μερικά συστήματα προσπαθούν να υποστηρίξουν τους κινητούς πράκτορες έξω από το πλαίσιο οποιασδήποτε γλώσσας προγραμματισμού, και να υποστηρίξουν πράγματι περισσότερες από μια γλώσσες προγραμματισμού.

Ara. Το Ara [PS97,Pei98] (Πανεπιστήμιο του Kaiserslautern, Γερμανία) υποστηρίζει τους πράκτορες που γράφονται σε Tcl, σε Java και C/C++. Οι πράκτορες C/C++ μεταγλωττίζονται και διερμηνεύονται. Η μεταγλώττιση του πηγαίου προγράμματος έχει αποτέλεσμα την παραγωγή ειδικού κώδικα, που ονομάζεται κώδικας byte (bytecode), ενώ ο κώδικας ολόκληρος ονομάζεται MACE. Ο κώδικας αυτός (παρά ο ίδιος ο κώδικας C/C++) στέλνεται από μηχανή σε μηχανή. Και για τις τρεις γλώσσες το Ara παρέχει οδηγία για τη διαβίβαση, που συλλαμβάνει αυτόματα την πλήρη κατάσταση

του πράκτορα, την μεταφέρει στη μηχανή προορισμού και επαναλαμβάνει την εκτέλεση πρακτόρων στο ακριβές σημείο. Το Ara επιτρέπει επίσης στον πράκτορα να ελέγχει την εσωτερική κατάστασή του οποιαδήποτε στιγμή κατά τη διάρκεια της εκτέλεσής. Αντίθετα από άλλα πολύ-γλωσσικά συστήματα, το σύστημα Ara είναι multi-threaded: ο εξυπηρετητής πράκτορας και οι διερμηνευτές των Tcl, Java και MACE τρέχουν μέσα σε ενιαίο περιβάλλον UNIX. Αν και αυτή η προσέγγιση περιπλέκει την εφαρμογή, έχει σημαντικά πλεονεκτήματα απόδοσης: όταν φθάνει ένας νέος πράκτορας, αρχίζει απλά την εκτέλεση σε ένα νέο νήμα και όταν θέλει ένας πράκτορας να επικοινωνήσει με άλλον, μεταφέρει απλά και μόνο τη δομή των μηνυμάτων στον πράκτορα-στόχο.

Το σύστημα Ara έχει προβλέψει και για τους αρχικούς μηχανισμούς ασφαλείας. Ο κώδικας ενός πράκτορα υπογράφεται κρυπτογραφικά από τον κατασκευαστή του (προγραμματιστή), ενώ τα ορίσματα και το γενικό επίδομα των στοιχείων συμπεριφοράς του υπογράφονται από τον ιδιοκτήτη του (χρήστη). Κάθε μηχανή έχει μια ή περισσότερες εικονικές θέσεις, οι οποίες δημιουργούνται από τους πράκτορες και έχουν προσδιορίσει τις λειτουργίες αποδοχής. Ένας πράκτορας που μετακινείται πρέπει να εισαγάγει μια συγκεκριμένη θέση. Όταν εισάγει τη θέση, η λειτουργία αποδοχής απορρίπτει τον πράκτορα ή του αναθέτει ένα σύνολο χορηγήσεων (allowances) βασισμένων στα κρυπτογραφημένα πιστοποιητικά της. Αυτές οι χορηγήσεις, που περιλαμβάνουν πρόσβαση στα συστήματα αρχείων και συνολική μνήμη, επιβάλλονται έπειτα στα εξωτερικά μέρη των λειτουργιών πρόσβασης.

D'Agents. Το σύστημα D'Agents [GKCR98] κινητών πρακτόρων, που ήταν αρχικά γνωστό ως πράκτορας Tcl, υποστηρίζει τους πράκτορες που γράφονται σε Tcl, σε Java και Scheme, παρόμοια με τους στάσιμους πράκτορες που γράφονται σε C και C++. Όπως στο σύστημα Ara, το D'Agent παρέχει οδηγία για τον δρόμο για κάθε γλώσσα και ταυτόχρονα συλλαμβάνει αυτόματα και αποκαθιστά το thread state ενός μεταναστεύοντα πράκτορα. Αντίθετα όμως από το Ara, μόνο ο εξυπηρετητής D'Agent είναι multi-threaded: κάθε πράκτορας εκτελείται σε μια ξεχωριστή διαδικασία, πράγμα που απλοποιεί την εφαρμογή αρκετά αλλά επιβαρύνει την εσωτερική επικοινωνία. Ο D'Agent server χρησιμοποιεί δημόσιο κλειδί στην κρυπτογράφηση για την πιστοποίηση ταυτότητας ενός πράκτορα που καταφτάνει: τμηματικά οι resource-manager πράκτορες αναθέτουν τα δικαιώματα πρόσβασης στον πράκτορα, βάση της αυθεντικότητας και των προτιμήσεων του διαχειριστή και οι συγκεκριμένες γλωσσικά ενότητες επιβάλλουν τα δικαιώματα πρόσβασης, είτε αποτρέπουν μια παραβίαση από την εμφάνιση (π.χ. πρόσβαση σε filesystem), είτε τερματίζουν τον πράκτορα όταν εμφανίζεται μια παραβίαση (π.χ. υπέρβαση του συνολικού χρόνου CPU). Κάθε resource-manager συνδέεται με ένα συγκεκριμένο στοιχείο πηγής όπως το filesystem. Οι resource-

manager μπορεί να είναι σύνθετοι, αλλά οι προκαθορισμένοι resource-manager συνδέουν απλά έναν κατάλογο δικαιωμάτων πρόσβασης με κάθε ιδιοκτήτη.

Αντίθετα από το Ara, οι περισσότεροι resource-manager δεν ερωτώνται όταν καταφθάνει ο πράκτορας, αλλά μόνο όταν ζητούν ρητά οι πράκτορες: 1) να έχουν πρόσβαση στους αντίστοιχους πόρους ή 2) στο συγκεκριμένο δικαίωμα πρόσβασης. Σε αυτό το σημείο, ο υπεύθυνος των πόρων διαβιβάζει όλα τα σχετικά δικαιώματα πρόσβασης.

Η μελέτη για τους D'Agents χωρίζεται σε τρεις κατηγορίες: (1) εξελισσιμότητα, (2) έλεγχος στην αγορά των πόρων, (3) υποστήριξη των κινούμενων υπολογιστικών περιβαλλόντων, όπου οι εφαρμογές έχουν να κάνουν με χαμηλό εύρος ζώνης και με αναξιόπιστες συνδέσεις δικτύου.

Οι D'Agents χρησιμοποιούνται για εφαρμογές εξόρισης πληροφοριών, για αναφορές τεχνικής φύσεως και στο σύστημα 3Dbase [CBC97] για τρισδιάστατο σχεδιασμό μηχανικών τμημάτων (CAD drawings).

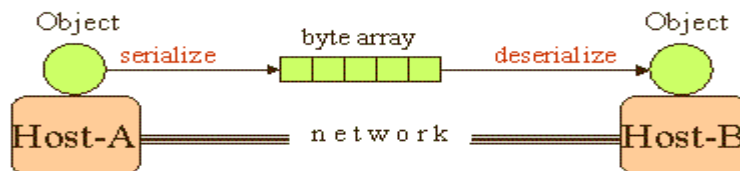
Tacoma. Το Tacoma [JSvR98a,JSvR98b] υποστηρίζει πράκτορες που γράφονται σε C, C++, ML, Perl, Python και διάφορες άλλες γλώσσες. Αντίθετα από το Ara και το D'Agents, το Tacoma δεν παρέχει την αυτόματη σύλληψη όλων των τμημάτων του πράκτορα. Αλλά, όταν θέλει ένας πράκτορας να μετακινηθεί σε μια νέα μηχανή, δημιουργεί έναν φάκελο που συσκευάζει τον κώδικά του και όλες τις επιθυμητές πληροφορίες. Ο φάκελος στέλνεται στη νέα μηχανή, όπου δημιουργείται το απαραίτητο περιβάλλον εκτέλεσης και καλεί έπειτα ένα γνωστό σημείο εισόδου μέσα από τον κώδικα του πράκτορα, για να επαναλάβει την εκτέλεση. Αν κι αυτή η προσέγγιση τοποθετεί το φορτίο ακριβώς επάνω στον προγραμματιστή, επιτρέπει όμως τη γρήγορη ένταξη των νέων γλωσσών στο σύστημα Tacoma, αφού οι υπάρχοντες διερμηνευτές και οι εικονικές μηχανές μπορούν να χρησιμοποιηθούν χωρίς καμία τροποποίηση.

Το Tacoma χρησιμοποιείται ειδικότερα στο StormCast, ένα κατανεμημένο μετεωρολογικό σύστημα ελέγχου και στον Tacoma Image Server, ένα σύστημα ανάκτησης δορυφορικών εικόνων. Κι οι δύο εφαρμογές περιλαμβάνουν μεγάλα ποσά στοιχείων και γρήγορες μεταβαλλόμενες ανάγκες των χρηστών. Αντί να φορτώνονται όλα τα δεδομένα για τοπική επεξεργασία, οι εφαρμογές των χρηστών εφαρμόζουν τις επιθυμητές -υψηλού επιπέδου- διαδικασίες αναζήτησης ως κινητούς πράκτορες και στέλνουν έπειτα αυτούς στις κατάλληλες τοποθεσίες.

Το μειονέκτημα του Tacoma είναι ότι δεν εγγυάται την ασφάλεια βάση κρυπτογράφησης αλλά έχουν αναπτυχθεί άλλοι μέθοδοι ελέγχου όπως: 1) συνεργασία με άλλους πράκτορες για το ψάξιμο αντιγράφων βάσεων δεδομένων σε παραλληλία και 2) χρησιμοποιώντας τα αυτόματα συστήματα ασφάλειας για τον καθορισμό της πολιτικής ασφάλειας μηχανής κι έπειτα γίνεται η απομόνωση βλαβών λογισμικού.

3.2 Συστήματα βασισμένα στη Java

Η Java είναι μια όλο και περισσότερο δημοφιλής γλώσσα για την ανάπτυξη των κατανεμημένων εφαρμογών. Η κοινή εφαρμογή της ως VM επιτρέπει στα προγράμματά της να τρέξουν σε μια ευρεία ποικιλία πλατφόρμων. Επιπλέον, έχει την ενσωματωμένη υποστήριξη για serialization, ασφάλεια, απομακρυσμένη επίκληση μεθόδου και δυναμική φόρτωση της κλάσης. Όλοι αυτοί οι παράγοντες κάνουν την Java μια φυσική επιλογή για τους κινητούς πράκτορες. Διάφορα συστήματα έχουν αναπτυχθεί με την Java.



- Mobile Agents in Java -

Aglets. Το Aglets [LO98] (IBM Τόκιο, Ιαπωνία) είναι ένα από τα πρώτα βασισμένα στη Java συστήματα. Όπως όλα τα εμπορικά συστήματα, (π.χ. Concordia), οι Aglets δεν συλλαμβάνουν το thread state ενός πράκτορα κατά τη διάρκεια της μετανάστευσης, δεδομένου ότι η σύλληψή του απαιτεί τροποποιήσεις στην πρότυπη JVM. Με άλλα λόγια, σύλληψη thread state σημαίνει ότι το σύστημα μπορεί να χρησιμοποιηθεί μόνο σε μια συγκεκριμένη VM. Έτσι, αντίθετα με τα συστήματα Ara και D'Agents, οι Aglets χρησιμοποιούν παραλλαγές του συστήματος Tacoma, όπου η εκτέλεση του πράκτορα επανεκκινείται από ένα γνωστό σημείο εισόδου μετά από κάθε μετακίνηση. Ειδικότερα, οι Aglets χρησιμοποιούν ένα μοντέλο 'οδηγημένου γεγονότος' (event-driven model). Όταν ένας πράκτορας θέλει να μεταναστεύσει, καλεί το μοντέλο αποστολών (dispatch model). Το σύστημα Aglets καλεί τη μέθοδο onDispatching του πράκτορα, η οποία εκτελεί τον καθορισμένο από την εφαρμογή καθαρισμό, καταστρέφει τα threads του πράκτορα, δημοσιοποιεί τον κώδικα και τον στέλνει στη νέα μηχανή. Εκεί, το σύστημα καλεί τη μέθοδο onArrival του πράκτορα, που εκτελεί την καθορισμένη από την εφαρμογή έναρξη και καλεί έπειτα τη μέθοδο run του πράκτορα για να επανεκκινήσει την εκτέλεση πρακτόρων.

Οι Aglets χαρακτηρίζονται από εμμονή, η οποία επιτρέπει σε έναν πράκτορα να γράψει τον κώδικα και να ορίσει τα αντικείμενά του στη δευτερεύουσα μνήμη και προσωρινά να ενεργοποιήσει τον ίδιο. Οι διαμεσολαβητές (proxies), που ενεργούν ως αντιπρόσωποι για τους Aglets, παρέχουν διαφάνεια της θέσης' μια υπηρεσία εντοπισμού των κινούμενων Aglets και μια σειρά από ευκολίες για την ανταλλαγή μηνυμάτων μεταξύ των πρακτόρων. Το μοντέλο ασφάλειας των Aglets είναι παρόμοιο με τα αντίστοιχα των Ara και D'Agent. Ένας Aglet έχει και ιδιοκτήτη και κατασκευαστή.

Όταν ο πράκτορας εισάγει ένα πλαίσιο σε κάποιο μηχάνημα, το πλαίσιο αναθέτει ένα σύνολο αδειών στον πράκτορα βασισμένο στην αυθεντικότητα του ιδιοκτήτη και του κατασκευαστή.

Concordia. Το Concordia [WPW+97,WPW98] είναι ένα σύστημα κινητών πρακτόρων βασισμένο στη Java που εστιάζει στην ασφάλεια και στην αξιοπιστία. Όπως τα περισσότερα βασισμένα στη Java συστήματα, κινεί τον κώδικα του αντικειμένου και τα στοιχεία, αλλά όχι το thread state, από μια μηχανή προς άλλη. Οι πράκτορες Concordia συσσωρεύονται σε διάφορα μέρη καθώς δρομολογούνται, ενώ επιτρέπουν στον πράκτορα να καθορίσει διαφορετική μέθοδο σε κάθε σταμάτημα. Οι πράκτορες, τα γεγονότα και τα μηνύματα μπορούν να περιμένουν στη σειρά εάν η απομακρυσμένη περιοχή δεν είναι προσβάσιμη. Οι πράκτορες αποθηκεύονται προσεκτικά, πριν την αναχώρηση και μετά την άφιξη στην καινούργια τοποθεσία ώστε να αποφευχθεί η απώλειά τους σε περίπτωση βλάβης του μηχανήματος. Οι πράκτορες προστατεύονται μέσω της κρυπτογράφησης κατά τη μετάδοση ή αποθηκεύονται σε δίσκο.

Jumping Beans. Το σύστημα Jumping Beans [AA98] είναι ένα ακόμα βασισμένο στη Java. Οι υπολογιστές που επιθυμούν να φιλοξενήσουν κινητούς πράκτορες τρέχουν μια αντιπροσωπεία Jumping Beans, η οποία συνδέεται με μια δικτυακή γειτονιά Jumping Beans. Κάθε τέτοια γειτονιά έχει έναν κεντρικό υπολογιστή, ο οποίος εξακριβώνει τους πράκτορες που ανήκουν στην ιδιοκτησία. Μέσα στην γειτονιά οι πράκτορες κινούνται από αντιπροσωπεία σε αντιπροσωπεία και μόνο αυτοί μπορούν να στείλουν μηνύματα σε άλλους πράκτορες. Οι μηχανισμοί εφαρμόζονται με τη διαβίβαση μέσω του εξυπηρετητή. Έτσι ο εξυπηρετητής γίνεται το κεντρικό σημείο για τον εντοπισμό, τη διαχείριση και την επικύρωση των πρακτόρων. Γίνεται επίσης το κεντρικό σημείο των αποτυχιών ή μιας δυσχέρειας απόδοσης, αν και γίνεται προσπάθεια ανάπτυξης εξελικτικών εξυπηρετητών που τρέχουν σε παράλληλες μηχανές. Μια άλλη προσέγγιση στην εξελισιμότητα είναι να δημιουργηθούν πολλές μικρές δικτυακές γειτονιές, κάθε μια με τον δικό της εξυπηρετητή. Στην τελευταία έκδοση, οι πράκτορες δεν μπορούν να μεταναστεύσουν μεταξύ των δικτυακών γειτονιών, αλλά σε μελλοντικές εκδόσεις θα γίνει εφικτό. Το δημόσιο κλειδί χρησιμοποιείται για την αυθεντικότητα των πρακτόρων στον εξυπηρετητή και αντίστροφα και οι λίστες του ελέγχου πρόσβασης χρησιμοποιούνται για να ελέγξουν την πρόσβαση των πρακτόρων στους πόρους, βάση των αδειών που δίνει ο ιδιοκτήτης τους.

Αν και υποστηρίζεται ότι οι Jumping Beans μετακινούν όλο τον κώδικα πρακτόρων και τα στοιχεία, δεν είναι σαφές από την τεκμηρίωσή τους εάν κινούν πραγματικά το thread state όπως οι D'Agents. Φαίνεται ότι εφαρμόζει την πιο αδύνατη μορφή της κινητικότητας πρακτόρων.

Voyager. Το Voyager [OBJ97] (ObjectSpace, US) είναι ένα βασισμένο στη Java περιβάλλον κινητών πρακτόρων, που ενσωματώνεται στη CORBA. Παρέχει έναν κατάλληλο τρόπο να αλληλεπιδράσει, με κάποια διαφάνεια, με τα απομακρυσμένα αντικείμενα (μέσω των αντικειμένων των διαμεσολαβητών αποκαλούμενων εικονικές αναφορές) και τα αντικείμενα για να κινηθούν από host σε host. Όταν ένα αντικείμενο κινείται, αφήνει πίσω του ένα αντικείμενο αποστολών που επαναπροσανατολίζει οποιοδήποτε μήνυμα στη νέα θέση. Τα αντικείμενα των πρακτόρων, μπορεί να κινηθούν αυτόνομα εφαρμόζοντας τη μεθόδου `moveTo ()` στον εαυτό τους. Ο Voyager κινεί τον κώδικα και τα στοιχεία του πράκτορα, αλλά όχι το thread state, προς τη νέα θέση και καλεί την επιθυμητή μέθοδο εκεί. Ο Voyager επιτρέπει επίσης στα αντικείμενα και τους πράκτορες να γίνουν επίμονοι μέσω ρητών κλήσεων `saveNow ()`, υποστηρίζει την πολλαπλή μετάδοση (multicast). Ο Voyager παρέχει τους ίδιους βασικούς μηχανισμούς ασφάλειας με τα άλλα βασισμένα στη Java συστήματα.

3.3 Άλλα συστήματα

Υπάρχουν λίγα μόνο συστήματα που αναπτύσσονται σε άλλη γλώσσα προγραμματισμού πλην της Java.

Messengers. Ο Messenger [TDM+94,DMTH95,Muh98] χρησιμοποιεί κινητό κώδικα για να χτίσει ευέλικτα κατανεμημένα συστήματα, όχι απαραίτητα συστήματα κινητών πρακτόρων. Στο σύστημα αυτό, οι υπολογιστές χρησιμοποιούν το Messenger Operating System (MOS), που παρέχει μόνο λίγες υπηρεσίες. Το MOS μπορεί να στείλει και να λάβει Messengers, που είναι μικρά πακέτα δεδομένων και κώδικα γραμμένα στη δική τους γλώσσα προγραμματισμού. Το MOS μπορεί να διερμηνεύσει τα προγράμματα και έχει πρόσβαση σε μια από δύο υπηρεσίες: το γενικό λεξικό, που επιτρέπει την ανταλλαγή στοιχείων μεταξύ των Messengers και το λεξικό υπηρεσιών, το οποίο είναι μια λίστα των Messengers που προσφέρουν τις υπηρεσίες τους σε άλλους Messengers. Τελικά, τις περισσότερες υπηρεσίες, συμπεριλαμβανομένων των κατανεμημένων υπηρεσιών, προσφέρουν οι στατικοί και κινητοί Messengers. Οι Messengers μπορούν να κουβαλήσουν UNIX κώδικα, που εγκαθίσταται και εκτελείται στο MOS, ενώ οι κλήσεις του συστήματος επιτρέπουν τη γρήγορη εκτέλεση των κρίσιμων ρουτινών, διατηρώντας την ευελιξία του κινητού κώδικα.

Obliq. Το Obliq [Car95,BN97] είναι μια διερμηνευμένη, αντικειμενοστραφής γλώσσα. Ένα αντικείμενο Obliq είναι μια συλλογή ονομασμένων πεδίων που περιέχουν μεθόδους, ψευδώνυμα και τις τιμές. Ένα αντικείμενο μπορεί να δημιουργηθεί σε απομακρυσμένο σημείο, να κλωνοποιηθεί στο απομακρυσμένο σημείο ή να μεταναστεύσει με έναν συνδυασμό κλωνοποίησης και επαναπροσανατολισμού.

Η εφαρμογή των κινητών πρακτόρων είναι απλή. Ένας πράκτορας αποτελείται από μια καθορισμένη από το χρήστη διαδικασία που παίρνει έναν χαρτοφύλακα ως όρισμά της: ο χαρτοφύλακας περιέχει τα αντικείμενα Obliq που η διαδικασία απαιτεί για να εκτελέσει τη στοιχειώδη εργασία της. Ο πράκτορας μεταναστεύει στέλνοντας τη διαδικασία και το χαρτοφύλακα στη μηχανή προορισμού, η οποία καλεί τη διαδικασία για να επαναλάβει την εκτέλεση πρακτόρων. Το εικονικό Obliq [BC95] χτίζει πάνω από τις ικανότητες μετανάστευσης του Obliq. Το εικονικό Obliq είναι διαλογικός 'χτίστης' εφαρμογής που περιλαμβάνει: 1) ένα εικονικό περιβάλλον προγραμματισμού για την παρουσίαση γραφικών διασυνδέσεων με τον χρήστη και 2) έναν εξυπηρετητή πρακτόρων που επιτρέπει στις εικονικές εφαρμογές Obliq να μετακινηθούν από μηχανή σε μηχανή. Όταν η εφαρμογή μετακινηθεί, η γραφική διασύνδεση συλλαμβάνεται αυτόματα και αναδημιουργείται με ακρίβεια στη νέα μηχανή. Το Obliq δεν αντιμετωπίζει τα ζητήματα ασφάλειας. Το εικονικό Obliq παρέχει την πρόσβαση ελέγχου, ελέγχονται οι εντολές Obliq αλλά δεν υπάρχουν μηχανισμοί αυθεντικότητας. Απλώς ο χρήστης ερωτάται αν θα επιτραπεί στον πράκτορα να εκτελέσει την λειτουργία του.

Telescript. Το Telescript [Whi94b,Whi94a,Whi97] (General Magic Inc., US) ήταν το πρώτο εμπορικό σύστημα κινητών πρακτόρων. Στο Telescript, κάθε περιοχή δικτύου έχει έναν εξυπηρετητή που επιβλέπει μια ή περισσότερες εικονικές θέσεις. Ένας εισερχόμενος πράκτορας προσδιορίζει τη θέση άφιξης. Η θέση επικυρώνει την ταυτότητα του ιδιοκτήτη του πράκτορα με την εξέταση των κρυπτογραφικών πιστοποιητικών του και αναθέτει έπειτα ένα σύνολο δικαιωμάτων πρόσβασης ή αδειών στον πράκτορα. Μια άδεια, παραδείγματος χάριν, μπορεί να καθορίσει μέγιστη διάρκεια ζωής του πράκτορα, ενώ άλλη να προσδιορίσει μέγιστο ποσό χρήσης των δίσκων. Ένας πράκτορας που προσπαθεί να παραβιάσει τις άδειές του τερματίζεται αμέσως. Εκτός από τη διατήρηση των θέσεων και την επιβολή των περιορισμών ασφαλείας, ο κεντρικός υπολογιστής καταγράφει συνεχώς την εσωτερική κατάσταση της εκτέλεσης των πρακτόρων, έτσι ώστε οι πράκτορες να μπορούν να αποκατασταθούν μετά από μια αποτυχία στους κόμβους.

Ένας Telescript πράκτορας γράφεται σε μια αντικειμενοστραφής γλώσσα, παρόμοια με την Java και με τη C++ και μεταγλωττίζεται σε bytcodes. Όπως και στους D'Agents, ένας Telescript πράκτορας μεταναστεύει με συγκεκριμένες οδηγίες, οι οποίες περιλαμβάνουν τον κώδικα του πράκτορα, τα στοιχεία και το thread state. Στη νέα μηχανή, ο πράκτορας συνεχίζει την εκτέλεση από το αμέσως ακριβώς στάδιο. Ένας Telescript πράκτορας μπορεί να επικοινωνήσει με άλλους πράκτορες με δύο τρόπους: 1) μπορεί να συναντηθεί με έναν πράκτορα που είναι στην ίδια θέση (καλούν αμφότεροι τις μεθόδους τους) και 2) μπορεί να συνδεθεί με ένα αντικείμενο σε ένα διαφορετικό μέρος (οι δύο πράκτορες ανταλλάσσουν αντικείμενα κατά μήκος της σύνδεσης).

Παρά το γεγονός ότι το Telescript παραμένει ένα από τα ασφαλέστερα, ανθεκτικά στις βλάβες και ικανά συστήματα κινητών πρακτόρων, έχει αποσυρθεί από την αγορά, κατά κύριο λόγο της γρήγορης διάδοσης της Java. Το Telescript έχει στην ιδιοκτησία του το Odyssey.

4. Ομοιότητες και διαφορές

Όλα τα συστήματα κινητών πρακτόρων έχουν την ίδια γενική αρχιτεκτονική: ένας εξυπηρετητής αποδέχεται τους εισερχόμενους πράκτορες, προετοιμάζει ένα κατάλληλο περιβάλλον εκτέλεσης και φορτώνει τις πληροφορίες του πράκτορα στο περιβάλλον. Μερικά συστήματα, όπως τα βασισμένα στη Java, έχουν multi-threaded εξυπηρετητές και τρέχουν κάθε πράκτορα σε ένα thread, άλλα συστήματα έχουν εξυπηρετητές με πολλαπλή επεξεργασία και τρέχουν κάθε πράκτορα σε ξεχωριστή επεξεργασία διεργασίας και τα υπόλοιπα συστήματα χρησιμοποιούν συνδυασμό των δύο παραπάνω. Οι D'Agents, για παράδειγμα, έχουν multi-threaded εξυπηρετητή για να αυξήσουν την αποτελεσματικότητα, αλλά χρησιμοποιούνται ξεχωριστοί διεργασίες για να απλοποιηθεί η εφαρμογή. Οι Jumping Beans αποτελούν ιδιαίτερη περίπτωση αφού χρησιμοποιούν μια συγκεντρωμένη αρχιτεκτονική (στην οποία οι πράκτορες πρέπει να περάσουν μέσα από κεντρικό εξυπηρετητή για να πάνε από μια μηχανή σε άλλη) και δεν χρησιμοποιούν peer-to-peer αρχιτεκτονική (στην οποία οι πράκτορες κινούνται άμεσα από μια μηχανή προς άλλη). Αν και αυτός ο συγκεντρωμένος εξυπηρετητής μπορεί εύκολα να προκαλέσει δυσχέρεια απόδοσης, απλοποιεί πολύ την ασφάλεια, τον εντοπισμό, τη διοίκηση και άλλα ζητήματα, ίσως αυξάνοντας την αρχική αποδοχή αγοράς.

Οι Aglets είναι αυτήν την περίοδο ένα πειραματικό σύστημα. Παρέχει έναν απλό τρόπο εφαρμογής των πρακτόρων και έχει μια αυξανόμενη βάση χρηστών. Μερικά σημαντικά ζητήματα, όπως η ασφάλεια, δεν έχουν αντιμετωπιστεί λεπτομερώς. Δημιουργήθηκαν στο περιβάλλον της Ταϊτή, επικοινωνώντας μέσω αρχών με τους Aglets Workbench. Οι τελευταίοι, δεν υποστηρίζουν πλήρη διαφάνεια και συνήθως αυτό πραγματοποιείται από τον υπεύθυνο για την ανάπτυξη εφαρμογής. Αποτελεσματική η αποστολή μηνυμάτων αλλά δύσκολη επέκταση.

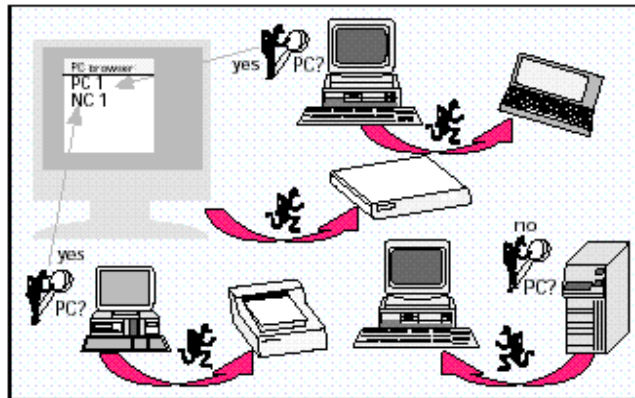
Στο Odyssey, τα αντικείμενα δρουν σαν host υπηρεσίες για τους πράκτορες. Εξετάζουν επίσης την πρόσβαση στους πόρους, όπως βάσεις δεδομένων σε host μηχανήμα. Η επικοινωνία μεταξύ των αντικειμένων στηρίζεται στη Java RMI. Καλή ασφάλεια, ευκολία στην επέκταση, αλλά δεν υπάρχει υπηρεσία καταλόγου, δεν υπάρχει συγχρονισμός στην αποστολή μηνυμάτων, φτωχή τεκμηρίωση.

Ο Voyager δεν παρέχει ασφάλεια διότι η υπηρεσία ασφαλείας του περιορίζει τη λειτουργία των ξένων αντικειμένων που φορτώνονται σε ένα δίκτυο. Το σύστημα είναι ελαφρώς εξελικτικό, καλή αποστολή μηνυμάτων, όχι υπηρεσίες καταλόγου, πολύ καλή τεκμηρίωση. Επιπλέον, οι διάρκειες ζωής των πρακτόρων μπορούν να ελεγχθούν, επιτρέποντάς τους την αυτοκαταστροφή μετά από μια παρεχόμενη περίοδο.

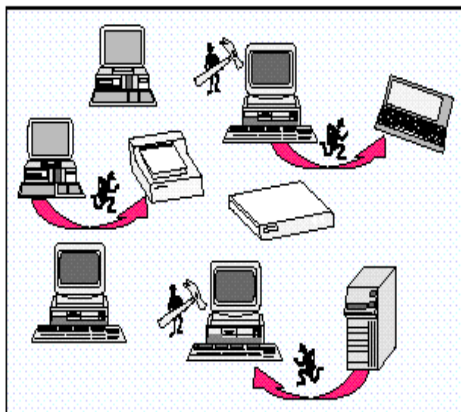
Το Ara προσφέρει ισχυρή λειτουργία κινητικότητας και στοχεύει να εφαρμόσει την κινητικότητα πρακτόρων στα συστήματα weak connection/high volumes, όπως οι περιοδικά συνδεδεμένοι υπολογιστές ή οι κατανεμημένες βάσεις δεδομένων.

Υπάρχουν κι άλλα συστήματα όπως το Mole, που χρησιμοποιεί παρόμοιες τεχνικές με τους Aglets, παρέχει ασφάλεια και υπηρεσία καταλόγου αρχείων. Το Mole υποστηρίζει τους πράκτορες των χρηστών που επικοινωνούν μεταξύ τους και είναι ένα νέο σχετικά σύστημα αφού υποστηρίζει μια υπηρεσία γεγονός που μπορεί να ειδοποιήσει τους άλλους πράκτορες. Τέλος, το σύστημα Kafka (Fujitsu, Ιαπωνία) είναι ξεχωριστό γιατί επιτρέπει επέκταση στις υπηρεσίες host μέσω της χρήσης της αντανάκλασης. Αυτό επιτρέπει στο σύστημα να είναι προσαρμοστικό, ανάλογα με το περιεχόμενο. Παρέχει τις ελάχιστες λειτουργίες και προωθεί την καθορισμένη από τον χρήστη πολιτική, π.χ. οι χρήστες μπορούν να εφαρμόσουν τις πολιτικές ασφαλείας τους.

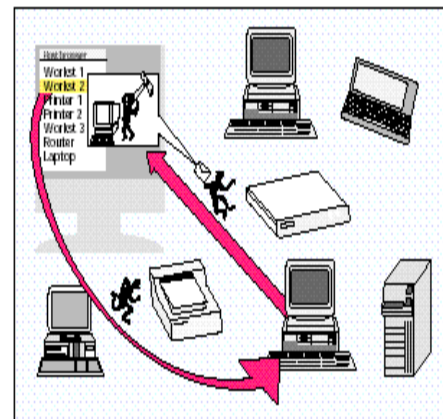
Τα συστήματα κινητών πρακτόρων παρέχουν γενικά το ένα από δύο είδη μετανάστευσης: 1) ισχυρή κινητικότητα (strong mobility), όπου το σύστημα συλλαμβάνει την κατάσταση αντικειμένου ενός πράκτορα, τον κώδικα και το thread state, επιτρέποντας στον πράκτορα να συνεχίσει την εκτέλεση από το ακριβές σημείο στο οποίο βρίσκεται, 2) αδύνατη κινητικότητα (weak mobility), όπου το σύστημα συλλαμβάνει μόνο την κατάσταση και τον κώδικα αντικειμένου του πράκτορα και καλεί ένα γνωστό σημείο εισόδου μέσα στον κώδικα για να επανεκκινήσει τον πράκτορα στη νέα μηχανή. Το μοντέλο ισχυρής κινητικότητας είναι καταλληλότερο για τον τελικό προγραμματιστή, αλλά απαιτείται περισσότερη εργασία από τον υπεύθυνο για την ανάπτυξη συστημάτων δεδομένου ότι οι ρουτίνες για να συλλάβουν την κατάσταση συμπεριφοράς πρέπει να προστεθούν στους υπάρχοντες διερμηνευτές. Όλα τα εμπορικά βασισμένα στη Java συστήματα χρησιμοποιούν την αδύνατη κινητικότητα, δεδομένου ότι η αγορά απαιτεί τα συστήματα να τρέχουν πάνω από τις χωρίς τροποποιήσεις JVMs και οι παρούσες JVMs δεν υποστηρίζουν τη σύλληψη του thread state. Τα ερευνητικά συστήματα χρησιμοποιούν και τις δύο τεχνικές μετακίνησης.



- Discovery agents -



- Maintenance agents -



- Browsing applets -

Αυτήν την περίοδο, για λόγους φορητότητας και ασφάλειας, σχεδόν όλα τα συστήματα κινητών πρακτόρων είτε διερμηνεύουν τις γλώσσες τους άμεσα, είτε μεταγλωττίζουν τις γλώσσες τους σε bytcodes και διερμηνεύουν έπειτα τα bytcodes. Η Java, είναι η δημοφιλέστερη γλώσσα πρακτόρων, δεδομένου ότι: 1) είναι φορητή αλλά ικανή, 2) οι υπάρχοντες μηχανισμοί ασφάλειάς της επιτρέπουν την ασφαλή εκτέλεση του επικίνδυνου κώδικα και 3) υπάρχει διαδεδομένη διείσδυση στην αγορά. Η Java χρησιμοποιείται σε όλα τα εμπορικά συστήματα και σε διάφορα ερευνητικά συστήματα. Οι ερευνητές πειραματίζονται στην 'on-the-fly' μεταγλώττιση. Τα πιο πρόσφατα JVMs χρησιμοποιούν αυτή τη μεταγλώττιση και μόνο τα συστήματα κινητών πρακτόρων βασισμένα στη Java μπορούν να αποκομίσουν τα οφέλη.

Τα υπάρχοντα συστήματα κινητών πρακτόρων εστιάζουν στην προστασία μιας μεμονωμένης μηχανής από τους κακόβουλους πράκτορες. Ο ιδιοκτήτης ενός πράκτορα ή το μηχάνημα που αποστέλλει υπογράφει ψηφιακά τον πράκτορα ή λαμβάνουσα μηχανή ελέγχει την υπογραφή, αποδέχεται ή απορρίπτει τον πράκτορα, αναθέτει τους περιορισμούς πρόσβασης και μετά εκτελεί τον πράκτορα σε ασφαλές περιβάλλον. Τα περισσότερα υπάρχοντα συστήματα δεν παρέχουν οποιαδήποτε προστασία για τον

πράκτορα ή για μια ομάδα μηχανών που δεν είναι υπό τον ενιαίο έλεγχο του διαχειριστή.

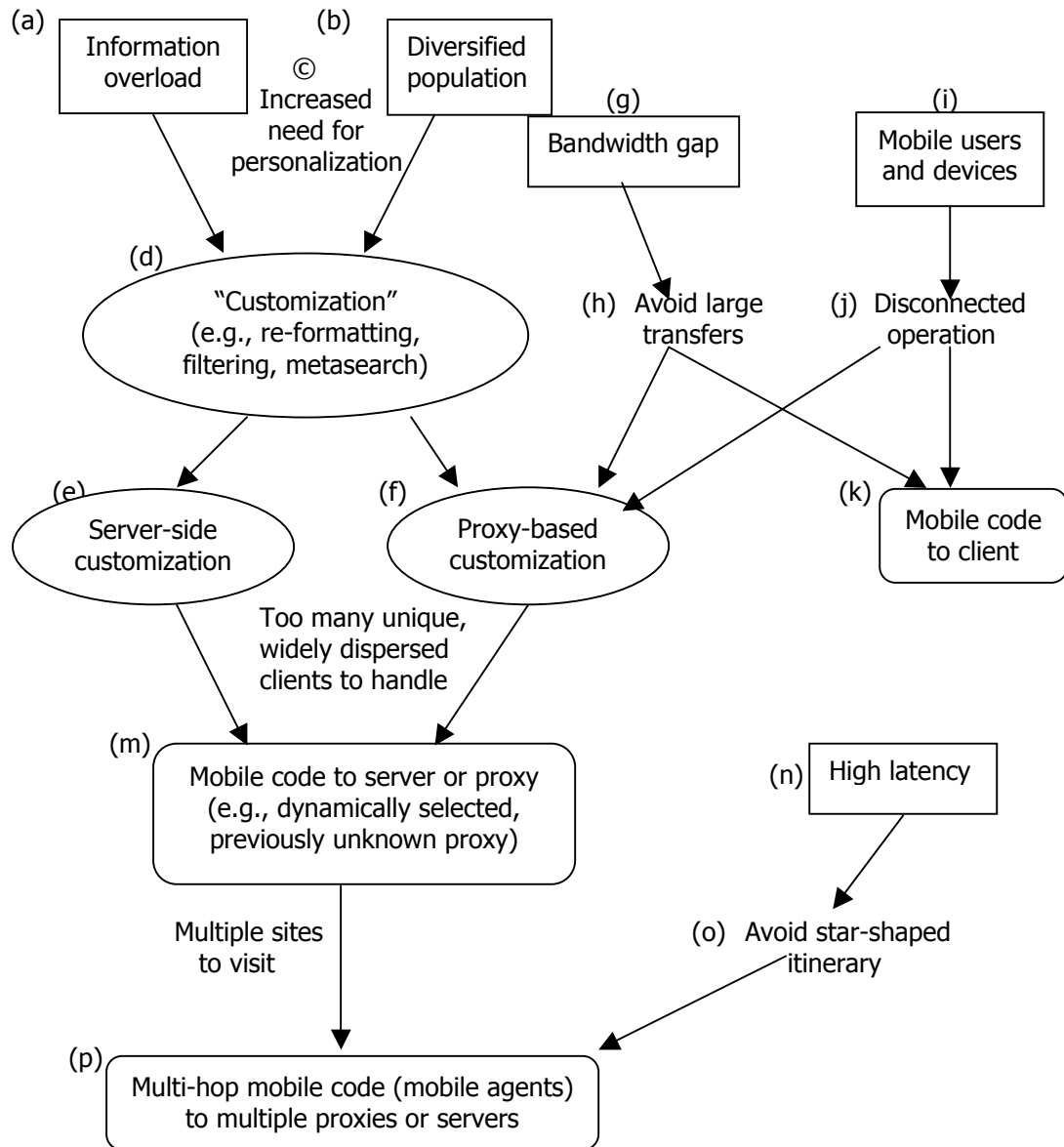
Άλλες διαφορές που υπάρχουν μεταξύ των συστημάτων κινητών πρακτόρων είναι η κατατεμάχιση (*granularity*) των μηχανισμών επικοινωνίας, είτε όταν χτίζονται στην κορυφή είτε αλληλεπιδρώντας με CORBA είτε προσαρμόζονται στα αναδυόμενα πρότυπα κινητών πρακτόρων όπως MASIF [MBB+99] και FIPA. Παρά τις διαφορές, όλα τα συστήματα (με εξαίρεση τους Messengers, που είναι ένα σύστημα *lighter-weight*), προορίζονται για τις ίδιες εφαρμογές, όπως διαχείριση δικτύων και αυτοματοποιημένη εγκατάσταση λογισμικού.

Όλα τα συστήματα είναι κατάλληλα για τη κατανεμημένη ανάκτηση πληροφοριών και η απόφαση για το ποιο σύστημα θα χρησιμοποιήσουμε πρέπει να βασιστεί στην επιθυμητή γλώσσα εφαρμογής, το αναγκαίο επίπεδο ασφάλειας και την αναγκαία απόδοση. Τα εμπορικά συστήματα, παραδείγματος χάριν, έχουν υψηλότερη απόδοση από τα περισσότερα ερευνητικά συστήματα, απλά λόγω των περισσότερων ανθρωποωρών των προγραμματιστών και βέβαια τη σημασία της απόδοσης σε μια εμπορική προσφορά.

5. Συμπεράσματα

Οι κινητοί πράκτορες έχουν έξι δυνάμεις: συντήρηση του εύρους ζώνης, μείωση στις λανθάνουσες καταστάσεις, μείωση του συνολικού χρόνου ολοκλήρωσης, υποστήριξη στις προβληματικές λειτουργίες (αποσύνδεση), υποστήριξη για την εξισορρόπηση φόρτωσης και υποστήριξη για τη δυναμική επέκταση. Αν και καμία από αυτές τις έξι δυνάμεις δεν είναι μοναδική στους κινητούς πράκτορες, καμία ανταγωνιστική τεχνική δεν μοιράζεται και τις έξι. Η αληθινή δύναμη των κινητών πρακτόρων είναι για τις κατανεμημένες εφαρμογές. Ικανοποιητικό αποτέλεσμα υπάρχει όταν τα μηχανήματα πελατών και εξυπηρετητών συνδέονται με ένα μέσης ταχύτητας ή γρηγορότερο δίκτυο (όπως 10 Mb/s Ethernet). Επιπλέον, τα παρόντα συστήματα δεν προστατεύουν επαρκώς έναν πράκτορα από τις κακόβουλες μηχανές και αυτό περιορίζει τις επιλογές μετακίνησης ενός πράκτορα. Μερικοί πράκτορες θα πρέπει να ενεργήσουν όπως οι παραδοσιακοί πελάτες και να έχουν πρόσβαση σε μια υπηρεσία μέσω δικτύου, παρά να μετακινηθούν σε νέα θέση και να είναι ανοιχτοί στις επιθέσεις.

Υπάρχουν διάφορες τάσεις που επηρεάζουν τη τεχνολογία και δραστηριότητα Internet: το εύρος ζώνης, οι mobile μηχανισμοί, οι mobile χρήστες, τα Intranets, τα μεγάλα ποσά πληροφοριών, η προσαρμογή των χρηστών, οι διαμεσολαβητές. Οι τάσεις φαίνεται ότι οδηγούν στους mobile agents.



- Οι τάσεις οδηγούν στους mobile agents -

Οι ερευνητές μελετούν διάφορες ελπιδοφόρες λύσεις σε αυτά τα δύο προβλήματα της απόδοσης και της ασφάλειας, εντούτοις, και τώρα μοιράζονται έναν σαφή και εφικτό στόχο: ένα σύστημα κινητών πρακτόρων που: (1) επαρκώς προστατεύει έναν πράκτορα από τις κακόβουλες μηχανές, (2) υποστηρίζει τη μετακίνηση πρακτόρων και (3) επιτρέπει στους πράκτορες την εκτέλεση σχεδόν τόσο γρήγορα σαν να μεταγλωττίστηκαν (άμεσα) σε κώδικα μηχανής. Οι ποιοτικές δυνάμεις των κινητών πρακτόρων, όπως η υποστήριξή τους για την αποσυνδεδεμένη λειτουργία και τη δυναμική επέκταση, κάνουν τα παρόντα συστήματα μια ελκυστική επιλογή για μερικές εφαρμογές ακόμα κι αν οι συνολικοί χρόνοι ολοκλήρωσης είναι οι χειρότεροι. Δεδομένου ότι τα ζητήματα απόδοσης και ασφάλειας αντιμετωπίζονται, τα συστήματα κινητών πρακτόρων θα γίνουν ελκυστικά για ένα όλο και περισσότερο ευρύ φάσμα των καταναμημένων εφαρμογών κι από ότι φαίνεται η Ευρώπη έχει το προβάδισμα.

6. Αναφορές

Βιβλιογραφία

- [1]. [Baumann97] J. Baumann, F. Hohl, K. Rothermel, and M. Straber : « Mole-Concepts of a Mobile Agent System », Universtat Suggart, Fakultat Informatik, Bericht 1997/15, August 1997
- [2]. [Chess94] D. Chess, C. Harrison et A. Kershenbaum, "Mobile Agents: are they a good idea ?", IBM Research Report, RC 19887, December 1994.

www διευθύνσεις

- [1]. <http://csrc.nist.gov/staff/jansen/pp-attCert-final.pdf>
Μελέτη των Wayne Jansen, Tom Karygiannis (National Institute of Standards and Technology) με τίτλο «Privilege Management of Mobile Agents» που ασχολείται από γενική σκοπιά με τη διαχείριση των αρχών πρόσβασης βάση των δικαιωμάτων και αδειών σύμφωνα με τα χαρακτηριστικά τους (passport), που τους επιτρέπουν να αποκτήσουν μεγαλύτερη ευελιξία και να πραγματοποιήσουν τους σκοπούς τους.
- [2]. <http://citeseer.nj.nec.com/cache/papers/cs/20004/http:zSzzSzdsp.jpl.nasa.govzSzmemberszSzpaymanzSzswarmzSzbieszczad98-comsoc.pdf/bieszczad98mobile.pdf>
Εργασία του Andrzej Bieszczad, (BELL Laboratories), Bernard Pagurek & Tony White, (Carleton University) με τίτλο «Mobile agents for network management», όπου μελετούνται οι χρήσεις των mobile agents στη διαχείριση των δικτύων, ορίζονται οι πράκτορες λογισμικού και ένα μοντέλο κινητικότητας. Χρησιμοποιήθηκαν διάφορες εικόνες.
- [3]. <http://agent.cs.dartmouth.edu/workshop/1997/slides/lange/sld001.htm>
Slides του Danny Lange του Department of Computer Science, Dartmouth College, με θέμα «Is Java the "BASIC" of Mobile Agents? Is Tcl? Is Scheme?...»
- [4]. <http://www.msci.memphis.edu/~franklin/AgentProg.html>
Άρθρο των Stan Franklin and Art Graesser, Institute for Intelligent Systems University of Memphis, με τίτλο «Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents» όπου προτείνουν έναν ορισμό και καθορίζουν τον αυτόνομο πράκτορα και τον διαχωρίζουν από οποιοδήποτε πρόγραμμα. Ταξινόμηση των αυτόνομων πρακτόρων και μελέτη για περαιτέρω διαχωρισμό.
- [5]. <http://citeseer.nj.nec.com/rd/97364000,295489,1,0,Download/http%3A%2F%2Fwww-ra.informatik.uni-tuebingen.de/mitarb/schwehm/MYpapers/RotSch98.ps.gz>
Μελέτη των K.Rothermel και M.Schwehm για βασικά θέματα των mobile agents, όπως κινητικότητα, τύποι, τοποθεσίες και επικοινωνία. Παραθέτονται περιληπτικά τα πλεονεκτήματα από τη χρήση των mobile agents, οι γλώσσες προγραμματισμού και τα υπάρχοντα συστήματα. Τέλος, εκθέτονται οι προοπτικές από τις έρευνες και τη χρήση των mobile agents.
- [6]. <http://agents.umbc.edu/mobile/>
Σελίδα του UMBC AgentWeb. Παραπέμπει σε εξειδικευμένο site για τους mobile agents. Λίστες και μελέτες για agents, εταιρίες, λογισμικό, συστήματα, εφαρμογές, μαθήματα, σεμινάρια, δημοσιεύσεις, άρθρα. Συστήματα Ara, Aglets και πολλά άλλα.
- [7]. <http://www.dyade.fr/actualites/c11evaluation.pdf>
Paper των L.Ismail, D.Hagimont, J. Mossiere από διάφορα ινστιτούτα με θέμα «Evaluation of the Mobile Agents Technology and Comparison Studies», όπου παρουσιάζεται συγκριτική μελέτη της τεχνολογίας των mobile agents βάση επικοινωνιακών κοστών.
- [8]. <http://www.infosys.tuwien.ac.at/Research/Agents/special.html>
Σελίδα που συγκρίνει τα συστήματα: ObjectSpace Voyager, General Magic Odyssey, IBM Aglets: A Comparison by ObjectSpace (June 1997), βάση του τρόπου αποστολής μηνυμάτων, διάρκεια ζωής, υπηρεσίες, κινητικότητα, συνδέσεις, ασφάλεια, τιμολόγηση, μέγεθος και παρουσίαση.

- [9]. <http://link.springer.de/link/service/series/0558/papers/1419/14190137.pdf>
PDF του Giovanni Vigna (Dip. Elettronica e Informazione, Politecnico di Milano) με τίτλο «Cryptographic Traces for Mobile Agents» που μελετάει την αυθεντικότητα των κινούμενων μελών του mobile agent και την προστασία των hosts από τους επικίνδυνους κώδικες. Παρουσιάζεται μια διαδικασία ιχνηλάτησης και κρυπτογραφίας που βοηθάει στην ασφαλή εκτέλεση και μετακίνηση των τμημάτων λογισμικού.
- [10]. <http://www.cs.dartmouth.edu/~dfk/papers/kotz:future/>
Άρθρο των David Kotz and Bob Gray από τα Department of Computer Science / Thayer School of Engineering Dartmouth College με τίτλο «Mobile Agents and the Future of the Internet», που αποδεικνύουν ότι σύμφωνα με τις σημερινές τεχνολογικές τάσεις τα συστήματα για τις κατανεμημένες εφαρμογές θα κάνουν χρήση των mobile codes και των mobile agents. Μελετώνται διάφορα τεχνικά και μη εμπόδια για την πορεία αυτή.
- [11]. http://www.argreenhouse.com/papers/fanjum/buffalo_for.pdf
Paper των Ravi Jain, Farooq Anjum and Amjad Umar (Telcordia Technologies, Inc. (formerly Bellcore) Morristown, NJ.) με τίτλο «A Comparison of Mobile Agent and Client-Server paradigms for information retrieval tasks in virtual enterprises» για τη νέα γενιά επιχειρήσεων και χρηστών που απαιτούν την ανάκτηση μεγάλων ποσών δεδομένων από απομακρυσμένα μέρη. Η μελέτη δείχνει ότι οι mobile agents δεν είναι πάντα καλύτεροι από τις κλήσεις client-server όσον αφορά τον μέσο χρόνο απόκρισης.
- [12]. <http://www.newcastle.research.ec.org/cabernet/sota/report/node50.html>
Σελίδα CaberNet, Information Society Technologies (IST Programme) Network of Excellence in Distributed and Dependable Computing Systems, όπου παρουσιάζονται διάφορα συστήματα όπως Aglets, Odyssey, Telescript, Kafka, Voyager και υπάρχουν links για το καθένα.
- [13]. www.ansa.co.uk/ANSATech/97/Present/202301.pdf
Μελέτη του Mike Bursell (June 1997) με θέμα «Mobile Agent Comparison». Συγκρίνει τα συστήματα CHAOS, Odyssey, Voyager και Aglets βάση της ασφάλειας, υποστήριξη ORB, διαθεσιμότητας, χρηστών, αποστολής μηνυμάτων.
- [14]. <http://www.cigital.com/conferences/ecbs02/AgentWS.html>
Ποσοτική και ποιοτική σύγκριση των mobile agents systems από το Lund University, Lund, SWEDEN.