Національний університет

***“Києво-Могилянська Академія”***

Департамент комп’ютерних технологій

Кафедра інформатики

**Мобільні об’єкти та мобільні агенти:**

 **майбутнє розподілених обчислень?**

Реферат з курсу “Програмне забезпечення інтелектуальних систем”

виконав студент V курсу

**Кравченко Іван Олександрович**

Київ, 2000 р.

Зміст

1. Вступ 3

2. Що таке агент як програмний продукт ? 3

3. Що таке мобільний агент? 5

4. Підстави для використання мобільний агентів 6

5. Парадигми обчислень в мережі 9

6. Мобільні агенти в якості застосувань 10

7. Сучасні системи мобільних агентів. 13

8. Стандартизація мобільних агентів: MASIF 15

9. Підсумок 18

10. Література і ресурси 19

1. Вступ

 Ця робота познайомить вас із світом мобільних агентів - новітньої технології, яка дозволяє швидко розроблювати і підтримувати розподілені системи. Ви дізнаєтесь, що мобільні агенти зменшують мережевий потік обміну, забезпечують ефективні засоби подолання мережевої латентності, і, можливо, найважливіше, що через їхню здатність діяти асінхронно і автономно від процесу, який створив їх, є стійкими до аварійних ситуацій. Разом ми розглянемо агентів – як мобільних, так і статичних, проаналізуємо всі переваги мобільних агентів і продемонструємо той вплив, який вони здійснюють на проектування розподілених систем, а також зробимо стислий перегляд деяких сучасних систем мобільних агентів.

2. Що таке агент як програмний продукт ?

Так що таке програма - агент, з чого він складається, і чим відрізняється від звичайної програми, - протягом декількох останніх років ці питання були і є темою для обговорення і суперечок. У той час, поки ці дебати тривають, на практиці дедалі більше зустрічається програмних продуктів, для яких існує таке, суто емпіричне визначення: “Агенти – це програми, які допомагають людям і працюють в їх інтересах”. Це визначення більше схоже, власне, на мету, задля якої створюються агентні системи. Зазначимо для справедливості, що всі програмні продукти мають на меті в той чи інший спосіб допомагати кінцевим користувачам.

Визначення агента ( з точки зору кінцевого користувача )

 **Агент** - *це програма, яка допомагає людям і діє в їх інтересах. Агенти функціонують в рамках дозволу, який делегується користувачем.*

Незважаючи на те, що таке визначення по суті є правильним, в ньому ніяким чином не зазначається механізм роботи агентів. Агенти можуть бути різних типів і працювати в різний спосіб. Їх можна знайти в комп'ютерних операційних системах, мережах, базах даних тощо. Які риси програми дозволяють називати її агентом?

 Ми не будемо досліджувати характеристики численних агентних систем, які доступні для використання на сучасному ринку інформаційних технологій. Але, якщо ви розглянете всі такі системи, то знайдете принаймні одну властивість, яка є спільною для агентів, а саме той факт, що вони функціонують в деякому специфічному середовищі. Агенти здатні взаємодіяти з власним програмним середовищем, а також діяти асінхронно і автономно. Ні від кого не вимагається доставляти інформацію до агента або споживати його вихідні дані. Агент просто безперервно діє, переслідуючи власну мету.

На відміну від об’єктно-орієнтованих програмних продуктів, до яких треба безпосередньо звертатися через певний інтерфейс для того, щоб отримати результат, агенти є активними об'єктами, які функціонують відповідно до так званого принципу Голлівуда: “Не кличте нас, ми самі вас покличемо!”

Визначення Агента ( з точки зору системи )

**Агент** – *це програмний продукт, який:*

* розміщується всередині середовища виконання;
* має такі обов’язкові властивості:

а) здатність фіксувати зміни в середовищі і діяти відповідно до них;

б) здатність контролювати власні дії;

в) здатність самостійно активізуватися для досягнення власної мети;

г) здатність працювати впродовж деякого (невизначеного) терміну;

* крім того, може мати такі необов’язкові властивості:

а) здатність до обміну повідомленнями з іншими агентами;

б) здатність до пересування з одного серверу (комп’ютера) до іншого;

в) здатність до адаптації з урахуванням попереднього досвіду (здатність до навчання);

 г) здатність проявляти/ приховувати своє існування для користувача.

3. Що таке мобільний агент?

Мобільність є необов’язковою властивістю агентів, тобто, не всі агенти повинні бути спроможними до пересування в комп’ютерній мережі. Довільний агент може функціонувати на одному сервері і співпрацювати з середовищем за встановленими правилами, в тому числі шляхом віддаленого виклику процедур і обміну повідомленнями. Назвемо агентів, які не можуть або не “хочуть” пересуватися *стаціонарними* (статичними).

 Визначення стаціонарного (статичного) агента.

**Стаціонарний агент** – *агент, який функціонує тільки в тій системі, де почалося його виконання.*

 Якщо такому агенту потрібна інформація, відсутня у власній системі, або необхідно взаємодіяти з агентом в іншій системі, то він, звичайно, використовує будь-який механізм зв’язку, наприклад, віддалений виклик процедури (RPC).

На відміну від такого визначення, мобільний агент не прив'язується до системи, де почалось його виконання. Мобільний агент може вільно мандрувати між комп'ютерами в мережі. Створений в одному середовищі виконання, він може перемістити свій код і стан до іншого середовища виконання в мережі, де й продовжить свою роботу.

Під терміном "*стан*" (“state”) розуміють сукупність значень атрибутів агента, яка допомагає йому визначити наступні кроки при завершенні роботи в певному середовищі. Під терміном “*код*” розуміють, в контексті об’єктно-орієнтованого програмування, визначений клас коду, необхідний для виконання агента.

Визначення мобільного агента

**Мобільний агент** – *агент, не прив'язаний до системи, де розпочалось його виконання. Він має здатність переміщувати себе від однієї системи в мережі до іншої.*

 Здатність рухатись дозволяє мобільному агенту пересуватися до тієї системи, яка містить об'єкт, з яким цей агент хоче взаємодіяти, а також, працювати на тому ж комп'ютері або в тій же мережі, де знаходиться цей об'єкт.

4. Підстави для використання мобільний агентів

Хоча технологія мобільних агентів викликає нечуване захоплення в сучасному світі, інтерес до них мотивується не технологією по суті, а скоріше перевагами, які з’являються при створенні розподілених систем на основі такої технології. Нижче наведено сім добрих причин для того, щоб використовувати мобільні агенти.

**Вони зменшують завантаження в мережі**. Розподілені системи часто залежать від протоколів зв'язку, які передбачають багатократну взаємодію для опрацювання одного завдання. Це особливо стосується тих систем, в яких велику увагу приділяється конфіденційності і безпеці роботи. Результатом є різке зростання мережевого трафіку. Мобільні агенти дозволяють пакувати можливий діалог (при віддаленому виклику процедури) і послати його до комп'ютера адресата, де взаємодія відбудеться локально. Мобільні агенти також корисні для зменшення трафіку при передачі великих обсягів даних в мережі. Дійсно, коли великі томи даних зберігаються в окремих комп'ютерах, наприклад, серверах баз даних, певну їх обробку було б зручніше виконувати там же, ніж передавати їх по мережі до машини, на якій встановлено програму обробки. Принцип тут простий: *пересунути обчислення до даних скоріше, ніж дані до обчислень*.

**Вони долають мережеву латентність.** Системам реального часу, як, наприклад, роботам в процесах виробництва, потрібно реагувати на зміни в їхніх середовищах максимально швидко. Контролювання таких систем через фабричну мережу великого розміру призводить до значної латентність, яка абсолютна неприйнятна для критичних систем реального часу. Використання в такій мережі мобільних агентів є чудовим рішенням, бо вони можуть бути послані від центрального контролера для дій у певному місці і безпосередньо виконувати його інструкції.

**Вони інкапсулюють протоколи.** Коли в розподіленій системі відбувається обмін даними, кожний комп'ютер має програмне забезпечення, яке реалізує протоколи, потрібні для належного кодування вихідних даних і інтерпретації вхідних. Однак, протоколи мають властивість розвиватися, а перехід на нову, більш ефективну або захищену версію протоколу є важким, іноді цілком неможливим завданням по модернізації встановленого програмного забезпечення. Як результат, часто виникає проблема з підтримкою старих версій, або проблема несумісності версій протоколів. Мобільні агенти можуть пересуватися до віддалених комп'ютерів "каналами", які базуються на власних протоколах. До речі, перехід на нову версію протоколу в мережі можна виконати за допомогою мобільних агентів, які у визначеному порядку встановлять на кожному комп’ютері нове програмне забезпечення.

**Вони працюють асінхронно і автономно.** Часто зовнішні мобільні пристрої (ноутбуки, мобільні телефони, автомобільні комп’ютери) залежать від дорогих або просто неякісних зв'язків з мережею. З цього випливає, що завдання, які вимагають безперервного зв’язку між мобільним зовнішнім пристроєм і деякою корпоративною мережею, певно будуть неекономними або технічно складними в реалізації. При використанні агентів такі завдання можуть бути вміщені в мобільні агенти, які після цього можуть бути послані в мережу. При встановлені зв’язку з мобільним пристроєм мобільні агенти переміщуються в його систему і функціонують незалежно від створившої їх системи, працюють асінхронно і автономно. Мобільний зовнішній пристрій може повторно з'єднатися пізніше, щоб забрати агента і дізнатися про результат його дій.

**Вони адаптуються динамічно.** Мобільні агенти здатні аналізувати своє програмне середовище виконання і реагувати відповідно до змін, які в ньому відбуваються. Багато мобільних агентів здатні розподілюватися між комп'ютерами в мережі таким чином, щоб підтримати оптимальну конфігурацію для розв'язання специфічної проблеми.

**Вони природно різнорідні (гетерогенні).** Розподілені обчислення в своїй більшості різнорідні, як з точки зору апаратних засобів, так і програмного забезпечення. Оскільки мобільні агенти є, в принципі, комп'ютер- і транспорт-незалежні, і залежать тільки від їх середовища виконання, то вони створюють оптимальні умови для безшовного інтегрування розподілених систем.

**Вони стійкі до несправностей в мережі**. Здатність мобільних агентів динамічно реагувати на несприятливі ситуацій і події робить легким побудову стійких до критичних ситуацій розподілених систем. Якщо комп'ютер припиняється свою роботу, то виконання агентів на цій машині зупиниться, і їм буде наданий час, щоб переміститися і продовжити роботу на іншому комп'ютері в мережі.

5. Парадигми обчислень в мережі

Досвід переконує, що мобільні агенти надають могутню однорідну парадигму для розподілених обчислень. Мобільні агенти можуть стати новим кроком в розробці розподілених систем. Зробимо короткий перегляд і порівняння трьох парадигм для реалізації розподілених обчислень: *клієнт-сервер, код-на-запит (Code-on-Demand) і рухливих агентів*. Зверніть увагу, що ми зробимо наголос на тому, як парадигма сприймається розробником, а не на архітектурі програмного забезпечення та обладнання.

 **Парадигма клієнт-сервер.** В парадигмі клієнт-сервер сервер пропонує множину сервісів, які забезпечують доступ до деяких ресурсів (наприклад, баз даних). Код програми, яка відповідає за цей сервіс, знаходиться локально на сервері. Ми кажемо, що сервер має know-how. Врешті-решт, сервер безпосередньо виконує програму доступу до ресурсу, і в такий спосіб використовує процесор. Якщо клієнт зацікавлений в доступі до деякого ресурсу на сервері, він просто використовуватиме один або більше з сервісів, які забезпечуються сервером. Зверніть увагу, що клієнту потрібен деякий "інтелект", щоб визначитись, яким сервісом йому треба скористатись. Сервер має know-now, ресурси, і процесор.

До теперішнього часу, більшість розподілених систем базувалися саме на цій парадигмі. На ній побудовано велику кількість технологій як, наприклад, віддалений виклик процедури (RPC), брокери об'єктного запиту (CORBA) і виклик віддаленого методу в Java (RMI).

Відповідно до парадигми *код-на-запит* (Code-on-Demand), передача коду (know-how) на комп’ютер відбувається по запиту з цієї машини. Кажуть, що один комп'ютер (А) спочатку неспроможний виконати завдання через відсутність коду (know-how). Але інший комп'ютер (B) в мережі має потрібний код (програму). Як тільки А-комп’ютер отримує код, обчислення проводиться всередині А-комп’ютера. Ця машина має в розпорядженні процесор і локальні ресурси. На відміну від парадигми клієнт-сервер, А не потребує знання про віддалений комп'ютер після того, як весь необхідний код буде завантажений. Кажемо, що один комп'ютер (А) має ресурси і процесор, а інший (B) має know-how. Аплети Java є найкращим прикладом реалізації такої парадигми. Аплети завантажуються в броузер і виконуються локально.

**Парадигма мобільного агента.** Ключовою характеристикою парадигми мобільного агента є те, що будь-якому комп'ютеру в мережі дозволяється гнучко конфігурувати множину know-how, ресурсів і процесорів, як засобів для обробки даних. Процесорні потужності можуть бути об'єднані з локальними ресурсами. Know-how (в вигляді мобільних агентів) не нав'язується конкретному комп'ютеру, але є доступним через мережу.

Якщо порівнювати ці три парадигми, можна побачити хронологічний напрям в розвитку, спрямований до більшої гнучкості. Клієнт і сервер зливаються в один комп'ютер. Аплети (клієнти) і серверлети (серверні додатки), відповідно, об'єднуються і вдосконалюються з появою парадигми мобільних агентів.

6. Мобільні агенти в якості застосувань

Розглянемо ближче деякі програмні продукти, які реалізовано в рамках парадигми мобільних агентів.

**Електронна комерція.** Мобільні агенти добре відповідають вимогам електронної комерції. Комерційна обробка запиту (транзакція) може вимагати доступу в реальному часі до віддалених ресурсів як, наприклад, запасів на складах, поточних цін на товар і, можливо, навіть детального узгодження умов операції. Різні агенти мають різну мету, і розвивають різні стратегії для їх досягнення. Уявимо собі агентів, які втілюють мету їхніх творців, представляють інтереси, ведуть переговори замість, власне, людини. Технологія мобільних агентів є зворушливим розв'язком проблеми такого типу.

**Персональна допомога.** Здатність мобільних агентів працювати на віддалених комп'ютерах робить їх придатними для використання асистентами, які будуть виконувати завдання в мережі від імені користувача. Віддалений асистент діятиме незалежно від того, чи існує в даний момент зв’язок з комп’ютером-“батьком”, і користувач може вимкнути машину після того, як агент почне роботу в мережі. Наприклад, для призначення зустрічі з іншою людиною користувач зможе послати мобільного агента для взаємодії з таким самим агентом людини, з якою треба зустрітися. Агенти зустрінуться і домовляться про час та місце зустрічі відповідно до побажань користувачів.

**Приватні (корпоративні) зустрічі.** Цікаве застосування мобільних агентів може бути в робочих групах, де не всі співробітники довіряють один одному. У такому випадку, окремі користувачі могли б призначити місце збору їх агентів на надійному комп'ютері, де взаємодія може відбуватися без ризику, що ще чиїсь агенти одержать конфіденційну інформацію.

**Розподілений пошук інформації.** Пошук інформації є одним з найзгадуваніших прикладів застосування мобільних агентів. Замість того, щоб передавати великі обсяги даних до центрального пошукового серверу, де буде створено пошуковий індекс, до віддалених джерел інформації посилаються агенти, які локально створюють пошукові індекси і повертаються з ними до місця відправлення. Мобільні агенти також можуть виконувати розширений пошук, який триває годинами, проте не вимагає роботи комп'ютера користувача.

**Телекомунікаційні послуги.** Підтримка і управління провідними телекомунікаційними сервісами характеризуються динамічною мережевою переконфігурацією. Фізичний розмір цих мереж і вимоги до їх експлуатації такі, що використання технології мобільних агентів значно підвищить гнучкість і ефективність систем. Агенти виконуватимуть роль "клею", який утримує такі системи.

**Організація групових робіт.** За своєю природою групова робота вимагає обміну інформацією між співробітниками. Мобільні агенти будуть корисними для організації автономної роботи над певними частинами проекту, зберігаючи при цьому цілісність всього проекту.

**Моніторинг і служба повідомлень.** Це одне з класичних застосувань мобільних агентів, яке висвітлює їх асинхронну природу. Агент може спостерігати за деяким джерелом інформації незалежно від комп’ютера, з якого він був надісланий. Агенти можуть бути послані, щоб чекати на появу певного виду інформації. Важливо те, що агенти контролю мають недетермінований термін дії, який перевищує або не залежить від обчислювальних процесів, що створюють таких агентів.

**Поширення інформації.** Мобільні агенти втілюють так звану модель Internet-"поштовху" (Internet push). Вони можуть розповсюджувати інформацію типу новин або нових версій програмного забезпечення для продавців та зареєстрованих користувачів. Агенти здатні доставити і встановити новий програмний компонент на комп’ютер в такий самий спосіб, як і звичайні процедури установки.

**Паралельні обчислення.** Відомо, що мобільні агенти можуть клонуватись в мережі, а тому, ще одним потенційним застосуванням мобільних агентів є управління паралельними обчисленнями. Якщо деяке обчислення вимагає великих ресурсів, то воно може бути розподілене між декількома процесорами (комп’ютерами), які входять до мережі, в якій діють специфічні агенти паралельних обчислень.

7. Сучасні системи мобільних агентів.

Які види систем мобільних агентів доступні зараз? Після того, як з’явилась Java, на її платформі було згенеровано багато експериментальних систем мобільних агентів. Численні системи розробляються зараз, і більшість з них доступна для оцінки в Інтернеті.

Кількість систем зростає так швидко, що будь-яка спроба перелічити та класифікувати їх приречена на неповноту й застарілість, тому в цьому досліджені здається доцільним згадати тільки найцікавіші базові системи мобільних агента, які реалізовано на Java: Aglets, Odyssey, Concordia і Voyager.

**Aglets.** Ця система, яка створюється General Magic Inc., є розвитком аплетів в Java. Мета проекту - надати мобільності Java-аплетам. Термін *аglet* є об'єднанням слів *agent* і *applet*. Розробники спробували створити аглети в якості експериментальної дослідницької моделі, і сподіваються, що програмісти оцінять шляхи розвитку, що пропонують аглети як нащадки аплетів.

**Odyssey.** General Magic Inc. є винахідником мобільного агента і свого часу створив першу комерційну систему мобільних агентів Telescript. Базуючись на власній мові і архітектурі мережі, Telescript мав короткий термін життя. Зважаючи на популярність Internet і приголомшуючий успіх мови Java, General Magic Inc. створила нову систему мобільних агентів Odyssey, реалізовану на Java. Ця система ефективно втілює принципи Telescript вже у вигляді Java-класів. Результатом роботи є бібліотека Java-класів, яка дає розробникам можливість створювати власні застосування мобільних агентів.

**Concordia.** Mitsubishi's Concordia є продуктом для розробки і управління системами мобільних агентів, який може функціонувати в будь-якому середовищі, яке підтримує Java. Concordia вміщує компоненти, написаних на Java, які разом надають завершене середовище для створення розподілених застосувань. Система Concordia, в найпростішому вигляді, складається з стандартної віртуальної машини (інтерпретатора) Java, серверу і множини агентів.

**Voyager.** Voyager кампанії ObjectSpace - це платформа для розподілених обчислень за допомогою агентів, яка також реалізована на Java. Voyager забезпечує обмін повідомленнями між об'єктами та дозволяє об'єктам пересуватися, як це роблять агенти в мережі. Можна сказати, що Voyager дозволяє програмістам на Java створювати застосування як за традиційною, так і за “агент-орієнтованою” технологіями створення розподілених обчислювальних систем.

Звернемо увагу, що системи мобільних агентів, реалізовані на Java мають багато спільного. Справа в тому, що вони прив’язані до стандартних версій віртуальної машини Java і механізму серіалізації об’єктів. Загальна серверна архітектура властива всім системам. Істотні відмінності спостерігаються в механізмах переміщення агентів і обміну повідомленнями.

Хоча більшість сучасних систем мобільних агентів базуються на Java, існують інші мови для їх реалізації. До них, в першу чергу, належать Tcl і Python.

**Agent Tcl.** Це система, агенти якої пишуться на Tcl. Agent Tcl, розроблена в коледжі Дартмоут (Dartmouth), має потужні сервіси для переміщення і зв'язку, механізм захисту інформації, а також засоби для відлагодження і трасування. Головною компонентою Agent Tcl є сервер, який працює на кожній машині і дозволяє агентам рухатись і обмінюватись повідомленнями. Коли агент мігрує до іншої машини, він викликає функцію agent\_jump, яка запам’ятовує стан агента і посилає цю інформацію до серверу на машині адресата. Сервер машини адресата викликає на виконання Tcl, завантажує інформацію про стан в середовище виконання, і запускає агента в тому ж стані, в якому він завершив свою роботу.

**Ara.** Ara, розробка університету Кайзерслотерн (Kaiserslautern), це заснована на Tcl платформа для безпечної роботи мобільних агентів в різнорідних мережах. Дослідницький проект, перш за все, спрямовано на підтримку системи мобільних агентів, як середовища для безпечного і ефективного функціонування.

**TACOMA.** Проект TACOMA спрямований на дослідження операційних систем, як середовища для підтримки агентів, а також на використання агентів для розв’язання задач, які традиційно належать до функцій операційних систем. TACOMA базується на UNIX і TCP. Система підтримує агентів, які написані на C, Tcl/Tk, Perl, Pyhton і Scheme (Elk). Власне систему реалізовано на C.

Загальною рисою для ряду проектів, що базуються на Tcl, є те, що вони передбачають рух до підтримки декількох мов, що по суті означає можливість інтеграції з системами мобільних агентів, реалізованих на Java.

8. Стандартизація мобільних агентів: MASIF

Очевидним є те, що вищезазначені системи дуже відрізняються архітектурою і реалізацією, і таким чином перешкоджають сумісності, а згодом і стрімкому розгортанню технологій мобільних агентів на сучасному ринку. Тому здається природнім для підвищення сумісності нормалізувати деякі аспекти технології. Компанії Crystaliz, General Magic Inc., GMD Fokus, IBM Corporation і Open Group спільно розробили проект “Засоби сумісності систем мобільних агентів” (Mobile Agent System Interoperability Facility – MASIF).

MASIF описує стандартні інтерфейси систем мобільних агентів (не плутати з інтерфейсом власне агентів і систем їх підтримки !). MASIF не містить домовленості про єдину мову. Мовна сумісність для мобільних об'єктів – надскладна проблема, і в MASIF вимагається тільки сумісність систем мобільних агентів, які пишуться на одній мові, але потенційно різними продавцями. До того ж, в MASIF не нормалізуються локальні дії агента як, наприклад, інтерпретація агента, серіалізація або виконання. Можна сказати, що MASIF визначає інтерфейси на рівні систем мобільних агентів, а не на рівні агента.

В MASIF стандартизовано наступні чотири області:

**Управління агентами.** В множині мобільних агентів бажано стандартизувати управління агентом. Необхідно, щоб адміністратори систем агентів різних типів могли застосовувати стандартні процедури для створення агентів за ім'ям класу агента, переривати виконання агента, відновлювати виконання та завершувати виконання агента.

 **Передача агентів.** Бажано, щоб застосування могли породжувати таких агентів, які будуть здатні вільно пересуватися між системами мобільних агентів різних типів.

 **Агент і його системне ім’я.** На додаток до стандартизації операцій, які забезпечать сумісність систем мобільних агентів, підлягає стандартизації синтаксис і семантика різних параметрів. Ім'я агента та ім'я системи агентів повинні бути нормалізовані. Це дозволить системам агентів і агентам розпізнавати один одного.

**Системний тип агента і синтаксис розміщення.** Синтаксис розміщення повинен бути стандартизований так, щоб агент мав доступ до інформації про тип системи агентів бажаної (тієї, куди має намір переміститися) системи агентів. Передача агента може статися тільки в тому випадку, якщо тип системи агентів адресата підтримує агента.

9. Підсумок

Мобільний агент - це агент, який не прив'язаний до системи, де починається його виконання. Він здатен переміщувати себе від однієї системи в мережі до іншого. Здатність мандрувати дозволяє мобільному агенту пересуватися для досягнення певної мети, яка задається користувачем. Існує принаймні сім гарних причин для використання мобільних агентів: вони зменшують трафік в мережі, долають мережеву латентність, вони герметизують протоколи, виконуються асінхронно і автономно, динамічно адаптуються до середовища, природно різнорідні і стійкі до несправностей в мережі.

Прикладними областями, де проявляються переваги технологій мобільних агентів є: електронна комерція, персональна допомога, приватне спілкування, розподілений пошук інформації, послуги телекомунікацій, організація групових робіт, моніторинг, розповсюдження інформації і паралельне виконання.

Щоб підвищити сумісність, деякі аспекти технологій мобільних агентів було стандартизовано за допомогою OMG's MASIF. Як вплине часткова стандартизація на розвиток технологій мобільних агентів і розподілених обчислень, буде відомо у майбутньому. Дослідження тривають.

10. Література і ресурси

1. Danny B. Lange - Mobile Objects and Mobile Agents: The Future of Distributed Computing?
2. Lange, D.B. and Oshima, M.: Programming and Deploying Java™ Mobile Agents with Aglets™, Addison-Wesley, 1998.

**Aglets** at www.trl.ibm.co.jp/aglets.

**Odyssey** at www.genmagic.com/agents.

**Concordia** at www.meitca.com/HSL/Projects/Concordia.

**Voyager** at www.objectspace.com/voyager.

**Agent Tcl** at www.cs.dartmouth.edu/~agent.

**Ara** at www.uni-kl.de/AG-Nehmer/Ara.

**TACOMA** at www.cs.uit.no/DOS/Tacoma.