Mobile Agents

Short Version

As Part of the Lecture on Distributes Systems

June 2000

Mobile Agents ←

are programs that *roam the Internet* and that do usefule things on behalf of their owner or user when visitng a far away host or WWW server. An attractive paradigm, but its complexity is often underestimated. Do we really need mobile agents? We will look into the various problems with mobile agents, but also talk about potential benefits.

Mobile agents are a solution in search of a problem

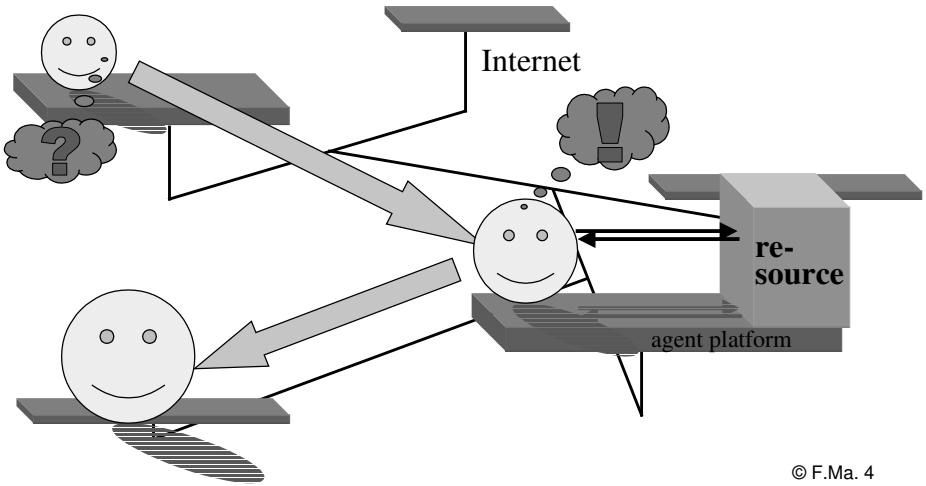
John Ousterhout

Outline

- What are agents and mobile agents?
  - mobile code
  - use of mobile agents
  - application areas
- Agent infrastructure and agent platforms
  - migration
  - communication
  - security
- Conceptual problems
- Pragmatics

© F.Ma. 3

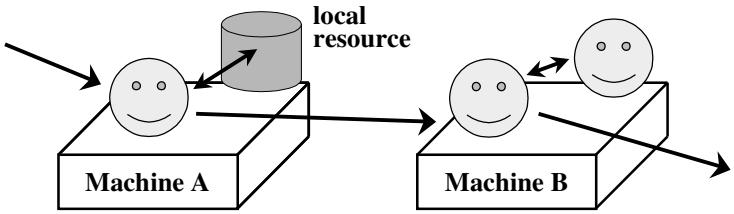
Mobile Agents



© F.Ma. 4

Mobile Agents - a First Characterization

- Program instances that are able move within a heterogeneous network (Internet, intranet)
  - under their own control
  - interact with local resources and other agents
- And do something useful
  - as a delegate of some "authority" (e.g., their creator)

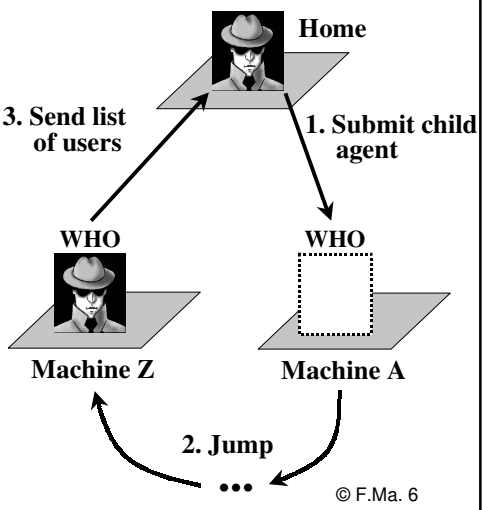


© F.Ma. 5

Example of a Roaming Agent: The "who" Agent (D'Agents Project)

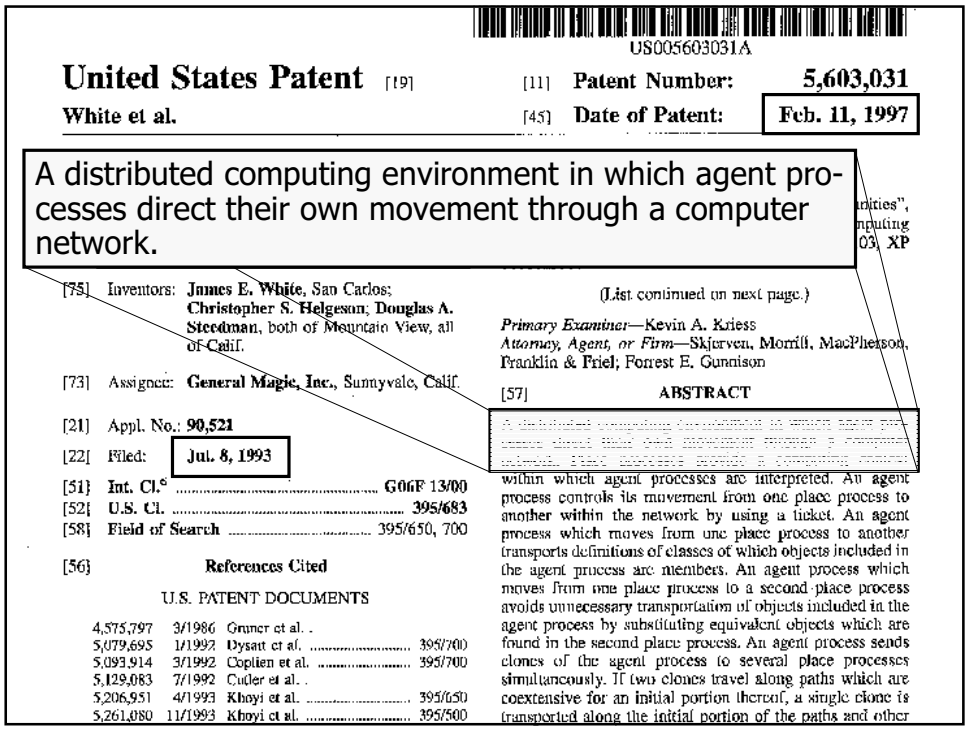
Child Agent

```
...
set results {}

  # migrate through machines

foreach machine $machines {
  agent_jump $machine
  append results \
      [exec who << {} ]
}

  # send back results

agent_send \
  $agent(root) 0 $results
...
```



source: D'agents project

© F.Ma. 6



US005603031A

United States Patent [19]

White et al.

[11] **Patent Number: 5,603,031**

[45] **Date of Patent: Feb. 11, 1997**

A distributed computing environment in which agent processes direct their own movement through a computer network.

[75] Inventors: **James E. White**, San Carlos; **Christopher S. Helgeson**; **Douglas A. Steedman**, both of Mountain View, all of Calif.

[73] Assignee: **General Magic, Inc.**, Sunnyvale, Calif.

[21] Appl. No.: **90,521**

[22] Filed: **Jul. 8, 1993**

[51] **Int. Cl.**[6] **G06F 13/00**

[52] **U.S. Cl.** **395/683**

[58] **Field of Search** 395/650, 700

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,575,797	3/1986	Gruner et al. .	
5,079,695	1/1992	Dysart et al.	395/700
5,093,914	3/1992	Coplien et al.	395/700
5,129,083	7/1992	Cutler et al. .	
5,206,951	4/1993	Khoyi et al.	395/650
5,261,080	11/1993	Khoyi et al.	395/500

(List continued on next page.)

Primary Examiner—Kevin A. Kriess
Attorney, Agent, or Firm—Skjerven, Morrill, MacPherson, Franklin & Friel; Forrest E. Gunnison

[57] **ABSTRACT**

[redacted]

within which agent processes are interpreted. An agent process controls its movement from one place process to another within the network by using a ticket. An agent process which moves from one place process to another transports definitions of classes of which objects included in the agent process are members. An agent process which moves from one place process to a second place process avoids unnecessary transportation of objects included in the agent process by substituting equivalent objects which are found in the second place process. An agent process sends clones of the agent process to several place processes simultaneously. If two clones travel along paths which are coextensive for an initial portion thereof, a single clone is transported along the initial portion of the paths and other

Good Times for Mobile Agents

- Internet grows and becomes ubiquitous
  - becomes dynamic and mobile
  - networked embedded systems
- New applications
  - plug & play for the masses
  - adaptive software, customization
- Shorter development cycles
  - fast reaction to dynamic market
  - remote configuration & installation
- New market strategies
  - e.g., dynamic federations of services

Mobile agents may be well suited for these modern times: they allow dynamic and flexible system structures *(e.g., function shipping, call by visit, code on demand...)*

© F.Ma. 8

What is an Agent?

- No consensus on precise definition!
- Latin „agere": to act, to drive

- Authorized to act autonomously in the name of or in place for someone
  - someone delegates a task to an agent
  - agents are representatives (of some „authority")

- Software agent
  - as opposed to persons or physical robots
  - „a computer program acting autonomously on behalf of..." [OMG MASIF]

© F.Ma. 9

General Agent Properties

- Autonomous
  - have control over their own actions and may operate without direct intervention of others
- Interactive, communicative
  - communicate with other agents, with humans, with their home base, with WWW resources…
  - problem: syntax and semantics of communication?

Intelligent Agent Properties - The AI View

- Goal oriented
- Reactive
  - they perceive their environment and respond to changes that occur in it
- Proactive
  - take initiative, not only respond to the environment
- Cooperative, social
- Intelligent
  - perform domain oriented reasoning
- Adaptive
  - they learn

Research on intelligent agents was initiated quite early, e.g. by Carl Hewitt („actor model", 1977) or Marvin Minsky („society of mind")

Mobility is not an issue here!

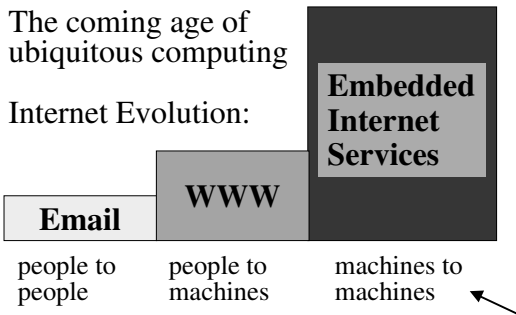
Agents: What Are They Good for?

- Find and filter information
- Automate work
  - respond to events, such as a new version
- Customize views of information
  - e.g., email
- Make recommendations
- Perform simple cooperative tasks
  - e.g., scheduling of meetings
- ...

In the AI view, agents are often perceived as intelligent assistants (e.g., personal agents or„interface agents" that learn from user actions)

© F.Ma. 12

Good Times for Intelligent Agents



- Automatic processes without user intervention
  - cooperating autonomous, decentralized entities
  - goal oriented (some useful task has to be done)
  - reactive (changing and dynamic environment)

==> Agent paradigm is quite natural here!
(already in the WWW age we have software robots and web crawlers)

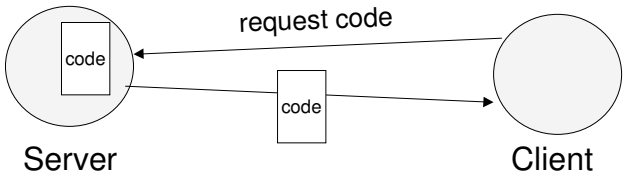
© F.Ma. 13

Mobile Code

- Basic idea:
  - program code is moved to a remote site and executed there
- Problems:
  - heterogeneity of systems
    - machine code, resources at remote site
  - security (as more parties are involved)
- Systems can be classified according to initiator of code transfer
  - *pull* or *push*

© F.Ma. 14

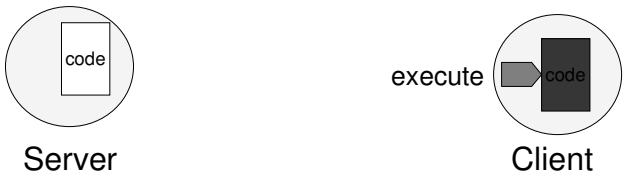
Mobile Code: Pull Principle



- Client asks for code and executes it at client site
- Example: Java applets

source: Fritz Hohl

Mobile Code: Pull Principle



- Client asks for code and executes it at client site
- Example: Java applets

source: Fritz Hohl

Applets As Mobile Code

- Java & WWW
- Not yet a mobile agent
  - applet cannot move further on
  - moves only from server to client, not the other way
  - does not migrate autonomously, has to be fetched

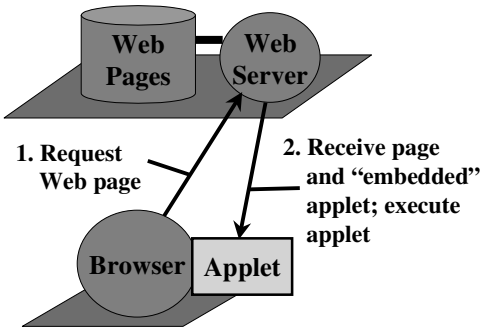
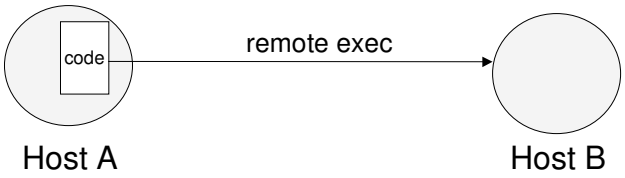


Image source: D'agents project

Mobile Code: Push Principle

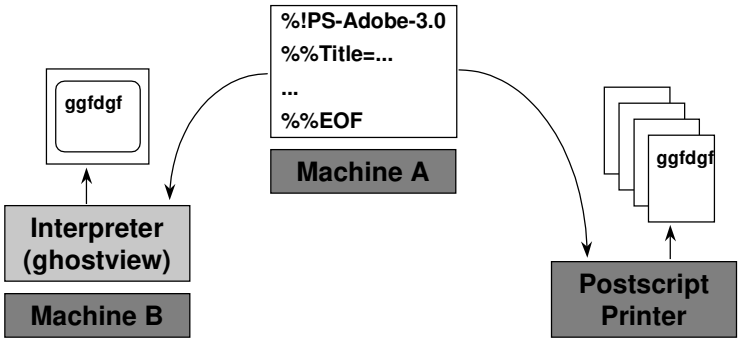


- Host sends code to another Host to execute it there ("remote execution")
- Example: postscript page description language

source: Fritz Hohl

Postscript - an Example of Mobile Code

- Postscript code is executed by remote machine
  - physical or virtual machine

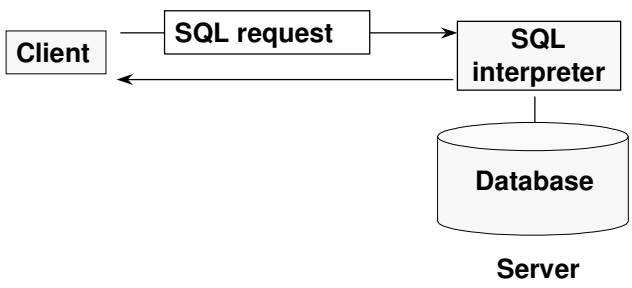


source: Hagimont

© F.Ma. 19

SQL - Another Example of Mobile Code

- SQL code sent to database server
- Code excuted by remote SQL interpreter



source: Hagimont

© F.Ma. 20

Stored Procedures vs. Mobile Code

- Typically used with Databases
- Not a mobile agent
  - procedure cannot move further on
  - cannot spawn other procedures
  - does not actively communicate with other objects

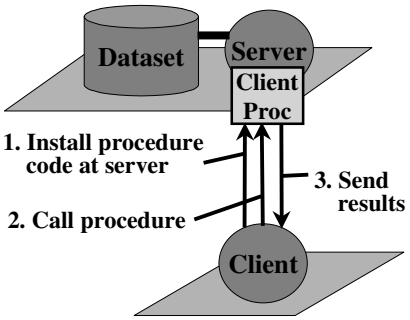


Image source: D'agents project

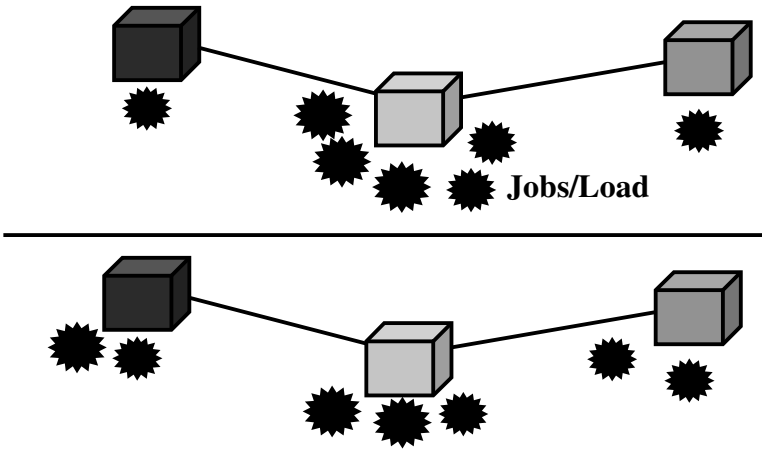
© F.Ma. 21

Mobile Code: Applications and Reasons

- Efficiency
  - load balancing
  - move function to faster machine or closer to the data
- Automatic software update
  - push principle
  - code may be provided by remote code server
- Dynamic extensions of services
- ...

© F.Ma. 22

Load Balancing with Mobile Code

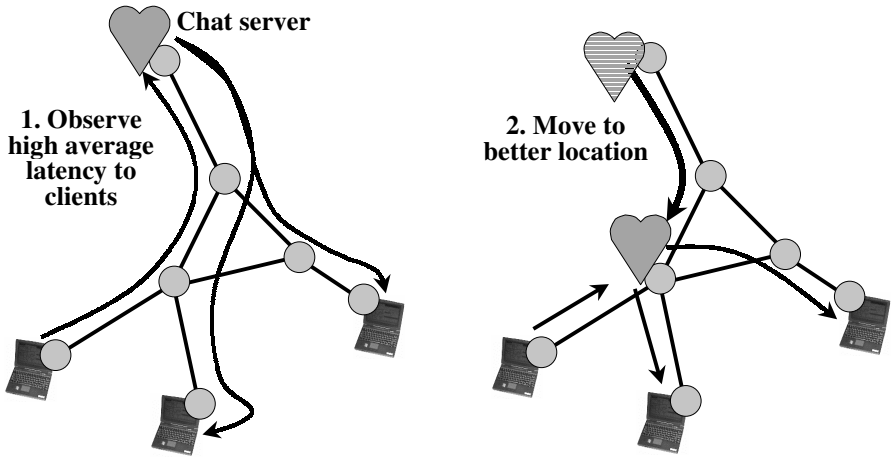


Jobs/Load migrate in a network of machines

source: D'agents project

© F.Ma. 23

Latency Reduction with Mobile Code



source: D'agents project

© F.Ma. 24

Mobile Code is Nothing New

- *Process migration* in operating systems
  - typically on a local scale (e.g., LAN cluster)
  - main goal: load balancing
  - migration initiated by the operating system
  - challenge: location transparency, efficency
  - more recent variant: *object* migration (e.g., Emerald, Cool)
- Active mail
- ...

Mobile Agents and Mobile Code

- Mobile agents consist of
  - code (are therefore also mobile code entities)
- But in addition
  - data state (i.e. variables)
  - execution state (stack, program counter)
- Agents decide on their own when to migrate
  - what about location transparancy?
  - aren't agents *location aware* by design and purpose?

All this makes agent migration more complex than simple code migration!

© F.Ma. 26

What Makes Mobile Agents Attractive?

Compared to classical client-server programming (Remote Procedure Call: RPC)

- More dynamics
  - compared to pre-installed remote procedures
- Asynchronous behavior and parallelism
  - while the agent acts on behalf of an authority, the authority may perfom other tasks
- Improved real-time abilities, fast reaction to remote events
  - acts near the source of the event (but represents a distant entity)

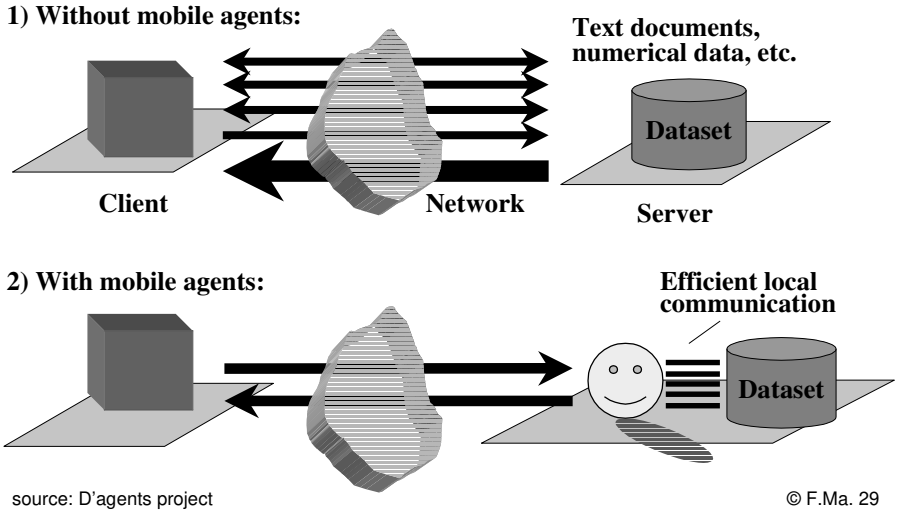
© F.Ma. 27

...Attractive?

- Higher degree of robustness and fault tolerance (?)
  - agent may decide autonomously to go to a backup server
  - however: more things can go wrong with mobile agents!
- Support of nomadic computing
  - „migrate and disconnect": launch agent during brief connected session
- Use of local resources that are not „exported"
- Reduced communication bandwidth
  - but: when is it better to send data to the program instead of sending the program to the source of data?

© F.Ma. 28

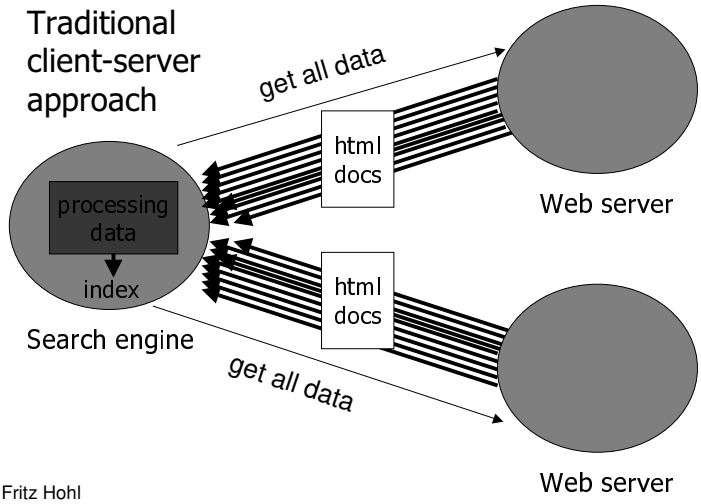
Bandwidth Conservation



source: D'agents project

© F.Ma. 29

Example: Search Engine



source: Fritz Hohl

© F.Ma. 30

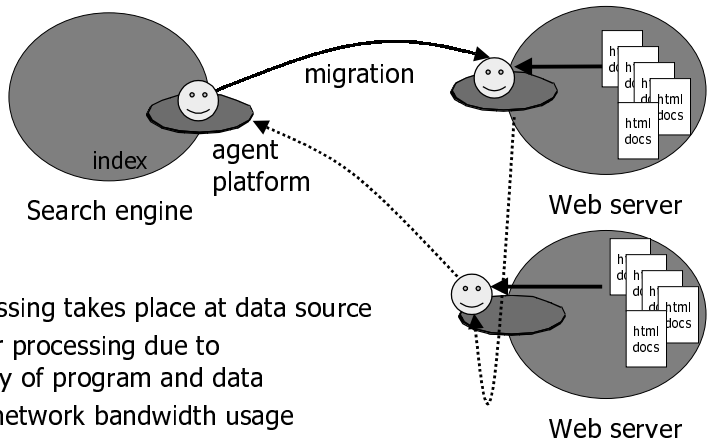
Example: Search Engine

- The traditional client-server approach
  - does not scale
  - induces extensive network usage
- Bottlenecks:
  - network bandwidth
  - processing capacity of search engine

source: Fritz Hohl

© F.Ma. 31

The Mobile Agent Approach

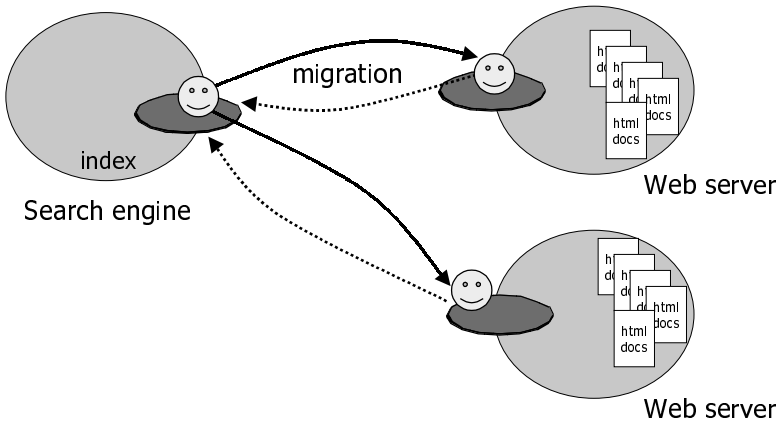


- Processing takes place at data source
- Faster processing due to locality of program and data
- Less network bandwidth usage

source: Fritz Hohl

© F.Ma. 32

Parallel Processing: Spawning Clones

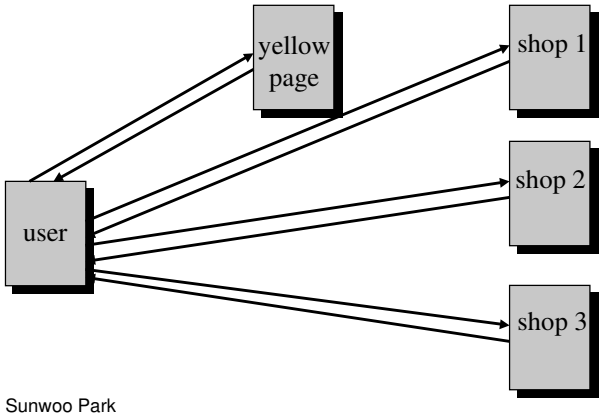


source: Fritz Hohl

© F.Ma. 33

Another Scenario: Electronic Commerce

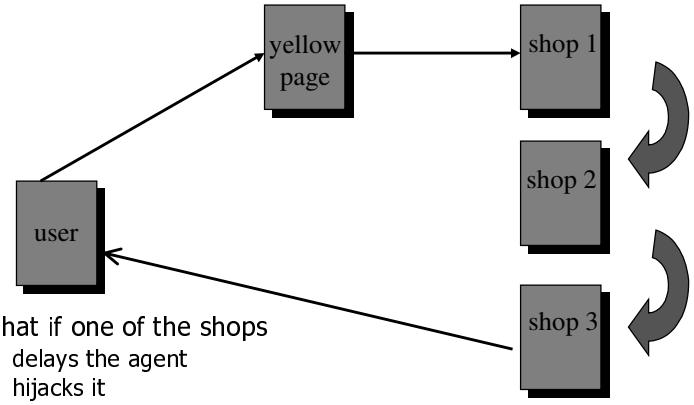
The traditional client-server approach



source: Sunwoo Park

© F.Ma. 34

Electronic Commerce Based on Agent Roaming



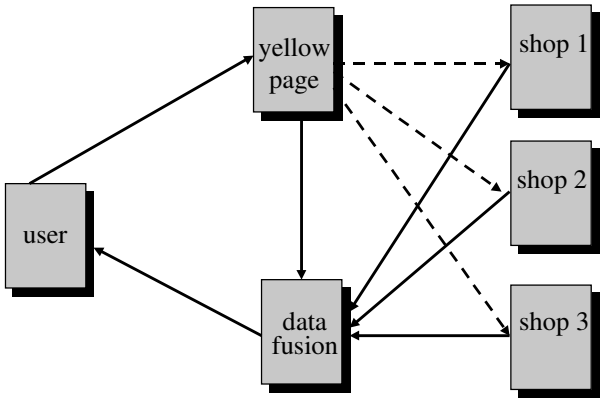
But what if one of the shops

- delays the agent
- hijacks it
- forges the information
- steals data from other shops?

source: Sunwoo Park

© F.Ma. 35

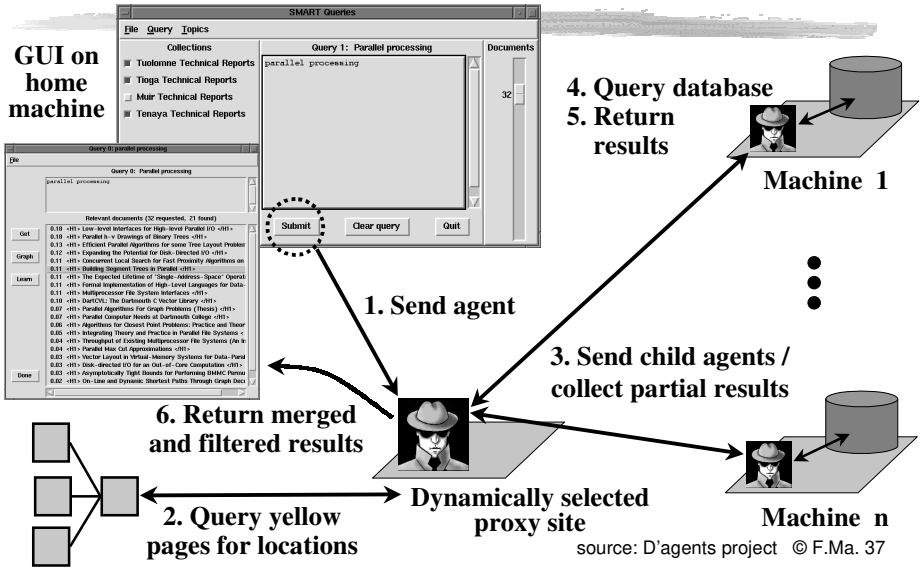
Electronic Commerce Based on Agent Cloning



source: Sunwoo Park

© F.Ma. 36

Example: Technical Reports
(D'Agents Project, Darthmouth College)



source: D'agents project © F.Ma. 37

Semantic Information Compression with Agents

- Client sends an agent with a specific search routine to a server
- Server does remote filtering of its data base according to the search routine
- Only the filtered documents, not the whole data base is sent back to the client
- Similarly, the agent could carry an application specific compression algorithm or an encryption scheme

© F.Ma. 38

Application Areas of Mobile Agents

- Electronic Commerce
  - agent as a seller, buyer, trader, broker...
  - send personal agent on a shopping tour
- Information search and retrieval
  - remote filtering and compression
- Mobile computing
  - migrate-and-disconnect style of operation
- Remote control, diagnosis, maintenance
  - monitoring (e.g., network management)
  - remote installation of software components
- Watchdogs react locally on behalf of an authority
  - my agent at Dow Jones will alert me...

© F.Ma. 39

... Application Areas

- Workflow management, groupware applications
  - active documents with semantic routines to process their content
  - cooperation services (e.g., scheduling of meetings)
- Personalized Internet services
- Entertainment
  - represent player on a game host
  - distributed multi-user games
- Mobile agents as a programming paradigm
  - higher-level abstraction
  - unifies "process" and "object" ("active objects")

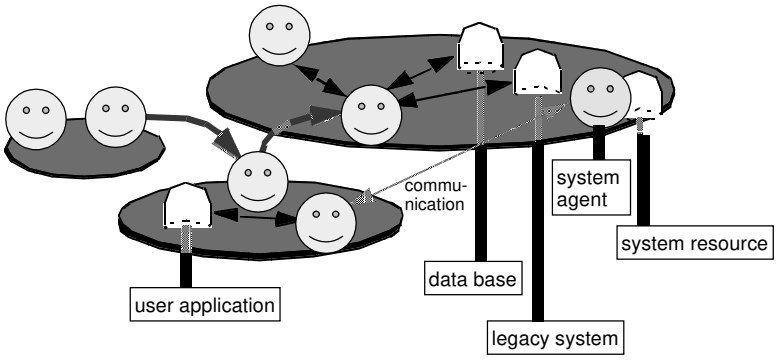
© F.Ma. 40

Outline

- What are agents and mobile agents?
  - mobile code
  - use of mobile agents
  - application areas
- Agent infrastructure and agent platforms
  - migration
  - communication
  - security
- Conceptual problems
- Pragmatics

© F.Ma. 41

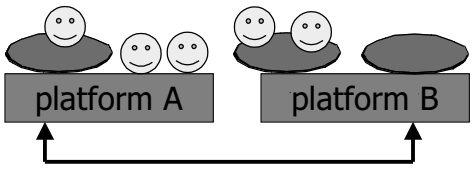
Agent Places



- Places are abstractions of computer nodes
  - local system resources may be "wrapped" by system agents
- Allow to distinguish locality from globality
  - however: one machine may contain many places

© F.Ma. 42

Mobile Agent Platform



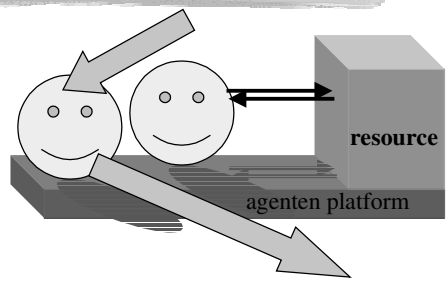
Often, there is no distinction made between places and platforms

- "Docking station" and meeting point for agents
- Local infrastructure similar to services of a hotel
  - launching, receiving, providing residence for agents
  - provides services, resources, run-time support to agents
  - e.g. "concierge service" (inform about locally available resources, other resident agents,...)
- Cooperation among platforms (--> global infrastructure)
  - forwarding agents, locating agents, communication mediation
  - requires standard protocols and interfaces

© F.Ma. 43

Local Infrastructure

- Mobility support
- Communication
  - with local / remote agents
  - with „authority" (user, owner)
  - communication paradigm?
- Take care of security
- Event delivery / management
- Exception handling
- Fault tolerance
  - transactions?
- Resource management
  - service mediation, access rights
  - fair usage (cpu, memory...)



- Can be quite complex
- Requires API or „language"
  e.g., negotiation of local resource usage

© F.Ma. 44

Global Infrastructure

- Migration
  - also: exception handling when goal not reachable
- Locating agents, searching lost agents
  - more than a simple „name service"!
- Forwarding messages
- Termination of agents / groups of agents
  - orphan detection
- Brokering, service directory, yellow pages
- Connection to established system infrastructures
  - CORBA, Jini, WWW, Internet services,...
  - application frameworks and legacy systems
- Fault tolerance?
- Monitoring, management, debugging?
- Interoperability?

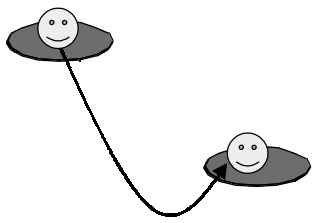
Requires cooperation among all platforms

Can be quite complex!

© F.Ma. 45

Migration

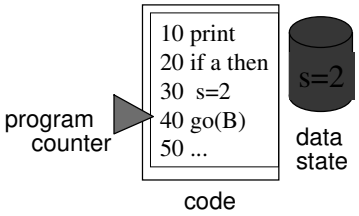
- Transport of an agent to another place
- Autonomous
  - agent initiates migration (and decides upon goal)
- *Weak* migration
  - only program code is migrated
  - agent restarted at migration goal
- *Strong* migration
  - agent keeps its context
  - continues with next instruction



```
x := ...;
if (x > 17) then go supercomputer;
y := f(x) + 29;
```

© F.Ma. 46

Strong Migration

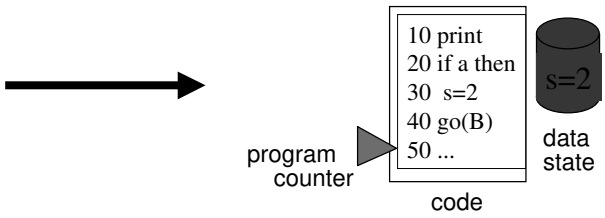


- Platform has all data associated to the agent, i.e.
  - code
  - data state (contents of variables)
  - execution state
    - program counter
    - operand stack
    - heap
    - calling stack

But does the platform know *how* to access all these data?

© F.Ma. 47

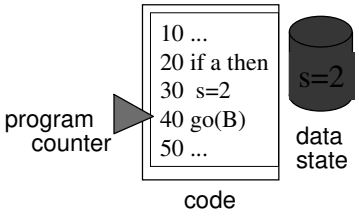
Strong Migration



- Platform serialises data and sends it to next node
- Program continues at next statement
- Transparent to the programmer
  - question: is it transparent to the *agent?*

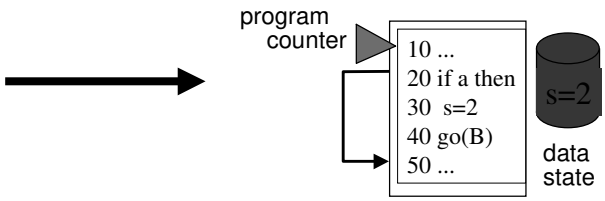
© F.Ma. 48

Weak Migration



- Platform does not access execution state of the agent, but only
  - code
  - data state

Weak Migration



- Program restarts at some entry-point
  - typically at the beginning
- Non-transparent to the programmer
  - manually save important parts from execution state
  - when restarting: restore relevant parts of saved state and jump to a "continuation label"
    - non-trivial for nested control structures

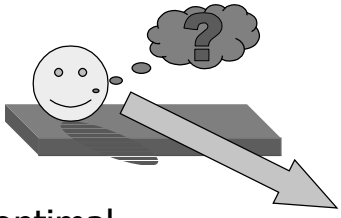
Strong vs. Weak Migration

- Strong migration
  - easier to use (for the agent programmer)
  - challenge for the implementor
    - migration in the middle of an expression evaluation when calling a function?
    - can strong migration be realized on top of Java without changing the JVM?
- Weak migration
  - easier to implement
  - resource rebinding and relevant execution state restauration has to be handled explicitly
  - in general more efficient due to less overhead
  - probably sufficient in many cases

© F.Ma. 51

Why Migrate?

- Another machine has unique resources
- Plan dictates migration
  - e.g., roaming shopping agents
- Event initiates migration
  - e.g., agent called home
- Local platform has become suboptimal
- Local platform will be disconnected or cease to exist

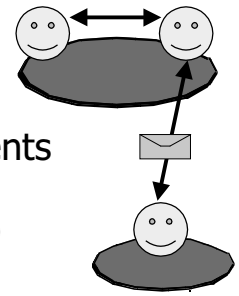


© F.Ma. 52

Communication Between Agents

- For agents at the same place ("meetings")
  - procedure call
  - method invocation
  - …

call by visit



- Communication between distant agents
  - RPC (or RMI)
  - messages (mailboxes at agent places?)
  - abstract global information spaces
    - e.g., tuple spaces
- Address of target agent?
  - global name (name service required!)
  - <name of platform> + <local id>

Languages for higher-level communication?
- semantics?
- ontologies?

© F.Ma. 53

Communication with Migrating Peers?

- How to find the peer?
- What to do if an agent migrates during communication?
- Approaches: location service, forwarding chains

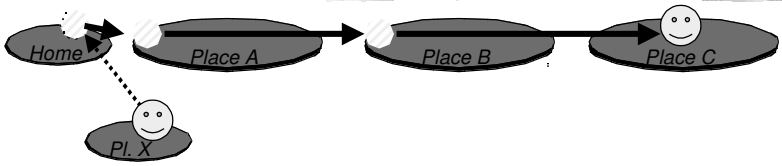
© F.Ma. 54

Location Service

- Agent informs location service about new location when migrating
- Sender may ask location service for "current" location of receiver
- Problems:
  - atomicity of location information update?
  - location while in transit?

© F.Ma. 55

Message Forwarding with Proxies



- Agent leaves a path of proxies that reference to next target place
- Communication is done via proxies
- Proxies relay communication to next proxy
- Problems:
  - possibly long forwarding chains
  - message may never catch a fast moving agent

© F.Ma. 56

Outline

- What are agents and mobile agents?
  - mobile code
  - use of mobile agents
  - application areas
- Agent infrastructure and agent platforms
  - migration
  - communication
  - security
- Conceptual problems
- Pragmatics

© F.Ma. 57

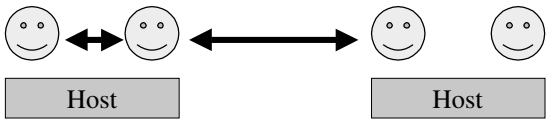
Security is Important

- We want open mobile agent systems, i.e.
  - everybody can deploy mobile agents
  - everybody can operate a host
  - everybody can offer services
- Electronic commerce is an important target
  - money is involved
  - security is important

Security is a key factor for agent technology!

© F.Ma. 58

Agent-Agent Security

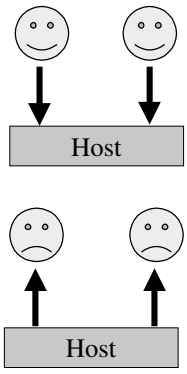


- Some attacks:
  - manipulation of other agents
  - masking the identity
  - denial-of-service
- Measures:
  - access control
  - authentication
  - not a mobility problem

☛ **Existing techniques can be applied!**

© F.Ma. 59

Host-Agent Security



- Protection of hosts from malicious agents
  - that's the easier part: well-known techniques
  - malicious agent similar to a virus
- Protection of agents from malicious hosts
  - difficult

© F.Ma. 60

Protection of Hosts from Agents

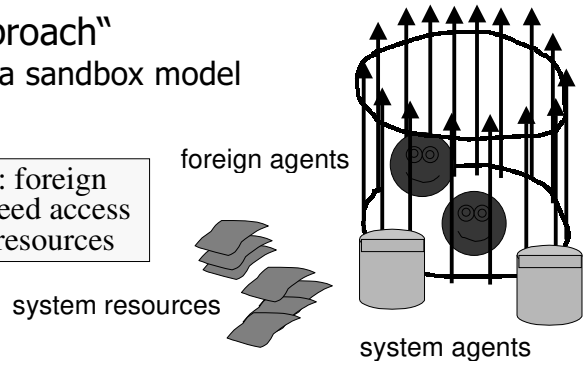
- Code signing with cryptographic means
  - authentication: the receiver will trust the sender
  - belief that trusted sources behave correctly
- Code verification: the host may try to examine the code of the agent
  - only of limited applicability
    - what properties can be verified automatically?
  - proof carrying code?
    - proof (that some safety property is not violated) is delivered with code; receiver verifies it (which is easier than proving)
- Access-level control
  - security manager checks whether activities are allowed
  - sandbox model

© F.Ma. 61

Protection of Hosts from Agents

- „Cage approach"
  - e.g., Java sandbox model

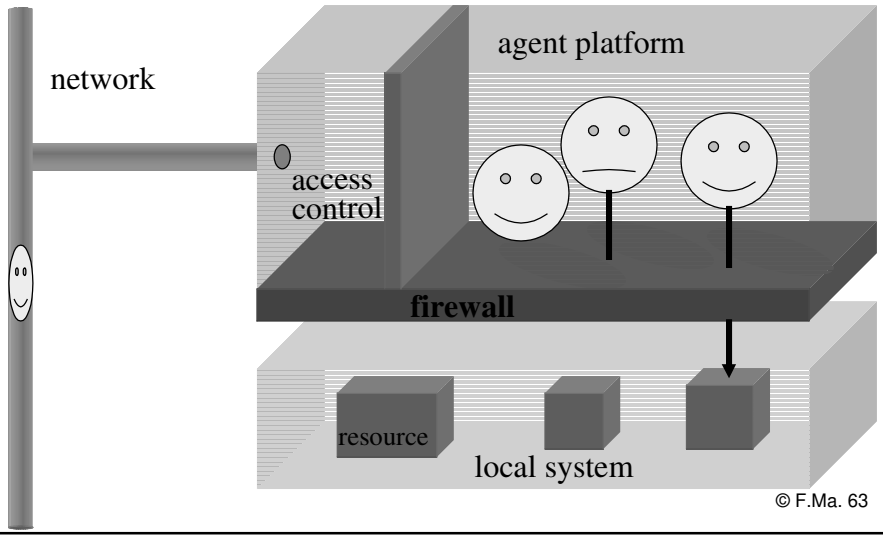
problem: foreign agents need access to local resources



- Allow only controlled access to system resources
  - negotiate consumption limits
  - record agent activities

© F.Ma. 62

Access Control and Local Firewall



© F.Ma. 63

Trusted Services and Platforms

- Agents can be inserted by anyone
- Services can be offered only by trusted users
- Agent system is operated by trusted party
- Many applications, but
  - no open service market (electronic commerce!)
  - bad integration of legacy systems

© F.Ma. 64

Protection of Agents from Malicious Hosts, Example:

Agent shall buy flowers for its authority:

1. find all flower shops
2. migrate chain of shops and get price
3. go to cheapest shop and buy flowers

	prevent
`Address` **`home`** `= mx17@mypda.eac.eu`	modification
`Money` **`wallet`** `= 20$`	reading
`float` **`maximumprice`** `= 20$`	reading
`good` **`flowers`** `= 10 red roses`	
`int` **`shoplistindex`** `= 0`	modification
`float` **`bestprice`** `= 20$`	modification
`Address` **`bestshop`** `= empty`	modification

source: Fritz Hohl

© F.Ma. 65

A Malicious Host Attack

Malicous host may modify the code and set bestshop to its shop:

```
if (shoplistindex >= (shoplist.length - 1)) {
  go(bestshop);
  buy(flowers,wallet);
}
```

go(MYSHOP);

© F.Ma. 66

Possible Attacks of Malicious Hosts

- Reading or modifying data or code
- Masking the identity of the host
- Incorrect execution of code
- Denial of execution or communication
- Reading/modifying communication with others
- Returning wrong results to system calls
- Wrong routing, no forwarding
- Cloning an agent
  - undecidable whether agent is a clone!

Agent cannot decide
• whether its code has been copied
• whether it has been cloned
• whether it is being executed correctly
• ...

Are there solutions?

Protection of Agents ?

- Thesis: Since the host has complete access to the agent, it is not possible to protect an agent from the host
  - agent must expose its code and data to the host which supplies the means for the agent to run
- But perhaps:
  - obfuscate or encrypt the agent code so that the host doesn't know what it is doing?
    - what does this mean and when does it help?
    - prevent at least reverse engineering?
  - use trusted hardware?

?

Tamper-Proof Environments (TPE)

- „Hardware blackbox"
  - secure execution environment for agents
  - trust (of the agent owner) towards TPE manufacturer
- Agent is encrypted with the public key of the TPE
  - TPE decrypts the agent using its private key and starts it
- Only crucial parts of the agent are kept in the TPE
- Probably limited to security-sensitive domains
  - e.g., banks, stock market

Measures Against Some Attacks

- The "trivial" solution: visit only trusted sites
- Determine whether the agent has been tampered
  - code signing
  - maintain a safe migration history (encrypted traces, audit logs); also use it against rerouting attacks
- Partition roaming agents
  - one component for each server visited
  - each server encrypts its component with the public key of the agent's authority
  - components are chained together (e.g., using hashes of earlier components)
  - decryption only on home base or on some other trusted site

• More difficult if there are collusions of several malicious hosts
• Some measures induce communi-cation with other hosts: this reduces autonomy and efficiency

Outline

- What are agents and mobile agents?
  - mobile code
  - use of mobile agents
  - application areas
- Agent infrastructure and agent platforms
  - migration
  - communication
  - security
- Conceptual problems
- Pragmatics

© F.Ma. 71

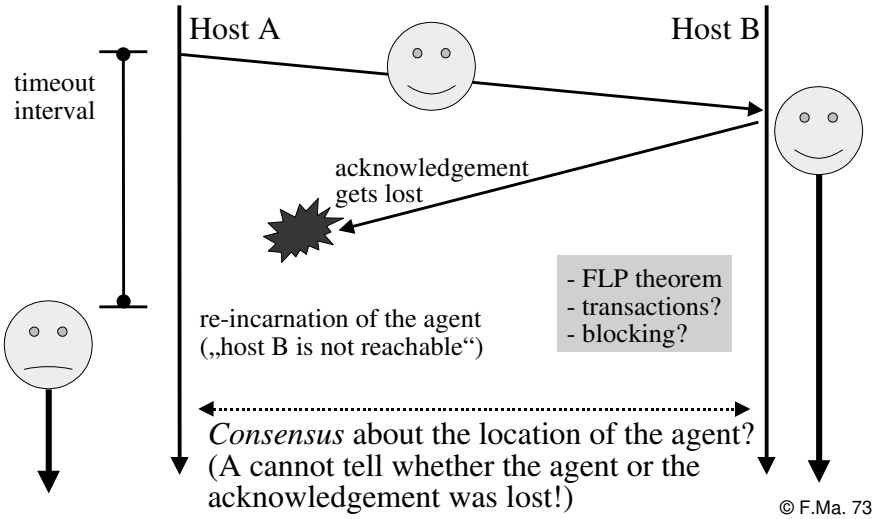
Conceptual Problems

- Exactly once migration?
- Hiding a secret?
- Semantics of mobility?

Many serious applications (such as electronic commerce) will only be realized with mobile agents when these problems (and other important problems such as security) are well-understood and solved!

© F.Ma. 72

Exactly-Once Migration?



© F.Ma. 73

Can an Agent Keep a Secret?

- Keep electronic money that cannot be copied?
- Execute a stategy, but without disclosing it to the host?
  - e.g., my mobile agent on the stock market server
  - it is the host that executes every single instruction of the agent, so the host „knows“ the agent code!

© F.Ma. 74

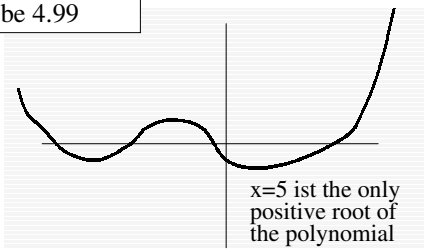
Secret Strategy? An Oversimplified Example

```
x = ask_for_price(…);
if (x > 5) then go to A;
        else go to B;
```

„right“ alternative; my price offer should be 4.99

equivalence transformation to hide the secret strategy

```
x = ask_for_price(…);
if (x**4+6*x**3-19*x**2-144*x)
> 180 then go to A;
```



But:
- when is that useful?
- what exactly is a „secret“?

One-way principle:
- equivalent expression easy to construct
- but difficult to „solve“ (and to understand!)

© F.Ma. 75

Semantics of Mobility?

- What is the formal meaning of „go to nextnode;“?
  - is it just „skip“?
  - is migration a purely nonfunctional issue?
- What exactly is the context of an agent?
- What parts of the context will be moved with the agent?
  - use same or only „equivalent“ local resource?
  - will `print(„hello“)` go to the local or remote display?
  - where will exceptions and error messages be delivered?
  - will events be forwarded?
- What happens in the case of an error?
  - e.g., migration goal not reachable
  - agent lost
  - resource not available at new location
  - local library for dynamic linking differs
  - source code no longer available with „copy by need“

All these are only *some* issues that have to be addressed!

© F.Ma. 76

Outline

- What are agents and mobile agents?
  - mobile code
  - use of mobile agents
  - application areas
- Agent infrastructure and agent platforms
  - migration
  - communication
  - security
- Conceptual problems
- Pragmatics

© F.Ma. 77

Implementation Issues

- Heterogeneity of machines and systems
- Interoperability?
- Should make use of established systems, e.g.
  - WWW protocols
  - Transport protocols and Internet infrastructure
  - security infrastructure
- Complete system is a major undertaking!

© F.Ma. 78

Mobile Agent Systems and Projects

- Serious development since about 1994
- List at University of Stuttgart contains (in February 2000) about 60 mobile agent systems
  - http://mole.informatik.uni-stuttgart.de/mal/mal.html
  - 50% Europe, 30% USA, 20% Asia
- Most are research prototypes
- Typically not interoperable with other systems

© F.Ma. 79

Programming Language?

- Agents typically migrate between heterogeneous systems
- Agents therefore should be written in a „universal" language
- Most scripting languages qualify for that
  - e.g., Tcl, Perl
  - interpreters are available for most systems
- Interpreters can easily influence the execution
  - e.g., filter system calls or trap illegal instructions
- Drawback: interpreted languages are less efficient
- Java-Bytecode is also available on most machines
  - interpreted by Java Virtual Machine (JVM)
  - generated by Java compiler from Java source code

© F.Ma. 80

Why Java in Most Projects?

- Available on "all" machines
- Large set of standard libraries and tools
- Portability: compiled into bytecodes for a stack-based virtual machine
- Code mobility ("code on demand")
  - object serialization
  - dynamic linking (local and remote: class loader, "code base")
- Security
  - access control by security manager

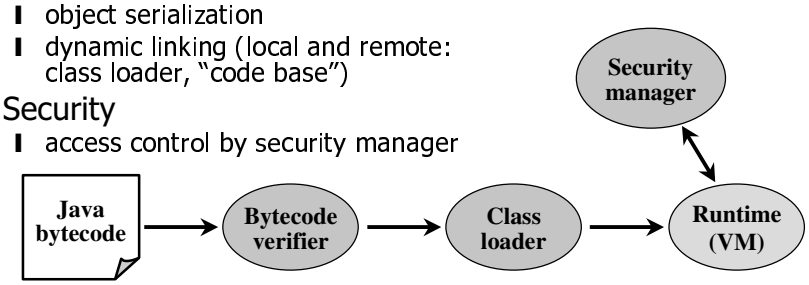


Image source: D'agents project

© F.Ma. 81

An Example of a Java-based System: Aglets

IBM **http://www.trl.ibm.co.jp/aglets/**

- Java-based Aglet-API
- Mobile agent = class that extends *Aglet*
- Graph of serializable objects
- Weak mobility
- Proxies for location transparency
- Persistent store

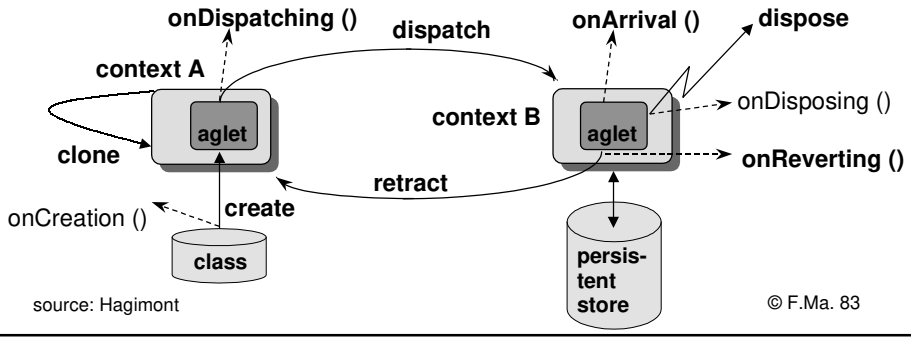
source: D'agents project

© F.Ma. 82

Aglets: Life Cycle and Event-driven Programming Model

- Event handler for „mobility events" associated to an aglet (*MobilityListener*)
  - onArrival (MobilityEvent l)
  - onDispatching (MobilityEvent l)
  - onReverting (MobilityEvent l)

  } to be implemented by the programmer



source: Hagimont

© F.Ma. 83

Messages in Aglets

- Consists of two fields
  - a type (*String*)
  - an object
- May be transmitted in three different ways
  - synchronous call
    - AgletProxy.sendMessage (Message msg)
  - asynchronous call
    - AgletProxy.sendAsyncMessage (Message msg)
  - oneway (asynchronous)
    - AgletProxy.sendOnewayMessage (Message msg)

source: Hagimont

© F.Ma. 84

Mobile Agent Standards

- OMG Object Management Group MASIF (1998)
  - "Mobile Agent System Interoperability Facility"
  - based on CORBA (ORB, IIOP,..)
  - defines some mobile agent systems aspects
  - some areas not addressed (e.g., migration)
- FIPA Foundation for Intelligent Physical Agents (1999)
  - defines intelligent and mobile agent issues
  - proposes KQML/KIF for communication
  - only two implementations
- Standards are not yet widely accepted

© F.Ma. 85

Mobile Agent Systems are Complex

- Takes time and effort to develop
- Paradigm seems to be attractive and nice
  - too many naïve pure research projects?
- Seems easy at first sight (after all, Java applets already exist) when ignoring:
  - infrastructure needed for real-world applications
  - interaction with the rest of the world
  - fundamental problems (fault tolerance, security…)
  - pragmatics (performance, e-commerce realities, simpler alternatives than mobile agents…)

© F.Ma. 86

Challenges and Problems

- Security!
- Fault tolerance (transactional support,…)
- Legacy system integration
- Interoperability; wide and easy deployment
  - build upon WWW infrastructure (e.g., platforms as browser plugins)?
- Global infrastructure
  - semantic routing
  - directories, location service, brokers,… (Jini, CORBA?)
  - common ontology (= vocabulary + meaning)
- Support of small, lightweight devices

© F.Ma. 87

Challenges and Problems (2)

- Agent control
  - trace and manage agents (suspend, restart, terminate)
  - orphan detection etc.
  - overpopulation by junk agents?
- Efficiency
- Go beyond prototype systems
- Market for agents?
  - services in an open Internet-based service market
  - value added service providers
- Intelligent mobile agents?
  - also: languages for agent cooperation

© F.Ma. 88

Conference !

2nd Int. Symp. on Agent Systems and Applications + 4th Int. Symp. on Mobile Agents

ASA/MA 2000

September 13-15, ETH Zurich

www.inf.ethz.ch/ASA-MA/

© F.Ma. 89

Mobile Agents Literature

James White:

Telescript Technology: The Foundation for the Electronic Marketplace. General Magic White Paper, 1994

Telescript Technology: Scenes from the the Electronic Marketplace. General Magic White Paper, 1994

Telescript Technology: An Introduction of the Language. General Magic White Paper, 1995

The classical papers

David Chess, Colin Harrison, Aaron Kershenbaum: **Mobile agents: Are they a good idea?.** In Jan Vitek; Christian Tschudin (eds.): Mobile Object Systems: Towards the Programmable Internet, pages 25-45. Lecture Notes in Computer Science No. 1222. Springer-Verlag, April 1997. http://www.research.ibm.com/massive/mobag.ps

Fundamental early mobile agents paper

© F.Ma. 90

Mobile Agents Literature (2)

Danny Lange, Mitsuru Oshima: **Programming and Deploying Java Mobile Agents with Aglets**, Addison-Wesley, 1998

Baumann, F. Hohl, K. Rothermel and M. Straßer (1998): **Mole - Concepts of a Mobile Agent System**, World Wide Web, Vol. 1, Nr. 3, pp. 123-137

http://www.informatik.uni-stuttgart.de/cgi-bin/ncstrl_rep_view.pl?/inf/ftp/pub/library/ncstrl.ustuttgart_fi/TR-1997-15/TR-1997-15.bib

Specific mobile agent systems

Milojicic, Breugst, Busse, Campbell, Covaci, et al: MASIF: **The OMG Mobile Agent System Interoperability Facility**, in: Rothermel; Hohl (eds.): Proceedings of the Second International Workshop on Mobile Agents, MA'98, Springer-Verlag, 1998

http://www.fipa.org/spec/fipa99spec0.2.htm

Mobile agents and standards

© F.Ma. 91

Mobile Agents Literature (3)

Giovanni Vigna (Ed.): **Mobile Agents and Security**. Springer-Verlag, 1998

A good collection of papers on an important subject

Stefan Fünfrocken, Friedemann Mattern: **Mobile Agents as an Architectural Concept for Internet-based Distributed Applications**. In: Steinmetz (Ed.): Proc. KiVS'99, pp. 32-43, Springer-Verlag, 1999

© F.Ma. 92

Mobile Agents: Resources

- General Resources

http://www.cs.umbc.edu/agents/
http://www.cetus-links.org/oo_mobile_agents.html

- The Mobile Agent List

http://mole.informatik.uni-stuttgart.de/mal/mal.html

- Mobile Agent Security Bibliography

http://mole.informatik.uni-stuttgart.de/security.html

- Mobility Mailing List

http://mobility.lboro.ac.uk/

- Agents Mailing List

http://www.cs.umbc.edu/agentslist/

© F.Ma. 93