



---

**Poseidon House  
Castle Park  
Cambridge CB3 0RD  
United Kingdom**

TELEPHONE:  
INTERNATIONAL:  
FAX:  
E-MAIL:

**Cambridge (01223) 515010  
+44 1223 515010  
+44 1223 359779  
apm@ansa.co.uk**

---

## **ANSA Phase III**

# **Scripts and Mobile Agents**

**Ashley McClenaghan**

### **Abstract**

This presentation has been prepared for the September 1995 visit to Fujitsu Labs. It provides an overview of ANSA's present work on Scripts and Mobile Agents.

"The Net is the computer" --- but how should it be programmed? Scripting and mobile agent technologies provide one possible answer to this question. The aim of the ANSA Workpackage on Scripts and Mobile Agents is to promote the evolution of open scripting and agent technologies for programming the Internet.

This presentation begins by introducing the world of scripts and agents, its motivations and research concerns. It then introduces two ANSA work tasks in this area. In the first task called CORBAagents, ANSA are developing a CORBA-compliant agent architecture. In the second task called Changeling, ANSA have developed a prototype Web server whose functionality can be extended by sending it scripts.

---

APM.1593.01

**Approved**  
Briefing Note

21st September 1995

---

**Distribution:**  
**Supersedes:**  
**Superseded by:**





# **Scripts and Mobile Agents**

## **Presentation to Fujitsu Labs, September 1995**

**Ashley McClenaghan**  
*am@ansa.co.uk*



## Contents

- **Global research areas**
- **Existing applications**
- **Standardisation**
- **Mobile agent paradigm**
- **Motivations**
- **“CORBAgent” (ANSA research)**
- **“Changeling” (ANSA research)**
- **Issues**



## Global Research Areas

“Intelligent agents”, “mobile scripts”, “softbots”, etc.

- **Mobile Scripts**
  - Designing scripting languages
  - Propriety languages: Telescript, Script-X, OREXX
  - Open languages: Tcl, Java, Python, Obliq
- **Agent interaction**
  - How agents communicate and find services
  - Interaction models: OpenDoc, OLE2, CORBA
  - Interaction languages: KQML, Cyc
- **Agent intelligence**
  - AI, neural nets, genetic algorithms

## ANSA's focus: Mobile Scripts



## Existing Applications

- **Fujitsu/CompuServe's WorldsAway Service using Agoric's Joule technology**
- **AT&T's PersonaLink Service using General Magic's Telescript technology**
- **Motorola's Envoy and Sony's MagicLink PDA's using General Magic's Telescript and Magic Cap technologies**
- **HotJava Web browser using Sun's Java technology**
- **Anderson Consulting's BargainFinder shopping agent using Web technology**

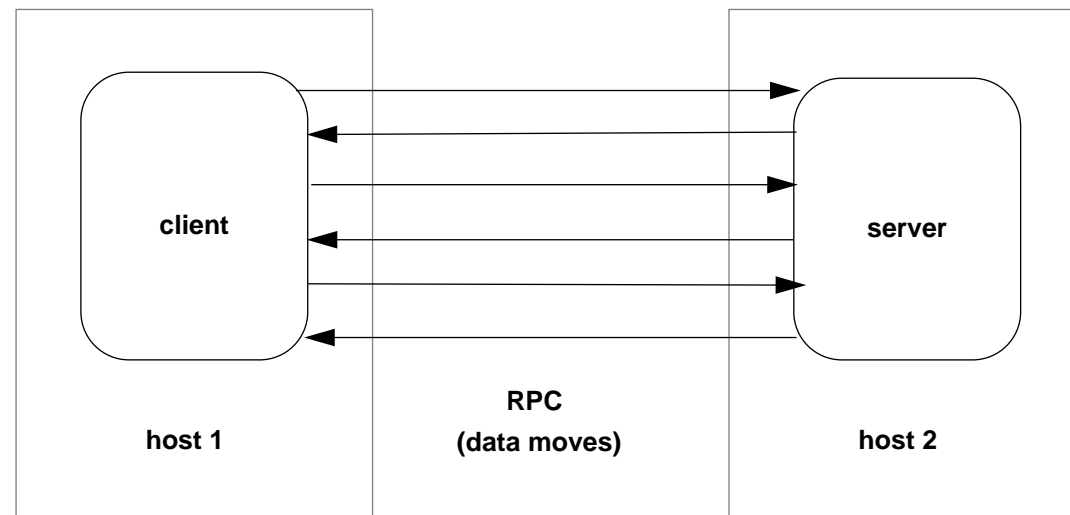


## OMG Standardisation

- **Developing an OMA (Object Management Architecture) compliant Reference Model which will allow heterogeneous agents to interwork**
- **Coordinated by the OMG's Common Facilities Task Force, Task Automation Group**
- **Relevant OMA Facilities:**
  - **Scripting: concerned with requirements for interpreted languages**
  - **Automation: providing methods that are semantically meaningful to access to large grained application objects such as spreadsheets, etc.**
  - **Rules Management and Workflow: representing organisational structures, business processes, flows of work using rule sets**
  - **Agent: requirements for agent execution engines and agent interaction languages**

## The Mobile Agent Paradigm: RPC

- **The traditional RPC model**

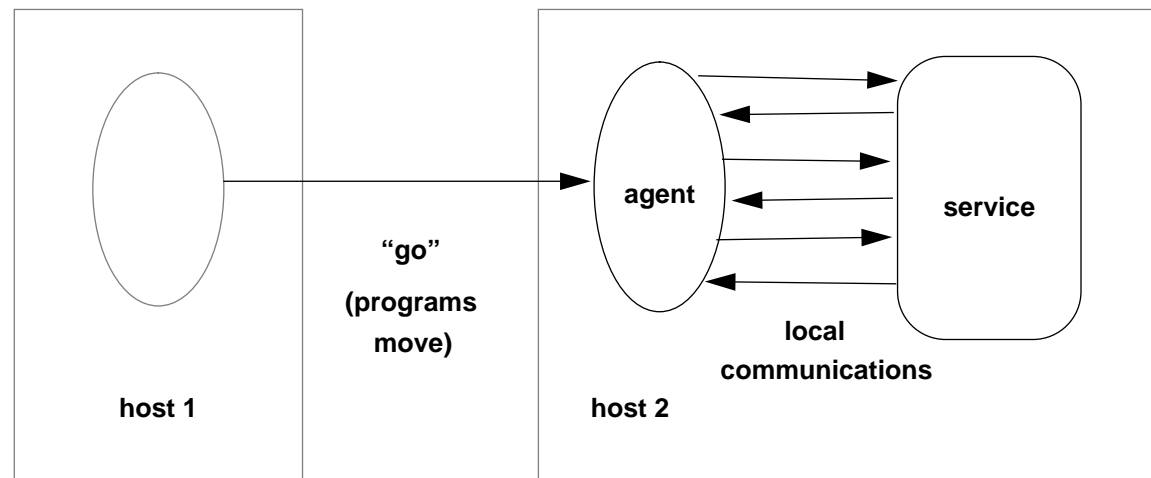


- **Client communicates with a non-local server**
- **The client/server interface is static**



## The Mobile Agent Paradigm: Remote Programming

- The new “remote programming” model



- The agent travels to the server’s host then interacts with the local server
- The client/server interface is not fixed --- interfaces and protocols can be introduced or upgraded on-line



## Motivations (1)

- **Intuitive structuring**
  - An agent has a “life”: it moves around its environment; it talks with other agents and services such as Directory, Banking and Taxi services
- **Reduced network communications**
- **Enabling remote computation**
  - Better use of resources
  - Off-load server bottlenecks by distributing the load to the clients
- **Simpler failure semantics**
  - Communication and state are always local
  - Less need for complex commit/rollback procedures to preserve atomicity across distributed communications



## Motivations (2)

- **Support intermittent connectivity**
  - E.g. PDAs intermittently connected to the network
- **Executable content**
  - Support remote decision making --- e.g. download decision making code into machinery on a factory floor, or into a spacecraft
  - Support in-service, software upgrade --- e.g. download code into TVs to upgrade on-screen displays, or to support new cable services
  - Invent and distribute new protocols and interfaces



## ANSA Scripts and Mobile Agents Workpackage

- **Aim:**
  - To research and promote the evolution of open scripting and agent technologies for our sponsors
- **Two present work tasks:**
  - CORBAgents: CORBA-compliant reference architecture for agents
  - Changeling: an extensible, Safe-Tcl script-based Web server



## CORBAgents: Overview

- **Task purpose:**
  - To produce a prototype system through which we can influence standardisation efforts by the **OMG** on a Reference Architecture for Agents
- **Task description:**
  - A CORBAgent system is a CORBA-compliant system with support for mobile agents
  - These agents communicate through CORBA IDL interfaces, and work within the **OMG's OMA** framework
- **Task deliverables:**
  - An architectural model of a CORBA-based mobile agent system
  - A test implementation of this model



## CORBAgents: Architecture

- **Two main object classes:**
  - Agency
  - Agent



## CORBAgents: Agency

- **A virtual space for Agents:**
  - An execution environment
  - A meeting and interaction environment
- **Agencies do not move**
  - Dependent on the underlying host computer
  - Agency bootstrapping and management is outside the scope of the agent programming environment
- **Standard services**
  - Meeting Service: enabling agents in the same agency to meet
  - Taxi Service: transports agents to other agencies
  - Yellow Pages Service: enables agents to find services
- **Additional services (e.g. Banking Service, Telephone Service)**



## CORBAgents: Agent

- **Agent == data + methods + life**
- **Agents execute concurrently and are persistent**
- **A CORBAgent application consists of a number of Agents moving between, and working within a number of distinctly located Agencies**
- **Agents move between agencies on different nodes**
  - **TaxiService.SendMe(<target agency>, <constraints>)**
- **Meet and interact with agents co-located in the same agency**
  - **objRef = MeetingService.MeetingRequest(<my banking agent>)**
  - **objRef.CreditAccount(<my account number>, <1500 yen>)**





## CORBAgents: Agent Methods

- **An agent is defined by a set of Agent class methods**
  - **Initialise():** used to customise the instance parametrically
  - **Live():** the “main” routine, a codification of the agent’s destiny!
  - **Owner():** returns the responsible human or organisation
  - **Entered():** invoked when agent enters an agency
  - **Leaving():** invoked when agent is leaving an agency
  - **Meeting():** invoked when another agent wants to meet



## **CORBAgents: CORBA Infrastructure**

- **Use ORB for transport**
- **Agents and agencies will be specified in CORBA IDL**
- **Agencies will be coded in any language with an IDL mapping**
- **Agents will be coded in scripting languages with IDL mappings**
- **An agency may support a set of language interpreters**
  - **Agents will be transferred only to target agencies that support their coding language**
- **An agent may leave shadow copies of itself at a previous agency for safety**



## **CORBAgents: Implementation**

- **Implement an agency as a CORBA server object**
- **Use Changeling as a basis for an agency implementation**
- **Use Safe-Tcl as the basis for the agent coding language**
- **Use Web\*'s TcIDii which allows Tcl scripts to invoke CORBA (Iona's Orbix) services via an IDL interface**



## Changeling: Overview

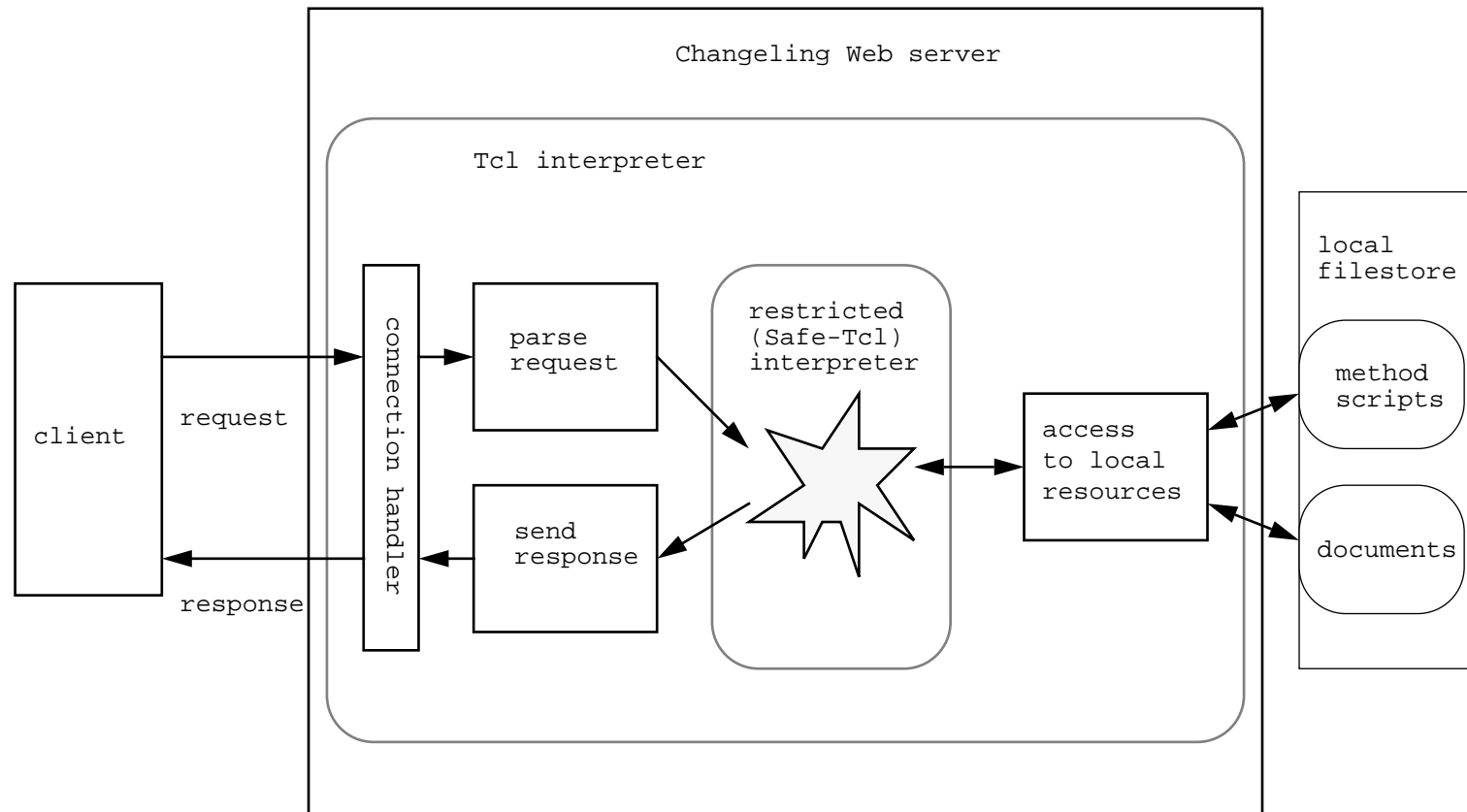
- **Task description:**
  - Build a Safe-Tcl script-based Web server with an extensible set of methods
- **Present situation:**
  - Web servers support only a standard set of methods (GET, POST, HEAD, PUT, etc.)
- **Task purpose:**
  - Learn about Web technology: safe scripting, HTTP
  - Learn about in-service software upgrades
  - No present server implements HTTP/1.0 extension methods
  - Use as a basis for prototyping other agent/Web work



## Changeling: The Key Idea

- **Kernel management methods**
  - **INFO** --- provides information about the method execution environment
  - **INSTALL** --- installs a Safe-Tcl script in the server as a new method
  - **REMOVE** --- removes a method
  - **REGISTER** --- registers a user (allocates method and doc. storage areas)
  - **UNREGISTER** --- unregisters a user (deallocates storage areas)
- **Standard library methods**
  - **GET, POST, HEAD, etc.**
- **User supplied methods**
  - **Installed in the server using the INSTALL method**

# Changeling: Architecture





## Issues

- **Issues for mobile agent research include:**
  - **Safety**
  - **Resource negotiation**
  - **Payment mechanisms**
- **Solutions to some issues (safety and payment) require policy decisions (the Law) and technical answers**



## Issues: Safety

- **Protect receiver from agent --- Possible**
  - Provide only safe, well understood primitives to agents
  - Limit resource consumption (CPU time, memory, file space)
  - Authenticate incoming agents
- **Protect agent from receiver --- Not completely possible**
  - Rely on trusted sites
  - Agents carry minimal necessary rights and sensitive data
  - Protect a data payload by using encryption
  - Leave clones at safe sites to stop kidnapping
- **Protect agents from agents --- Be careful**
  - Must think about emergent behaviour
  - E.g. feature interactions, hostile agents, viruses





## Issues: Resource Negotiation

- **When:** agents transfer between agencies, agents request extra resources, agents using third-party services
- **How:** negotiation protocol
  - need {cpu, memory, interpreter}
  - offer {{cpu, max, cost}, {memory, max, cost}, {interpreter, {Tcl, Java}}}
  - buy {{cpu, amount}, {memory, amount}, {interpreter, Tcl}}
  - present {creditReceipt, amount, bankAccountDetails}
  - sign {creditReceipt, amount, bankAccountDetails, signature}



## Documents

- **APM.1491 “May TC Presentation: Programming the Net”**
- **APM.1453 “The Changeling Web Server”**
- **APM.1473 “Agents for Knowledge Resource Mapping in the World-Wide Web”**