Redesigning and Enhancing the UWAgent Execution Engine

What is UWAgent?

UWAgent is a **Java-based mobile agent execution platform** developed at the UW Bothell Distributed Systems Laboratory (DSL). It is used as the infrastructure for the DSL's AgentTeamwork **grid computing** middleware system. The goal of this redesign and enhancement project was to make UWAgent more useful for AgentTeamwork, and to make the code more maintainable.

Project Accomplishments

- Replaced **Java RMI** (Remote Method Invocation) with **Java sockets** for agent navigation and communication
- Implemented three new features:
  - Navigation over **gateways**
  - **Monitor** commands
  - **Secure** Communication
- Tested behavior of UWAgent system with **identically-named classes** at the same UWPlace
- **Refactored** existing code base for consistency and ease of maintenance

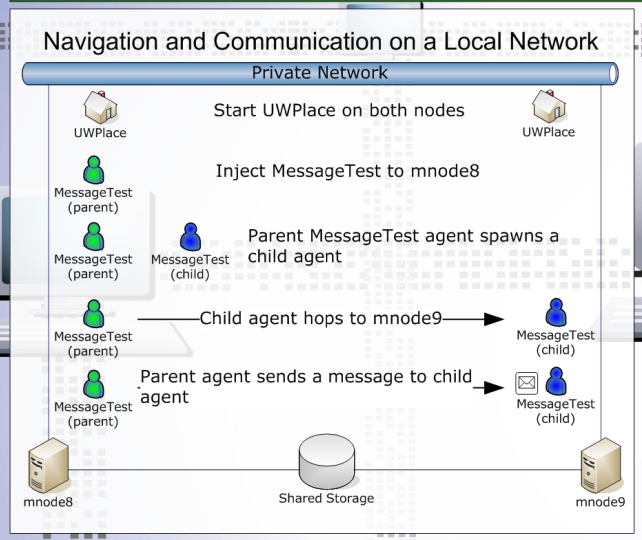
Why Use Mobile Agents?

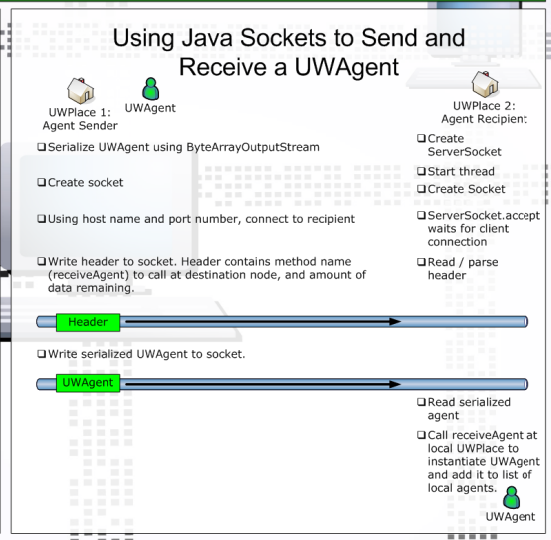
The UWAgent system is an example of a **mobile agent** system. Researchers Danny B. Lange and Mitsuru Oshima give seven reasons for using mobile agents:

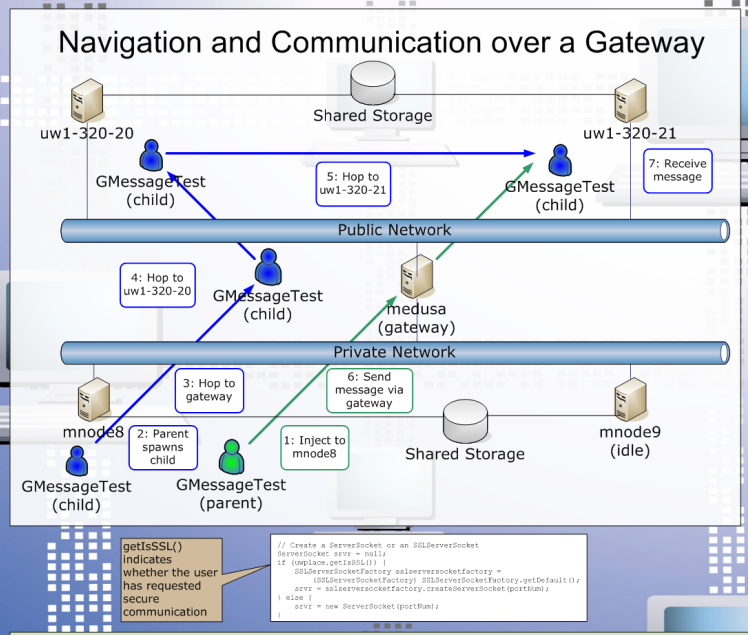
1. They **reduce network load** because processing takes place locally.
2. They **overcome network latency** for the same reason.
3. They **encapsulate protocols**, freeing hosts from the responsibility for implementing them.
4. They **execute asynchronously and autonomously**.
5. They **adapt dynamically** to changing conditions on their hosts.
6. They are **naturally heterogeneous** because they depend only on their execution environment, not the underlying machine.
7. They are **robust and fault-tolerant** because of their autonomy and their ability to adapt to changing host conditions.

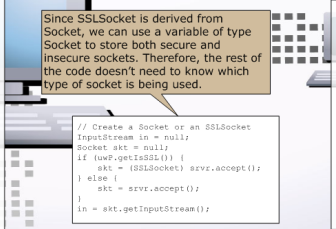
D. Lange and M. Oshima. Seven good reasons for mobile agents. Communications of the Association of Computer Machinery, 42:88–89, March 1999.

Agent Navigation and Communication









Why not use RMI?

RMI (**Remote Method Invocation**) is standard Java technology, but it has several drawbacks when used with UWAgent:

- Users must **manually** start and stop the `rmiregistry` process.
- RMI introduces a **communication layer** that must be configured.
- When the RMI client is on a gateway and its RMI server is on a private network, the server may need to establish a **reverse connection** to its client. Sometimes, the client may provide it with a **public IP address**, which the server cannot use.

Monitor Commands

The following UWAgent commands allow management of running agents:

- **as** (Agent Status): report status (Ready, Running, or Suspended) for each agent
- **kill**: terminate an agent
- **suspend**: pause an agent's execution
- **resume**: resume a paused agent

Class Name Collision

When an agent hops to another node, it can carry **additional classes** with it. A **name collision** could occur if another agent on the same node uses a class with the same name. Testing confirmed that UWAgent is not affected by this problem. Another mobile agent system, **IBM Aglets**, is affected because it uses **caching**. It solves the problem by using **multiple class loaders** that isolate the identically-named classes from each other.

Secure Communication

UWAgent implements **secure communication** as follows:

- A UWPlace **command-line option** allows the user to turn security support on or off.
- When security support is on, the **SSLServerSocket** and **SSLSocket** classes are used instead of the **ServerSocket** and **Socket** classes. Because the secure classes are **derived** from their non-secure counterparts, the amount of new code required for security support is minimal.
- UWPlace and UWInject must include a reference to a **certificate**.

Duncan Smith
duncans@u.washington.edu

Dr. Munehiro Fukuda
mfukuda@u.washington.edu

University of Washington, Bothell
Computing and Software Systems
CSS 497 Autumn Colloquium
Friday, December 9th, 2005