Title: Software Agent Technology in the Laboratory

Author(s): Torsten A. Staab
ESA-AET, Automation & Robotics Team
Los Alamos National Laboratory, Los Alamos, NM, USA

Submitted to: LabMation'2002 Conference
Philadelphia, PA, USA
October 2002
Title: Software Agent Technology in the Laboratory

Author: Torsten Staab
Los Alamos National Laboratory
ESA-AET, Automation & Robotics Team
Mailstop J580
Los Alamos, NM 87545
Phone: (505) 665-7345 E-Mail: tstaab@lanl.gov

Abstract

The IT (Information Technology) environment in today's laboratories is characterized as being highly distributed, heterogeneous, and in some instances extremely dynamic. Larger organizations have to deal with hundreds of different systems, ranging from stand-alone workstations and devices in laboratories to fully integrated LIMS (Laboratory Information Management System) and ERP (Enterprise Resource Planning) systems. An information system operating in such an environment must handle several emerging problems, such as heterogeneous hardware and software platforms, as well as distributed information sources and capabilities. It is also expected that the IT infrastructure scales well, easily integrates with legacy systems, allows resource sharing, and supports day-to-day operations such as information retrieval, data storage, validation, tracking, replication, and archival in a fully automated fashion.

By using real-world examples, this presentation will illustrate how software agent technology can be used to manage the ever increasing IT complexity and user demands in the laboratory of the future.
Abstract

The IT (Information Technology) environment in today's laboratories is characterized as being highly distributed, heterogeneous, and in some instances extremely dynamic. Larger organizations have to deal with hundreds of different systems, ranging from stand-alone workstations and devices in laboratories to fully integrated LIMS (Laboratory Information Management System) and ERP (Enterprise Resource Planning) systems. An information system operating in such an environment must handle several emerging problems, such as heterogeneous hardware and software platforms, as well as distributed information sources and capabilities. It is also expected that the IT infrastructure scales well, easily integrates with legacy systems, allows resource sharing, and supports day-to-day operations such as information retrieval, data storage, validation, tracking, replication, and archival in a fully automated fashion.

By using real-world examples, this presentation will illustrate how software agent technology can be used to manage the ever increasing IT complexity and user demands in the laboratory of the future.
Software Agent Technology in the Laboratory

Torsten A. Staab

LabMation 2002 Conference
Philadelphia, PA, USA
October 1st, 2002
Overview

- Software Agents Origins & Definitions
- Agent Technology Applications & Users
- Agent Typology
- Requirements for Agent-based Systems
- Agent Architecture & Languages
- Collaborative Agent Technology in the Lab
- Issues with Agent Technologies
- Conclusions
Origins of Software Agent

Software agent technology can be traced back to Hewitt's Concurrent Actor Model (1977)

Actor: a self-contained, concurrently executing object with encapsulated internal state, capable of responding to messages from other similar objects
Definitions

❖ **Agent:**
One that is authorized to act for another. Agents possess the characteristics of *delegacy, competency, and amenability.*

Examples of human agents: booking agents, sales agents, politicians.

❖ **Software Agent:**
An artificial agent that operates in a software environment.

❖ **Intelligent Software Agent:**
A software agent that uses Artificial Intelligence (AI) in the pursuit of the goals of its clients.
Agent Technology Applications & Users

♦ Applications:
Workflow Management, Network Management, Air-traffic Control, Business Process Re-engineering, Data Mining, Information Retrieval/Management, Electronic Commerce, Education, Personal Digital Assistants (PDAs), E-mail, Digital Libraries, Command and Control, Smart Databases, Scheduling...

♦ Users:
- Alcatel, Apple, AT&T, BT, Daimler-Benz, DEC, HP, IBM, Lotus, Microsoft, Oracle, Sharp, Reuters, Dow Jones, ...

Los Alamos National Laboratory
Agent Typology

- Mobile Agents
- Information Agents
- Reactive Agents
- Hybrid Agents
- Smart Agents

Collaborative Agents

Aggregate Agent Systems
Requirements for Agent-based Systems

There are several levels at which agent-based systems must agree, at least in their interfaces, in order to successfully interoperate. These levels include the following:

- Transport:
  - how agents send and receive messages

- Language:
  - what the individual messages mean (syntax + semantics)

- Policy:
  - how agents structure conversations

- Architecture:
  - how to connect systems in accordance with constituent protocols
Agent Architecture

Other Agents

Communications

Messages

Interaction

Goals

Interact

Motivation

Goals

Control

Action

Actions

Meta Data

Meta Storage*

*Virtual Knowledge Base (VKB)
KQML (Knowledge Query & Manipulation Language)

- is both a message format and a message-handling protocol to support run-time knowledge sharing among agents.
- most widely used ACL
- based on Speech Act Theory
  - Illocution (Purpose of Message)
  - Context of Message
  - Content of Message
- KQML messages are called Performatives
- KQML Example:
  Sender: Mettler-Toledo AX504
  Receiver: LECIS Controller
  In Reply To: ID9100.5
  Ontology: BL.SW
  Language: Prolog
  Content: "Weight(S1234, 20, mg)"
Agent Communication Languages (ACL)

KQML Performatives

achieve          evaluate           stream-about
advertise        forward            stream-all
ask-about        generator          subscribe
talk             insert             tell
ask-all           monitor            transport-address
ask-if            next
ask-one           pipe
break             ready
broadcast        recommend-all
broker-all        recommend-one
broker-one
deny              recruit-all
delete            recruit-one
delete-all        register
delete-one        reply
discard           rest
eos               sorry
error             standby
Collaborative Agent Technology—Why?

- Information available can be unorganized, multi-modeled, and distributed

- Number and variety of data sources and services is dramatically increasing and constantly changing

- Same piece of information can be accessible from a variety of different information sources

- Information can be ambiguous, erroneous, or obsolete

- Enhance modularity (which reduces complexity), speed (due to parallelism), reliability (due to redundancy), flexibility
Collaborative Agent Technology—Architecture

Application Layer

App 1       ...       App n

Task Layer

Task Agent   ...   Task Agent

Information Layer

Info Agent   ...   Info Agent

DB
Collaborative Agent Technology—Implementation

Example 1: Agent-based Information Retrieval
Collaborative Agent Technology—Implementation

Example 2: Device Control

System Integration

- Tecan FE500 Liquid Handler
- AB 7700 Sequence Detector
- Beckman ORCA Plate Handling Robot
- Device Controller & Scheduler

Network

UNIX

LIMS
Collaborative Agent Technology—Implementation

SOAP-based (Simple Object Access Protocol) Agents

What is SOAP?

- platform independent XML-based (eXtensible Markup Language) communication protocol
- transport protocol independent (runs on top of HTTP, SMTP, ...)
- created by Ariba Inc., Compaq, HP, IBM, IONA, Lotus, Microsoft, SAP AG, Userland, and others
- not intended to replace CORBA, DCOM, RMI
Collaborative Agent Technology—Implementation

Example of a SOAP Agent Request Message:

```xml
<SOAP-ENV:Envelope
   xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
   SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/>
   <SOAP-ENV:Header>
   <t:Transaction xmlns:t="www.lanl.gov"
   SOAP-ENV:mustUnderstand="1">LECIS</t:Transaction>
   </SOAP-ENV:Header>
   <SOAP-ENV:Body>
   <m:SETUP xmlns:m="www.lanl.gov/B-Division">
   <param>UNIT=mg</param>
   </m:SETUP>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
Collaborative Agent Technology—Implementation

Example 2: Agent-based Device Control

SIA: SOAP-based Interface Agent

Network

Liquid Handler

AB 7700 Sequence Detector

SIA Registry

Beckman ORCA

Device Controller & Scheduler

Los Alamos National Laboratory
Collaborative Agent Technology—Implementation

Example 3: Agent-based Automated Compound Retrieval

Images courtesy of K-Team S.A. (Koala Robot) and Remp AG (Plate Storage System), Switzerland
Issues w/ Agent Technologies

- Security
- Responsibilities
- Ownership
- Legal Issues
- Ethical Issues
Conclusions

Agent Technology...

- is a powerful tool to manage system complexity
- is domain & application-independent
- fosters resource sharing and reuse
- scales well
- integrates easily with legacy systems
- can be implemented using standard protocols (RMI/RPC, SOAP, HTTP, SMTP, ...), middleware (CORBA, COM+, JINI, ...), and programming languages (Java, TCL, C++, ...)