Use of Software Agents in Enterprise Integration

Bill Chu

Department of Computer Science
School of Information Technology
UNC Charlotte

Dec 1999
What is a software agent?

• No consensus yet, but several key concepts are important to this emerging paradigm.

• A software agent is a program that assists people and acts on their behalf. People can delegate tasks and work to agents.

• A software agent:
  – is an autonomous, goal-directed process
  – is situated in, is aware of, reacts to and adapts to its environment
  – cooperates with other agents (software or human) to accomplish its tasks
A “wish list” for Application Integration

• Plug and play with little static set up
• Adaptive interfaces (adaptive, reactive)
• Negotiating ontologies (autonomous, goal oriented, cooperative)
  – Structural mapping
  – Value mapping
• Handle errors, “self healing”, (autonomous, goal oriented)
CIIMPLEX
EECOMS

- **Funder:** National Institute of Standards and Technology / Advanced Technology Program
  - Technologies for the Integration of Manufacturing Applications (TIMA)
  - ~ $52M over 5 years in two ATP projects
- **Plug and Play framework for application integration** for MES, ERP, Finite Scheduling, and Capacity Analysis/Decision Support
- **Integration for multi-company supply chain management**
- **Objectives:** interoperability, configurability, adaptability, extensibility, plug and play.

Consortium for Integrated Intelligent Manufacturing Planning and Execution
Extended Enterprise Coalition for Integrated Collaborative Manufacturing Systems

**Participants**
- IBM Corp
- Universities
  - University of Maryland Baltimore County
  - University of North Carolina at Charlotte
  - University of Florida
- Baan USA Ltd.
- Boeing
- TRW
- Indx
- Ingersoll-Rand Co.
  - Demand Solutions
  - DLoG Remex Inc.
- Intercim
- EnvisionIt Software
- Vitria

Dec. 1999
© Bei-Tseng Chu
CIIMPLEX architecture

Source
Applications
Adapter - BUI Build
Scenario Build

Target
Applications
API
Adapter
Service calls
BOD's

Activity Manager

Environment Creation Tools

Broker

Services

BOD's Service calls

map repos

EMMA
security logging unique routing trans rules

pub/sub
CIIMPLEX Architecture

• Asynchronous, guaranteed message delivery based on MOM
• Modes of message:
  – asynchronous send
  – send/receive
  – send with acknowledgement to a third party
• All messages logged with a unique sequence number
EECOMS Architecture

Enterprise 1

EECOMS Enabled Application

Broker

EECOMS Security Technology

Enterprise 2

EECOMS Security Technology

Broker

EECOMS Enabled Application

email

https

Dec. 1999

© Bei-Tseng Chu
Data exchange

• Open Application Group (OAG)
  – Largest ERP vendors
  – Business Object Document (BOD)
  – XML encoding

• All applications will generate triggers that produced BODs.
<?DOCTYPE PROCESS_RFQ_001 SYSTEM "E2_process_rfq_001.dtd">
<PROCESS_RFQ_001>
  <CONTROLAREA>
    <BSR>
      <VERB>PROCESS</VERB>
      <NOUN>RFQ</NOUN>
      <REVISION>001</REVISION>
    </BSR>
    <SENDER>
      <LOGICALID>Boeing</LOGICALID>
      <COMPONENT>PURCHASING</COMPONENT>
      <TASK>PROCESSRFQ</TASK>
      <REFERENCEID>COMPREHENSIVE TEST CASE</REFERENCEID>
      <CONFIRMATION>0</CONFIRMATION>
      <LANGUAGE>EN</LANGUAGE>
      <CODEPAGE>CP000111</CODEPAGE>
      <AUTHID>JIAN&RONGMING</AUTHID>
    </SENDER>
    <DATETIME qualifier="CREATION">
      <YEAR>1999</YEAR>
      <MONTH>10</MONTH>
      <DAY>31</DAY>
      <HOUR/>
      <MINUTE/>
      <SECOND/>
      <SUBSECOND/>
      <TIMEZONE/>
    </DATETIME>
  </CONTROLAREA>
  <NOTE>
    This is a test note.
  </NOTE>
</PROCESS_RFQ_001>
<RFQHEADER>
    <RFQID>RFQ_1</RFQID>
    <DESCRIPTION/>
    <DATETIME qualifier="EXPIRATION">
        <YEAR>1999</YEAR>
        <MONTH>10</MONTH>
        <DAY>31</DAY>
        <HOUR/>
        <MINUTE/>
        <SECOND/>
        <SUBSECOND/>
        <TIMEZONE/>
    </DATETIME>
    <PARTNER>
        <NAME>BoeingHeaderQuarter Inc. </NAME>
        <ONETIME>0</ONETIME>
        <PARTNRID>BoeingHeaderQuarter</PARTNRID>
        <PARTNRTYPE>SOLDTO</PARTNRTYPE>
    </PARTNER>
    <PARTNER>
        <NAME>Supplier_4 Inc. </NAME>
        <ONETIME>0</ONETIME>
        <PARTNRID>Supplier_4</PARTNRID>
        <PARTNRTYPE>SUPPLIER</PARTNRTYPE>
    </PARTNER>
</RFQHEADER>
Practical approaches to agent application

- “Green field agents”
  - build distributed agent-based systems from scratch
- Agents in integration
  - find “white space” between applications
  - add value by concentration on integration/interoperation problems
A basic agent pattern

1. Advertise
2. Recommend-one
3. Tell
4. Subscribe
5. Tell
A KQML Message

• Represents a single *speech act* or *performative*
  ask       reply       forward
  tell      deny        subscribe
  achieve   monitor     evaluate...
• with an associated *semantics and protocol*
• with a list of *attribute/value pairs*, including
  :content   :receiver  :to
  :language  :reply-with :in-reply-to
  :topic     :sender     :label
  :ontology  :from      :aspect ...
A KQML Message

tell
:sender Gen-1
:receiver Gen-db
:in-reply-to id1
:ontology Genealogy
:language Prolog
:content “father(John,Eve)”
A Performative Landscape

Ask, ask-if, ask-in, ask-one, ask-all, evaluate

Ask, ask-if, ask-in, ask-one, ask-all, evaluate

Inform, tell, untell, deny, confirm, disconfirm

achieve, unachieve, request

Basic

Inform

Query

Request

Basic

Services

ACL

Performatives

Query

Reply

Reply

 OnTriggerEnter

Broker, recruit, recommend, subscribe, monitor,

advertise, import, export

Basic

Stream

standby, ready, next, rest, discard, generator, stream-in, stream-all

stream-in, stream-all

Comm

register, unregister, forward, broadcast, route

Dec. 1999

© Bei-Tseng Chu
Jackal

Jackal is a Java package which allows applications written in Java to communicate via KQML. Its features include:

- Communication using KQML
- Built-in support for KIF content language
- Multiple agents in one Java VM
- Multiple transport protocols (tcp/ip, http or smtp)
- Enforced conversation policies based on KQML semantics
- Presents programmer with a blackboard-style interface to the message flow, for maximum flexibility.
- Blocking and non-blocking message-waiting protocols.
- Flexible agent naming scheme.
- Built-in support for basic services, including ANS (agent name server), ACS (agent control server), and (soon) ASS (agent security server).
Conversation Templates

Conversation templates (specified by URL) define shared models of conversation protocols. Loaded at runtime. Example:

// Conversation Template: Initial and accepting states all caps,
// other states initial caps, arc-labels lower case.
(conversation
  (name kqml-ask-one)
  (author "R. Scott Cost")
  (date "3/4/98")
  (start-state START)
  (accepting-states TOLD)
(transitions
  (arc (label ask-one) (from START) (to Asked) (match "(ask-one)"))
  (arc (label tell) (from Asked) (to TOLD) (match "(tell)")
  (arc (label deny) (from Asked) (to TOLD) (match "(deny)")
  (arc (label untell) (from Asked) (to TOLD) (match "(untell)")
  (arc (label sorry) (from Asked) (to TOLD) (match "(sorry)")
  (arc (label error) (from Asked) (to TOLD) (match "(error)"))))

Dec. 1999
© Bei-Tseng Chu
Jackal Architecture

- The distributor provides simple **blackboard facility** interfacing Jackal modules and agent logic.
- Jackal modules and agent application logic post messages (incoming and outgoing) into the distributor and also have threads which *attend* for posted messages.
- Most uses of attend fall into two categories:
  - listen for the single response to a message that was sent, or
  - listen constantly for a certain type of message -- constitutes a service
- This is where security-related messages will be picked up by appropriate plug-in module.
Registration

- Legacy ERP
- Legacy MES

Advertise: I own manufacturing items

- BOD Registry Agent
  - Advertise: I own manufacturing WIP status
Initial synchronization

1. I want to initialize Items

2. Who owns items?

3. ERP

4. Initialize MES with Items from ERP

5. Get all Items

- BOD Registry Agent
- Application Integration Assistant
- Initialization Management Agent (IMMA)
- Legacy ERP
- Legacy MES
Exception Management Agent

1. New schedule
2. Item unknown: 123
3. Who owns Items
4. ERP
5. Please send item 123 to MES
6. Item 123…, please ack. EMMA
7. Replays erred BOD

Legacy Scheduling System
BOD Registry Agent
EMMA
Legacy ERP
Legacy MES

Dec. 1999
© Bei-Tseng Chu
CIIMPLEX architecture summary

- Plug and play with little static set up
- Adaptive interfaces (adaptive, reactive)
- Negotiating ontologies (autonomous, goal oriented, cooperative)
  - Structural mapping
  - Value mapping
- Handle errors, “self healing”, (autonomous, goal oriented)
- Registration / IMMA minimizes set up
- Through agents, an app can adapt to an environment
- Assume shared ontology:
  - BODs
  - Mapping and mediation service
- EMMA
Schedule validation analysis

• Production reality will eventually invalidate schedules
• Current scheduling tools do not address the need to validate a schedule against execution reality
Process rate calculation

1. Who owns WIP status?
2. MES
3. Subscribe to WIP status
4. WIP status
5. WIP status
6. Signal for reschedule

Legacy MES

CIIMPLEX Broker

BOD Registry Agent

Sensitivity Analysis
Statistical Throughput Control

STC Chart for large perturbation

Scheduling Horizon (days)

Shortage/Overage of machine use (days)

-1

-0.5

0

0.5

1

4 7 10 13 16 19 22 25 28

5%

95%

Nt-St

© Bei-Tseng Chu
Supply Chain and B2B EC

- A supply chain is a collection of enterprises that coordinate with each other to satisfy customer demands
- **Replace inventory with real-time information**
- Supply portals
  - Ford’s supply net
  - www.SupplyView.com, a market place for electronic components
- Supply portals creates efficient electronic market places where procurements can be decided quickly, reducing the need of inventory
Supply Portal

- Specialized network portal for a vertical market place
- Brokering service
  - Buyer and seller registration
  - Background verification (optional)
  - Call for proposals by buyers
  - Offers by sellers
- Additional services
  - Order selection
  - Auction
Order Selection

• A buyer selects a set of offers from suppliers to fulfill an order
• Driven by business rules that can be dynamically changed
• Can be performed by a software agent representing the buyer either at the portal site or at the buyer’s site
Core business process for supply chain

Buyer

Demand data: orders and forecasts

Request for proposal (quote)

Proposal

Request for proposal (quote)

Seller

Proposal

Seller

Proposal (order)

Buyer

Demand data: orders and forecasts

Counter Proposal

Seller
An agent approach to supplier portal

• Buyers and sellers represented as agents
• The supplier portal as a brokering agent
• The buyer agent has supplier selection logic based on constrain satisfaction principle
CSP as a model for the core supply chain negotiation process

- Possible Terms and conditions
- Offer
- Acceptable offer
- Unacceptable offer
- Negotiating counter offer
- CSP variables and their domains
- CSP value assignments
- Value assignments satisfying constraints
- Value assignments fail some constraints
- Relax constraints to make an unacceptable offer acceptable.
CSP Model

Variables

- Supplier
- Supplier lead time
- Supply transportation time
- Order (total needed by customer)
  - suborder (one per supplier)
  - release time
  - shipping date

Discrete Constraints

- Quantity conservation:
  - Suborders add to total order
- Temporal consistency
  - a supplier’s known capacity must not be less than the committed orders within the manufacturing lead time of the orders
- Shipping feasibility
  - shipping date + transportation time must coincide with due date

Objective functions

- Purchasing cost
- Quality
- Risk of supplier failing to meet due dates
Prototype Implementation of a buyer agent

```
Demand Data
  ↓
Dynamic Lot Sizing
  ↓
Selection via CSP
  ↓
Add Constraints
  ↓
reject
  ↓
reject counter
  ↓
accept
  ↓
accept counter
  ↓
RFP (quote)
  ↓
Suppliers
  ↓
Acknowledge
```
Dimensions of selection problems

• Dependency among orders
  – independent (buyer) orders
    • No constraints among different orders
      – due date constraint
      – quantity constraint
  – dependent (buyer) orders
    • Constraint among different (buyer orders)
      – Kitted
      – Temporal constraint due to assembly sequence
Dimensions of the selection problem

• Supplier Capacity constraints
  – No capacity constraints
    • Each order can be satisfied by a single vendor
  – With capacity constraints
    • An order need to be split and be fulfilled by multiple vendors.
Dimensions of the selection problem

• Negotiation
  – Constraint relaxation (e.g. due date, quantity, price)
Dimensions of selection: negotiation performatives

<table>
<thead>
<tr>
<th>accept-proposal</th>
<th>the action of accepting a previously submitted proposal to perform an action</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CFP</strong> proposal</td>
<td>the action of calling for proposals to perform a given action</td>
</tr>
<tr>
<td>reject-proposal</td>
<td>the action of rejecting a proposal to perform some acting during a negotiation</td>
</tr>
<tr>
<td>terminate</td>
<td>the action to finish the negotiation process</td>
</tr>
</tbody>
</table>
Motivation for Distributed Trust Management

- Centralized security management of information resources does not scale for supply chain environments characterized by dynamic and collaborative business relationships
- ACL-based access control does not offer scalable authorization capabilities
- Delegation of authorities is a common business practice
  - effective management of dynamic and collaborative business relationships
  - formalizing security policies
  - digital signatures and audit trail
- Use Public Key Infrastructure (PKI) and digital certificates to achieve distributed trust management
Authorization certificate

• Authorization certificates will be based on X.509 format
  – Optional field will be used to contain authorization information

• Types of authorization certificate
  – Role Authorization Certificate: asserting that a subject plays a given role (e.g., Company P is an Engine supplier)
  – Direct Authorization Certificate: giving direct authorization to a subject to access some resource
  – Delegation Authorization Certificate: giving authorization to a subject to delegate access to other subjects.

• The serial number of each authorization certificate: each authorization certificate has a distinguished serial number
Example Scenario of DTM:

Step 1: Company B and T exchanges certificates with P

“P is an Engine supplier, signed B.”
“P can read the specification table at B, signed B”
“P can read and write price table for P at B, signed B
P can delegate access to both tables within P, signed B”

“P is an Engine supplier, signed T.”
“P can read the specification table at T, signed T,
P can delegate access to both tables within P, signed T”
Marty is delegated the right to read B’s specification table

“I delegate Marty to read B’s specification table, signed manager”

Request a certificate to read B’s specification table, here are the signed documents….

Trust Establishment Agent

“Marty can read B’s specification table, signed TEA-P”
Marty queries B’s specification table

Select * from B.specification

“Marty can read B’s specification table, signed TEA-P”

“P is an Engine supplier, signed B.”
“P can delegate access to both tables within P, signed B”

Oracle DB, role based access control

Dec. 1999

© Bei-Tseng Chu
A distributed trust management architecture for supply chains

User Agent → Access Control Agent → Internet → Access Control Agent → Information Resource

- User/ Application
- DTM
- Policy Database
- TEA
- DTM
- Policy Database
- TEA
- resource

Dec. 1999
© Bei-Tseng Chu
Request

• **request1 (Marty) consists of**
  – user’s query/update request (e.g. Marty requests to read company B’s engine specification, signed Marty)

• **request2 (Marty’s user agent) consist of**
  – user’s request (e.g. Marty requests to read company B’s engine specification, signed Marty)
  – user’s appropriate authorization certificate (e.g. Marty is a design engineer, signed company P)

• **request3 (Company P’s Access control agent) consist of**
  – user’s query/update request (e.g. Marty requests to read company B’s engine specification, signed Marty)
  – company P’s certificate for the user’s authority (e.g. Marty can read company B’s engine specification, signed company P)
  – company’s appropriate authorization certificate issued by the resource owner (company P can read and delegate access to the specification database, signed company B)
Components

- **User Agent:**
  - It holds the user’s private key and can sign requests upon the request of the user.
  - Manage the user’s authorization certificates

- **DTM: A system used by a company to achieve distributed trust management. It consists of:**
  - **TEA (Trust Establishment Agent):**
    - holds company or resource owner’s private key
    - issue authorization certificates
    - manage certificate revocation list (CRL)
  - **ACA (Access Control Agent)** use access control policies to verify and approve
    - user requests for resources outside of the company
    - requests for local resource access
  - **Policy Database:** used to store the policies
    - Access Policies: Role-based access policy, Separation of duty policy, Chinese wall policy
Contributions

• Enterprise integration architecture
  – integrating mainstream technology and agent technology
  – demonstrate the advantage of incorporation agent-based approach

• A distributed trust management architecture for electronic enterprises
  – scalable authorization
  – for supply chain management
  – for collaborative systems

• Identified a number of key techniques to integrate manufacturing planning, scheduling and execution

• Developed a constraint-based model for buyer selection for supply chain management
On going activities

• Distributed trust management
• Constraint-based buyer selection
  – Complexity and heuristics
  – Interlingua
  – Theory of negotiation
Publications


Dec. 1999 © Bei-Tseng Chu
Publications

Other References


