Software Agent Technology

Introduction to Technology

Heimo Laamanen

What is an Agent?
- Attributes
  - Autonomy
  - Proactive
  - Goal-oriented
  - Mobile
  - Temporal continuity
  - Adaptivity
  - Communicative
  - Collaborative

Agents do it for money...

Objects do it, because they are commanded to...
Environment of Software Agents

Platform

- A place where agents live
  - not always needed
- Agent management
  - creation – termination
  - security
- Agent communication services
- Agent directory services
Introduction to Technology

Platforms - FIPA Implementations

- Jade (TiLab - Telecom Italy)
- FIPA-OS (Nortel - Emorphia)
- LEAP (EU -project)
- BlueJade (HP)
- MicroFIPA-OS (HY)

Questions? Comments. Objections!
Introduction to Technology

- Agent Communications
  - Modelled at higher abstraction level than ‘traditional’ data communications.
  - Speech Act theory
    - J.R. Searle (1969) *Speech Acts*
    - Classic example:
      - President says "I declare war" is actually an action.

Agent Communication Language (ACL)

- KQML
  (Knowledge Query and Manipulation Language)
- FIPA-ACL
  (Agent Communication Language)
Agent Communication Language

- FIPA-ACL

Communicative acts
- Propagate
- Propose
- Proxy
- Query-if
- Query-ref
- Refuse
- Reject-proposal
- Request
- Request-when
- Request-whenever
- Subscribe

Parameters
- :sender
- :receiver
- :content
- :reply-with
- :reply-by
- :in-reply-to
- :envelope
- :language
- :ontology
- :protocol
- :conversation-id

Example of request

The sender requests the receiver to perform some action.

(request
  :sender i
  :receiver j
  :content "open \"db.txt\" for input"
  :language vb)
Introduction to Technology

Semantic of Request

\(<i, \text{request}(j, a)\>
FP: \(FP(a) [i]j \land B_i \text{Agent}(j, a) \land \neg B_i \text{PGj} \text{Done}(a)\)
RE: \(\text{Done}(a)\)

\(i = \text{sender}, j = \text{receiver}, a = \text{action to perform}\)
\(\text{Done}(a) = \text{action performed}\)
\(B_i \text{Agent}(j, a) = \text{agent } i \text{ believes that } j \text{ can perform } a\)
\(\neg B_i \text{PGj} \text{Done}(a) = i \text{ does not believe that } \text{Done}(a) \text{ is } j \text{'s persistent goal}\)

Introduction to Technology

Content languages

- FIPA-SL
  - subsets SL0, SL1, and SL2
- FIPA-RDF0
  - RDF based content language
- FIPA-CCL
  - Constraint Choice Language
- KIF & FIPA-KIF

Introduction to Technology

Content languages

- KIF (Knowledge Interchange Format)

\((\text{salary 015-46-3946 widgets 72000})\)
\((\text{salary 026-40-9152 grommets 36000})\)
\((\text{salary 415-32-4707 fidgets 42000})\)

\(> (* (\text{width chip1}) (\text{length chip1}))\)
\(> (* (\text{width chip2}) (\text{length chip2}))\)

Introduction to Technology

Content languages

- FIPA-SL
  - Formal language used to define the semantics of the FIPA ACL
  - Logical propositions are expressed in a logic of mental attitudes and actions, formalised in a first order modal language with identity
Introduction to Technology

Content language

FIPA-SL example

```
(query-ref
 :sender (agent-identifier :name B)
 :receiver (set (agent-identifier :name A))
 :content
   (iota ?x (UKPrimeMinister ?x))
 :language FIPA-SL
 :reply-with query2)
```

```
(inform
 :sender (agent-identifier :name A)
 :receiver (set (agent-identifier :name B))
 :content
   (= (iota ?x (UKPrimeMinister ?x))
       "Tony Blair")
 :language FIPA-SL
 :in-reply-to query2)
```

Interaction protocols

Specify agent communication patterns.

Example:

```
FIPA-query-Protocol

Initiator

Participant

query
refuse
not-understood
failure
inform

Initiator

Participant
```

FIPA interaction protocols

- Request
- Query
- Request When
- Contract Net
- Iterated Contract Net
- English Auction
- Dutch Auction
- Brokering
- Subscribe
- Brokering
- Propose

Questions?

Comments.

Objections!
Ontology

The shared ontology can be **implicit** or **explicit**.

- Implicit ontology are typically represented only by procedures.
- Explicit ontology are (ideally) given a declarative representation in a well defined knowledge representation language.

**Ontology**

A common vocabulary and agreed upon meanings to describe a subject domain.

An ontology is a specification of a conceptualization.

An ontology is a description of the concepts and relationships that can exist for an agent or a community of agents.

**Example**

```
Ontology O1

fruit

apple  lemon  orange
```
Environments of software agents - another viewpoint

- Accessible vs inaccessible
  - An accessible environment is one in which the agent can obtain complete, accurate, up-to-date information about the environment's state.
  - Most moderately complex environments (including, for example, the everyday physical world and the Internet) are inaccessible.
  - The more accessible an environment is, the simpler it is to build agents to operate in it.

Deterministic vs non-deterministic

- A deterministic environment is one in which any action has a single guaranteed effect — there is no uncertainty about the state that will result from performing an action.
- The physical world can to all intents and purposes be regarded as non-deterministic.
- Non-deterministic environments present greater problems for the agent designer.

Episodic vs non-episodic

- In an episodic environment, the performance of an agent is dependent on a number of discrete episodes, with no link between the performance of an agent in different scenarios.
- Episodic environments are simpler from the agent developer's perspective because the agent can decide what action to perform based only on the current episode — it need not reason about the interactions between this and future episodes.
Environments of software agents

Static vs dynamic

- A static environment is one that can be assumed to remain unchanged except by the performance of actions by the agent.
- A dynamic environment is one that has other processes operating on it, and which hence changes in ways beyond the agent's control.
- The physical world is a highly dynamic environment.

Discrete vs continuous

- An environment is discrete if there are a fixed, finite number of actions and percepts in it.
Visions

Possible Applications
- Ambient intelligence
  - Seamless delivery of services and applications
    - Ubiquitous computing, ubiquitous communication and intelligent user interfaces.
- Bioinformatics and computational biology
  - Simulation modelling of biological systems.
- Grid computing
  - Virtual organisations
- Electronic business
  - Negotiating deals and making purchases.
- Simulations
  - Education and training
  - Entertainment

Dream about human-like services

Near-Term Future (c. 2003—2005)
- Systems are designed by same teams.
- Systems share common domain knowledge.
- Systems will increasingly be designed to cross corporate boundaries.
- Participating agents have fewer goals in common.
- Agent communication languages are based on standards.
Visions

Medium-Term Future (c. 2006-2008)
- Multi-agent systems will permit participation of heterogeneous agents designed by different teams.
- Any agent will be able to participate in these systems, provided their behaviour conforms to requirements.
- Systems will still be specific to particular application domains.
- Languages and protocols will be agreed and standardized.
- Systems will scale to large numbers of participants.

Open systems in specific domains.
- Bridge agents capable to translate between separate domains.
- Participation by any agent able to satisfy publicly-advertised standards.
- Agreed protocols and languages.
- Use of standard agent-specific design methodologies.

Longer-term future (c. 2009 onwards)
- Truly open and fully-scalable agent systems developed by diverse design terms
- Agents learn appropriate protocols and behaviour upon entry to system
- Languages, protocols, and behaviours emerge from actual agent interactions.
- Evolving organisational structure with multiple, dynamic, interacting organisations.
- Self-modifying agent communications languages.

Challenges
- Increase quality of agent software to industrial standard
  - Agent oriented design methodologies, tools and development environments.
  - Seamless integration with existing technologies.
- Provide effective agreed standards
  - Agent communication languages
  - Interaction protocols
  - Multi-agent architectures
Visions

Challenges

- Provide semantic infrastructure for open agent communities
  - Ontologies
  - Semantic Web
  - Matchmaking and broker architectures
- Develop reasoning capabilities for agents
  - Negotiation algorithms
  - Planning and BDI architectures
  - Ontological reasoning

Visions

Challenges

- Develop agent ability to understand user requirements
  - User profiling
  - Personalisation
  - Utility modelling
  - Knowledge acquisition tools

Visions

Challenges

- Develop agent ability to adapt to changes in environment
  - Learning.
- Ensure user confidence and trusting agents
  - Security technologies.
  - Deception-proof interaction protocols.
  - Models and infrastructure for trust and reputation.

Visions

Challenging to software agent technology

- Reasoning
  - representation of external stimulus
  - performance of reasoning
  - common sense
  - representation of information
  - maintaining truth
- Learning
  - methods
- Relations between agents
Visions

- Possible side-effects
  - Privacy
    - end-users’ profiles?
  - Security
    - mobile agents

Visions

- What can agents do?
  - ‘Common Sense’ - no way
  - Learn
    - user can teach
    - agent can follow and learn by following user’s actions
    - from other agents
  - Teach & guide
    - by giving examples
    - by telling what to do

Visions

- Users and software agents.
  - What do end users think about agents?
  - Do they trust on agents?
  - Do software agents really increase users’ productivity?

Visions

- Questions?
  - Comments.
  - Objections!
Introduction

Players in the field

Academic world

- University of Maryland Baltimore County
  http://agents.umbc.edu
- MIT
  http://agents.www.media.mit.edu/groups/agents/
- Stanford
  http://www-ksl.stanford.edu
- TU Berlin
  http://dai.cs.tu-berlin.de/english/
- QMW (part of the University of London)
  http://www.elec.qmw.ac.uk/isag
- Dartmouth College
  http://www.cs.dartmouth.edu/~agent/
- University of Washington
  http://www.cs.washington.edu/research/irdb.intro.html

Players in the field

Standardization

- OMG
  http://www.omg.org
- FIPA
  http://www.fipa.org
- W3C
  http://www.w3c.org

- EU
  EU sponsors several agent projects
- DARPA

Readings:

- Agent Technology
  Edited by Nicholas R. Jennings and Michael J. Wooldridge
- Mobile Agents and Security
  Edited by Giovanni Vigna
- Mobile Agents
  William T. Condrey, Michael Zyda
  Manning, ISBN 1-897777-36-8
- Intelligent Software Agents
  Water Brauer, Rudiger Zamski, Hartmut Wittig
  Springer, 1998
- Agent Technology - Foundations, Applications, and Markets
  Nicholas R. Jennings and Michael J. Wooldridge
  Springer, 1998
- Multi-Agent Systems, An Introduction to Distributed Artificial Intelligence
  Jacques Ferber, Addison-Wesley, 1999
Introduction