FIPA Standardization Activities in the Software Engineering Area

M. Cossentino¹, G. Hopmans²,³, J. Odell⁴

¹ Istituto di Calcolo e Reti ad Alte Prestazioni (ICAR), Consiglio Nazionale delle Ricerche(CNR), Palermo (Italy)

² Morpheus Software, Maastricht (The Netherlands),
³ Communications Research & Semiotics, University Maastricht

⁴ James Odell Associates, Ann Arbor, MI (USA)
FIPA Mission

The promotion of technologies and interoperability specifications that facilitate the end-to-end interworking of intelligent agent systems in modern commercial and industrial settings.
FIPA Structure

- Board of Directors
  - Secretariat
  - Image Committee
  - Finance and Audit Committee
  - Membership and Nomination Committee
  - Architecture Board
    - Technical Committees
    - Working Groups
    - Special Interest Groups

Interaction Protocols
Methodology
Modeling
Ontologies
Security
Semantics
Services
Interaction Protocols TC

Chair: Gabriel Hopmans
FIPA IP TC

• Goal: increase number of IP specs
  • Several papers describe the need for new Interaction Protocols.
  • Agentlink roadmap: “in the near-term but more in the medium-term future languages and protocols will be more agreed and standardized.”

• Activities and Work to be done
  • Develop/find new Interaction Protocols
  • Update of Interaction Protocol Library Specification

• Collaboration with Modelling TC : Feedback upon Communication Diagrams
• Collaboration with Metholodology TC: definition of Interaction Protocol and related terms
FIPA Protocols: limits

FIPA Protocols

- Small number of protocols used. Due to:
  - Number of patterns seem likely to capture many of the interaction patterns faced by business applications
  - Used informally – implemented as simple FSM and with no link to semantics of speech-acts within messages

- Coordination:
  - Protocols not easily linked to coordination relationships which may/should arise from interactions which is necessarily for use in open environments.
  - A possible avenue of future standardisation would appear to be drawing a closer relationship between protocols and stronger FIPA semantics to enable agents to more formally rely on agreements, commitments arising from the interactions
FIPA IP Semantics

• Performatives with defined semantics not used to date. Reasons:
  • Developers focused on protocols first, individual messages second (FSM style implementations)
  • Lack of reasoning tools for FIPA-SL
  • Difficulty in combining many layers of semantics (protocol, ACL, content-language and ontology) before full semantic reasoning could take place
Modeling TC

Chair: James Odell
Modeling TC

Purpose

• This work plan will focus on problems and notations needed to support an agent-based unified modeling language (AUML) and has the following goals:
  
  – 1. Gain an insight into how agent-oriented software engineering can benefit from UML and other modeling languages.
  
  – 2. Focus on problems and notations that are deemed necessary to support modeling of autonomous agents systems.
  
  – 3. Adopt notations that graphically express various aspects of agent-base modeling by extending UML and/or by using other notations.
  
  – 4. Address standards for AUML class and sequence diagrams in the first phase of this work plan. As other types of diagrams (such as deployment, goal, and activity diagrams) will be added to this work plan over time as the need and participation level reaches an appropriate level for a FIPA work plan.
Plan for Work and Milestones:

This work plan will create:

- 05/2003 Initial draft of language specifications for Class and Sequence Diagrams
- 10/2003 Initial draft of modeling language infrastructure specifications for Class and Sequence Diagrams
- 01/2004 Revised drafts of language specifications for Class and Sequence Diagrams
- 03/2004 Final draft of language specifications and infrastructure specifications for Class and Sequence Diagrams.
Modeling TC

• **Modeling Notation Sources**
  
  The Modelling TC participants have initially identified sources of notations that should be considered for a FIPA AUML.
  
  • UML 2.0 (Jim Odell)
  • AOR (Jim Odell)
  • PASSI (Massimo Cossentino)
  • MESSAGE (Radovan Cervenka)
  • Tropos (includes i* and GRL) (Radovan Cervenka)
  • ADELF E (Marc-Philippe Huget)
  • Gaia (Alfredo Garro)
  • BRIC (Jim Odell)
  • Styx (Radovan Cervenka)
  • Prometheus (Radovan Cervenka)
  • MADkit (Renato Levy)
  • OPM (Marc-Philippe Huget, Iris Reinhartz-Berger, Dov Dori, Onn Shehory, Arnon Sturm)
Modeling Areas

The Modelling TC participants have initially identified modelling areas that may be useful for representing and specifying agent-based systems.

• Multi- vs. single agent (delayed)
• Goal & soft goals (delayed)
• Social aspects (in progress)
• Environment
• Workflow/Planning (delayed)
• Levels of abstraction
• Temporal constraints
• Deployment and Mobility
Methodology TC

Chair: Massimo Cossentino
The proposed approach

Method Engineering

– The development methodology is constructed by the developer assembling pieces of the process (method fragments) from a method base.

– The method base is composed of contributions coming from existing methodologies and other novel and specifically conceived fragments.

Language: AUML
The Proposed Approach
Developing a system with ME

ME group

Perceives

System Development Object System

ME level

Perceives

Object system

SD level

Uses

Define a viewpoint on

Metamethods and CAME tools

Instantiate to

Development Methods and CASE tools

Instantiate to

System(s) specifications, System(s)

Represent
Phases of Methodology TC work

- Creation of the meta-model (for representing methodologies)
  - SPEM
- Identification of the method base architecture
- Collection of method fragments
- Description of techniques for method integration
- MAS meta-model descriptions
- Methodology terms glossary
Method engineering
What is a method

• A *method* (or *method fragment*) is a portion of (design) process to be performed
  – Expressed in SPEM – Software Process Engineering Metamodel

• It has some pre-conditions
  – Inputs of the fragment

• It produces (or maintains) one or more artifacts
  – Expressed with AUML if diagrams

• Usually some guidelines are provided
  – For instance, a set of rules which state:
    • by whom,
    • in what order,
    • in what way

*After completing the specified process we obtain a refinement of the system design*
Collection of method (fragments)

• Methods can origin from
  – most diffused methodologies
  – other specific contributions

• After their formalization (in terms of meta-model) they will be introduced in the method base
Identification of the method base architecture

- A technological infrastructure for the database of the method meta-models is necessary.
- The details of this base architecture strongly influence the support and availability of a CASE/CAME tool.
- In identifying the proper structure and implementation guidelines for the base, XMI and other XML-based languages will be considered.
  - ... and procedural aspects?
Description of techniques for method integration

• Guidelines will be identified for the integration of methods.

• An important role will play the specific architecture (FIPA) we are referring to
An example of method fragment
PASSI – The Agent Society Phase

A Work Definition

An UML Model
PASSI – A Fragment

• We will discuss a simple fragment from the PASSI methodology.
• This fragment builds the ontology of the system

<table>
<thead>
<tr>
<th>Input</th>
<th>To Be Designed</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Requirements doc,</td>
<td>Concepts</td>
<td>Ontology (MAS Meta-model component)</td>
</tr>
<tr>
<td>Glossary doc</td>
<td>Actions</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Predicates</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ontology elements Relationships</td>
<td></td>
</tr>
</tbody>
</table>
PASSI – The Domain Ontology Description
Fragment – Procedural aspects
PASSI – The Domain Ontology Description
Fragment – Structural aspects

FIPA-Platform Agent
Any question?