Implementing a Multi-Agent Architecture for Cooperative Software Engineering

Alf Inge Wang
Dept. Of Computer and Information Science
Norwegian University of Science and Technology (NTNU)

E-mail: al@finge.com
Web: http://home.finge.com
Implementing a Multi-Agent Architecture for Cooperative Software Engineering

• Agenda:
  – CAGIS multi-agent architecture for CSE
  – Requirements to technology
  – Technological guidelines
  – Experiences
  – Conclusion

• Introduction:
  – Hard to choose technology for prototypes
  – Hard to combine technologies
  – Technology guidelines for multi-agent development
CAGIS Multi-Agent Architecture for Cooperative Software Engineering

- **Agent:**
  - System agents
  - Local agents
  - Interaction agents

- **Agent Meeting Place:**
  - Facilitate inter-agent communication
  - Trade information and services
  - Define agent interaction
    - Ontology
    - Syntax
    - Services

- **Workspace:**
  - Temporary container
  - Private or shared

- **Repository:**
  - Persistent storage
  - Local, global or distributed
  - Experience base
CAGIS Multi-Agent Architecture for Cooperative Software Engineering II
Requirement to the technology

• General requirements
  – Open architecture
  – Minimum implementation cost
  – Performance not important

• System infrastructure
  – Heterogeneous and distributed environment
  – Infrastructure for WSs and AMPs:
    • Name registration
    • Service advertising
    • Monitor activities in WSs and AMPs
Requirement to the technology II

- Agent implementation and configuration
  - Open multi-platform
  - Mobile agents
  - Agent configuration

- Agent communication
  - Format to represent information
  - Agent-agent interaction
    - Syntax, semantics and pragmatics of conversation
    - Agent collaboration to reach a goal
    - Inter-agent coordination and negotiation
  - User-agent interaction
  - Agent access to external entities
Requirement to the technology III

- Knowledge sharing
  - Repositories
  - AMPs
  - Agents
- Information accessible for agents
Design proposal
Technological guidelines
Experiences from DIAS

• Four student used about 1000 hours
  – Skilled in Java-programming and XML
  – Never implemented agent software

• Choice of technology
  – Problem combining JATLite and Aglets
  – CORBA was not used as communication bus
  – KQML for agent communication
  – XML for information representation

• Experiences from using DIAS
  – Hard to implement new agents
  – Lacking of high-level support for inter-agent and inter-AMP communication
  – No integration with other systems
Experiences from DIAS II

- Two last-year students used about 1000 hours
  - Agent security
  - Integration of other systems through CORBA
  - Higher-level agent API
- Choice of technology
  - Wanted to migrate from JDK 1.1 to JDK 1.2
  - Implement CORBA support for DIAS
- Experiences from using DIAS II
  - High-level agent API
  - System support for agent and AMP communication
  - CORBA integration
Conclusion

- Be careful when choosing technologies
- Use accepted standards when available
- Implementing DIAS III
- Evaluating of CAGIS multi-agent architecture