A Multi-Agent Architecture for Cooperative Software Engineering

by

Alf Inge Wang, Reidar Conradi and Chunnian Liu

Dept. of Computer and Information Technology
Norwegian University of Science and Technology
Agenda

- Background / Motivation
- Presentation of a Multi-Agent Architecture
  - Agents
  - Workspaces
  - Agoras
  - Repositories
- An industrial scenario
- Application of our Architecture to the Scenario
- Conclusion
Background / Motivation

- SPT often focuses on strict organised, pre-planned way of working
- Introduce the term CSE
- Problem with traditional process architecture:
  - Too centralised
  - Too homogenous models
  - Hard to change tools and models
  - Need for open-ended spectrum of cooperative process tools
Multi-Agent Architecture (MAS)

Network of problem solvers

Advantages of MAS are:
- Decentralisation
- Reuse of previous components
- Cooperative work support
- Flexibility
Multi-Agent Architecture for CSE

CAGIS Multi-Agent Architecture:

- Agents
  - System agents
  - Local agents
  - Interaction agents
- Workspaces
- Agoras
- Repositories
Agent

- Piece of autonomous software
- System agents
- Local agents
- Interaction agents
  - Communication agents
  - Negotiation agents
  - Coordination agents
  - Mediation agents
Workspace (WS)

- Human and software agents access shared data and tools
- Private and shared workspaces
- Shared workspaces used for grouping people
Agora

* Place for software agent interaction
* Trading of information and services
* Facilities for:
  * Inter-agent communication
    * Provide a predefined set of conversation types
    * Specify a common syntax
    * Specify a common semantics
    * Specify pragmatics for agents
  * Inter-Agent Negotiation
Repository

- Information server
- Important types:
  - Production repository
  - Experience Base
- Software agent services
Interconnection of components

- Agents are created by people
- Agents are grouped mainly as people
- Interaction between agents is via Agoras
- Agents use repositories
- Within a group of agents, any process models are allowed
An industrial scenario

- Norwegian software company
- Developing products for MS NT and various UNIX platforms

Diagram:
- **Maintenance**
  - First line support
  - Maintenance group
  - Upd/Rel Plan group
- **Development**
  - Development group
- **Delivery**
  - Delivery/shipment group
- **Production & testing**
  - Production/QA group
A Multi-Agent Architecture for Cooperative Software Engineering

- First-line support
  - classification
  - forwarding
- Maintenance planning
  - register
  - estimate
  - allocate
- Development
  - coding
  - correction
  - module test
  - merging
- Update/Release planning
  - update per quarter
  - release per year
- Market/Technology requirements
- Resources
- Defect priority
- EB
- Change order
- Work order
- Agora 1
- Agora 2
- Customer reports
Conclusion

_advantages:
- Flexible and dynamic support
- Secretaries
- System adopts to changes
- Selection of process models

Advantages:
- Flexible and dynamic support
- Secretaries
- System adopts to changes
- Selection of process models

Disadvantages:
- Hard to get overview
- Termination of process, dead-locks