Enhancing Communication within Complex Information Systems: An Agent Approach

Prof. Jacques Pasquier
Agenda

1. Motivation
2. Our Approach to a Solution
3. Implementation
4. SimpleMAS: a Short Example
5. Conclusion
Agenda

1. Motivation
2. Our Approach to a Solution
3. Implementation
4. SimpleMAS: a Short Example
5. Conclusion
Information Systems

- Actual information systems can be viewed as:
  - A heterogeneous combination of software components.
  - Enhanced by structured human communication activity.
Human Layer

- Human activity characteristics:
  - Humans share common knowledge and vocabulary to communicate: they use an ontology.
  - Humans should act with respect to given rules, workflows, laws, regulations, procedures,…
  - Humans act autonomously, in cooperation or competition with others, trying to fulfill their goals.
  - Human can be considered as agents.
An **ontology** is a formal representation of a set of concepts within a domain and the relationships between those concepts.
Workflows

- Are useful to identify roles and responsibilities.
- Can be supported at run-time by workflows or rules engines.
Human activities are error prone:
- Bad communications and/or people not available.
- No respect of rules.
- Delay in reactivity → wasted time.
- Lack of traceability → difficult to establish ex-post responsibilities.

These problems prevent information to flow smoothly between human and software actors.
Agenda

1. Motivation
2. Our Approach to a Solution
3. Implementation
4. SimpleMAS: a Short Example
5. Conclusion
Main Idea

- Insert an integration layer between the "legacy" software layer and the human layer.
- It must act as a semi-automatic communication facilitator layer.
Main Characteristics

- **Virtual twin**: a dedicated software agent attached to each human one.
- Message exchanges based on a well-defined **ontology**.
- **Integration and special tasks agent-s**.
- Behaviors based on **declarative rules**, thus dynamic run-time modifications possible.
- Systemic qualities: clean design, generic and extensible.
Agenda

1. Motivation
2. Our Approach to a Solution
3. Implementation
4. SimpleMAS: a Short Example
5. Conclusion
Agent Metaphor

Basic properties of agents:

- **Autonomy**: agents have some degree of control over their actions and can work without external intervention.
- **Social ability**: agents can coordinate their actions, and cooperate with other agents to achieve their goals, using a common language to communicate with each other.
- **Reactivity**: agents can perceive their environment and respond to environmental changes.
- **Pro-activeness**: agents can act on their own initiative to achieve their goals instead of simply reacting to the environment.

According to this definition:
Humans can be considered as agents.
Agent Technology

- A **software agent** is a running program object, capable to initiate, receive, execute or reject a message autonomously to attain its goals during its life cycle.

- An **agent platform** is a software environment in which agents are incarnated and operate to achieve their goals. It provides following functionalities:
  - lifecycle management,
  - agent communication (naming and yellow pages services),
  - security mechanism.
Rule-Based Systems

- They must at least offer:
  - A knowledge base
    - Fact base / working memory
    - Rule base
  - An inference engine
    - Pattern matcher
    - Agenda
  - An execution engine
    - Forward chaining
    - Backward chaining

- Their nature is declarative.
Technological Choices

- The integration layer is built using JADE, a Java Agent Development Framework which offers the agent environment and a powerful communication architecture (flexible and efficient messaging, FIPA protocols, content languages, ontologies,…).

- JADE provides the shell of the agent, while the Jess rule engine is used as a decision component that performs all the necessary reasoning using knowledge supplied in the form of declarative rules.

http://jade.tilab.com/

http://www.jessrules.com/
The integration layer consists of two parts:

- a generic semi-finished structure: the MAIL (Multi-Agent Integration Layer) framework.
- an application specific implementation of this framework.
The **Virtual Twin** (aka Personal Agent) implementation: an agent-oriented implementation of the Observer pattern.

**MAIL ontology**: main concept of AgentOwner representing the human owner (flesh and bone) of a virtual twin.

Some useful **generic behaviors**: tailoring the behaviors offered by JADE.

**J2J toolkit**: generic tools easing the integration between JADE and Jess.

**Generic audit agent** (allows to log the critical message exchanges between agents) and **integration utilities** (e.g. database wrappers).

**A test and simulation toolkit**: test scenarios can be described with XML formalism and automatically executed.
Virtual Twin Model

1. Implementation

- JADE Main-Container
- JADE Container-1
- JADE Container-2
- JADE Container-3
- HTTP
- Mr. Blue
- UI Agents
- Core Agents

Software Engineering Group
Core Agents

- Is a stateful long-lived component and represents the human user in the system.
- Is deployed on a central and managed server.
- Communicates with other core agents of the system and with integration agents.
- Updates/synchronizes the potentially many UI agents attached to it.
UI Agents

- Are stateless short-lived components offering a user interface (e.g. graphical) to the human user.
- Are deployed on the human user's computers (e.g. desktops, laptops, smartphones, ...).
- Only communicates with their associated core agent.
Virtual Twin Implementation

1. Agent-Oriented Observer Pattern between Core- and UIAgent.
2. Classic Observer Pattern between UIAgent and UIs.
Agenda

1. Motivation
2. Our Approach to a Solution
3. Implementation
4. SimpleMAS: a Short Example
5. Conclusion
Description of SimpleMAS

- A simple multi-agent system (MAS) modeling human people that greet each other.
- Agents have some attributes: name, gender, age.
- Agents have the following simple behaviors:
  - Greet all the other existing agents.
  - Responds to the received greetings.
- Let us look at a first concrete scenario with three people:
  - Mr. James Bond, Male, 37
  - Mrs. Eva Green, Female, 28
  - Mr. Tom Anderson, Male, 35
SimpleMAS Running

1. 
2. 
3. 
4. SimpleMAS: a Short Example
5.
Changing the Rules

One can dynamically change the behaviors at run-time by changing the rules of each agent's rule engine.

Let us change the behavior of the agent James Bond to respond more warmly to women...

James Bond's new Jess rule is now:

(defrule MAIN::reply-greetaction-female
  ?rga <- (ReplyGreetAction (person ?p))
  ?per <- (Person (firstname ?fn) (lastname ?ln) (gender "female"))
  =>
  (modify ?rga (content (str-cat
                        "Hello " ?fn ". What a beautiful name!
                        My name is " (?p getFirstname) ".")
                      (return-fact-to-current-behaviour ?rga)))
SimpleMAS Running

1. SimpleMAS: a Short Example

2. Eva said:
   Hello, my name is Green, Eva Green.

3. Eva said:
   Hello, my name is Green, Eva Green.

4. Hello Mrs. Green, nice to meet you. My name is Tom.

5. James said:
   Hello Mrs. Green, nice to meet you. My name is James.

6. James said:
   Hello Eva, What a beautiful name! My name is James.

7. Tom said:
   Hello Mrs. Green, nice to meet you. My name is Tom.

8. Eva said:
   Hello, my name is Green, Eva Green.

9. Eva said:
   Hello, my name is Green, Eva Green.
Achievements

- The MAIL framework has passed through its first childhood diseases.
- Based on the current information system of a real medical laboratory, a prototypical alert notification system called MediMAS has been successfully developed. MediMAS is built on top of MAIL and thus validates the framework.
- Bachelor and Master projects brought first proofs of feasibility for the deployment on mobile devices.
- Although one can consider that the MAIL agents are "intelligent" (rule based actions), they are not goal oriented, i.e. their reasoning does not follow the BDI model. This kind of systems are investigated by the Jadex research project.
Future Work

- UniMAS (University Multi-Agent System): The idea is to use the MAIL framework for enhancing communication and collaboration between the different members and services of the academic microcosm (i.e. car sharing, offering a language course against one in programming, finding friends for various activities, ...).

- WADE (Workflows and Agents Development Framework) has been recently released. It would be interesting to investigate this new domain independent platform built on top of JADE, that gives JADE agents the ability to execute tasks defined according to the workflow metaphor. One of the main advantages of this approach is the possibility of representing processes in a friendly graphical form.

- The new JADE-Android add-on that allows running JADE agents on the Google Android platform offers exciting perspectives for new agent user interfaces and innovative use-cases.
Further Information

- Web page of the project:
  - http://diuf.unifr.ch/softeng/projects/aop.html

- Contact:
  - Prof. Jacques Pasquier
    Department of Informatics – University of Fribourg (CH)
    Software Engineering Group
  - jacques.pasquier@unifr.ch
  - http://diuf.unifr.ch/people/pasquiej