

#### Enhancing Communication within Complex Information Systems: An Agent Approach

Prof. Jacques Pasquier

7/11/2008

Workshop: Reasoning, Inference, Decision and Other Things... Department of Informatics, University of Fribourg, Switzerland

#### Agenda

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







#### Agenda

- 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.





#### Ontologies

6/32

**JNIVERSITA** 



An ontology is a formal representation of a set of concepts within a domain and the relationships between those concepts.





Can be supported at run-time by workflows or rules engines.





UNIVERSITÉ DE FRIBOURG /DEPARTMENT OF INFORMATICS

UNIVERSITAS

# Automation Gap

- Human activities are error prone:
  - Bad communications and/or people not available.
  - No respect of rules.
  - Delay in reactiveness wasted time.
- These problems prevent information to flow smoothly between human and software actors.





- 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 semiautomatic communication facilitator layer.



2. Our Approach to a Solution



# 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.





2. Our Approach to a Solution





- 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**





# **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.











# A Framework Approach

17/32

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.



# Architectural Cornerstones

- The Virtual Twin (aka Personal Agent) implementation: an agentoriented 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.









# **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.



3. Implementation



UNIVERSITÉ DE FRIBOURG /DEPARTMENT OF INFORMATICS



# **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.







3. Implementation

# Virtual Twin Implementation



- Agent-Oriented Observer Pattern between Core- and UIAgent.
- Classic Observer Pattern between UIAgent and Uls.





#### 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

#### 4.SimpleMAS: a Short Example

25/32
25/32
23/32

UNIVERSITAS

🕌 JamesUlAgent@diuf1x05:1099/JADE	🔮 EvaUlAgent@diuflx05:1099/JADE	
======================================	 Tom said: Hello Mrs. Green, nice to meet you. My name is Tom.  James said: Hello Mrs. Green, nice to meet you. My name is James.	
Greet Everybody Clear	Greet Everybody Clear	

🖆 TomUlAgent@diuflx05:1099/JADE	_ 🗆 🔀
======== Eva said: Hello, my name is Green. Eva Green.	
	<b>~</b>
Greet Everybody Clear	



INIVERSITÉ DE FRIBOURG /DEPARTMENT OF INFORMATIC

# 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

#### 4.SimpleMAS: a Short Example

97	122
21	SZ.

JamesUlAgent@diuflx05:1099/JADE     JamesUlAgent@diuflx05:1099/JADE     Eva said:     Hello, my name is Green. Eva Green.     Eva said:     Hello, my name is Green. Eva Green.	EvaUlAgent@diuflx05:1099/JADE   Hello Mrs. Green, nice to meet you. My name is Tom.   James said:   Hello Mrs. Green, nice to meet you. My name is James.   James said:   Hello Mrs. Green, nice to meet you. My name is James.
Greet Everybody Clear	Tom said: Hello Mrs. Green, nice to meet you. My name is Tom.
JamesAgentAdmin	🖆 TomUlAgent@diuflx05:1099/JADE 📃 🗖 🔯
(defrule MAIN:reply-greetaction-female         ?rga <       (ReplyGreetAction (person ?p))         ?per <       (Person (firstname ?fn) (lastname ?fn) (sex "female"))         =>       (nodify ?rga         (content       (content         (str-cat "Hello "?fn ". What a beautiful name! My name is "(?p getFirstname) ".")))         (return-fact-to-current-behaviour ?rga)	Eva said: Eva said: Hello, my name is Green. Eva Green. Eva said: Hello, my name is Green. Eva Green.

Greet Everybody Clear



#### Agenda

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







#### Achievements

3. 4. 5.Conclusion

- 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

5.Conclusion

30/32

- 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**

5.Conclusion

- Web page of the project:
  - <u>http://diuf.unifr.ch/softeng/projects/aop.html</u>
- Contact:
  - Prof. Jacques Pasquier
     Department of Informatics University of Fribourg (CH)
     Software Engineering Group
  - iacques.pasquier@unifr.ch
  - <u>http://diuf.unifr.ch/people/pasquiej</u>









#### QUESTIONS ?

UNIVERSITÉ DE FRIBOURG 7 DEPARTMENT OF INFORMATICS

7/11/2008

UNIVERSITAS FRIBURGENSIS

> Workshop: Reasoning, Inference, Decision and Other Things... Department of Informatics, University of Fribourg, Switzerland