Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

Communication Ontological Description Process Fragment

M. Cossentino, V. Seidita

Rapporto Tecnico N.:
RT-ICAR-PA-10-01 **Giugno 2010**

Consiglio Nazionale delle Ricerche, Istituto di Calcolo e Reti ad Alte Prestazioni (ICAR)
– Sede di Cosenza, Via P. Bucci 41C, 87036 Rende, Italy, URL: *www.icar.cnr.it*
– Sede di Napoli, Via P. Castellino 111, 80131 Napoli, URL: *www.na.icar.cnr.it*
– Sede di Palermo, Viale delle Scienze, 90128 Palermo, URL: *www.pa.icar.cnr.it*

Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

Communication Ontological Description Process Fragment

M. Cossentino[1], V. Seidita[2]

Rapporto Tecnico N.:
RT-ICAR-PA-10-01

Data:
Giugno 2010

---

[1] Istituto di Calcolo e Reti ad Alte Prestazioni, ICAR-CNR, Sede di Palermo Viale delle Scienze edificio 11 90128 Palermo
[2] Università degli Studi di Palermo Dipartimento di Ingegneria Informatica Viale delle Scienze 90128 Palermo

Communication Ontological Description

Process Fragment

Author: M. Cossentino, V. Seidita

Last saved on: 23/06/2010

Index

Fragment Goal

Describing agent communications in terms of exchanged knowledge (referred to an ontology), content language and interaction protocol.

Fragment Origin

The presented fragment has been extracted from *PASSI* (Process for Agent Societies Specification and Implementation) design process.
PASSI is a step-by-step requirement-to-code methodology for designing and developing multi-agent societies. The methodology integrates design models and concepts from both Object Oriented software engineering and artificial intelligence approaches.

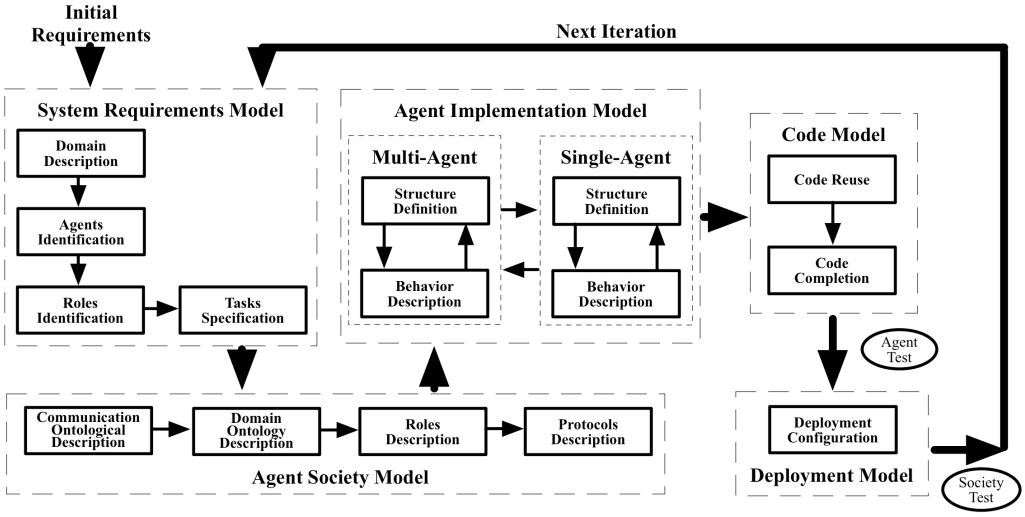

Figure 1. The PASSI design process

The design process is composed of five models (see Fig. 1): the System Requirements Model is a model of the system requirements; the Agent Society Model is a model of the agents involved in the solution in terms of their roles, social interactions, dependencies, and ontology; the Agent Implementation Model is a model of the solution architecture in terms of classes and methods (at two different levels of abstraction: multi and single-agent); the Code Model is a model of the solution at the code level and the Deployment Model is a model of the distribution of the parts of the system (agents) across hardware processing units, and their movements across the different available platforms.

In the following the PASSI process will be described by initially considering its whole process and then its five components, each of them representing a phase, a portion of work for which a specific outcome and milestones can be identified and represented in the following diagram.

The PASSI Process lifecycle

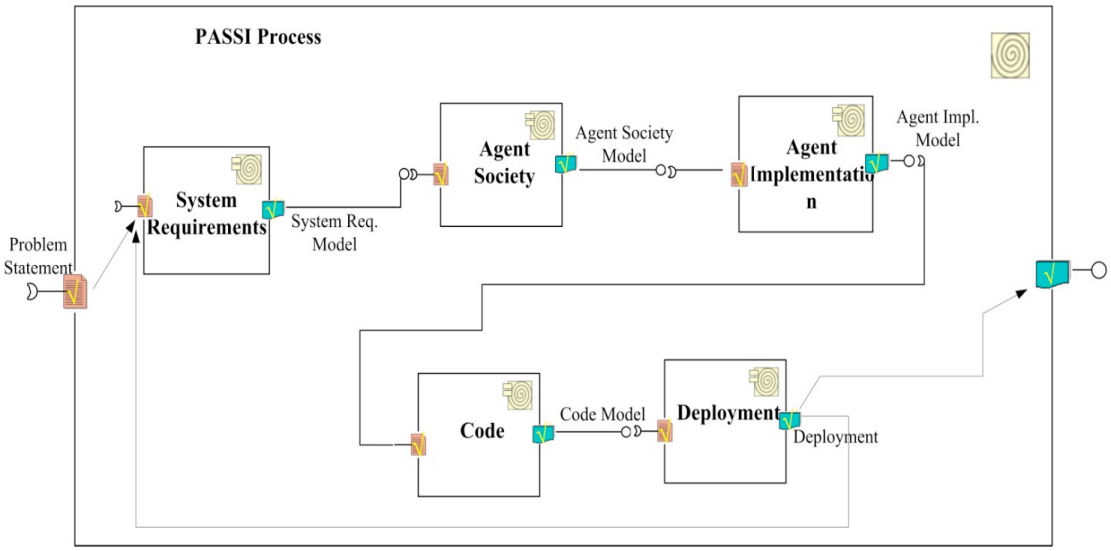

Figure 2. The PASSI process 5 phases

PASSI includes five components (see Figure 2):

- System Requirements: It covers all the phases related to Req. Elicitation, analysis and agents/roles identification
- Agent Society: All the aspects of the agent society are faced: ontology, communications, roles description, Interaction protocols
- Agent Implementation: A view on the system's architecture in terms of classes and methods to describe the structure and the behavior of single agent.
- Code: A library of class and activity diagrams with associated reusable code and source code for the target system.
- Deployment: How the agents are deployed and which constraints are defined/identified for their migration and mobility.

We can draw a direct correspondence between the above-cited components and the five phases that characterise the procedural representation of PASSI represented in Figure 1.
In fact, the PASSI process is composed by five different phases: System Requirements, Agent Society, Agent Implementation, Code and Deployment (the same names of the components).
Each phase produces a document that is usually composed aggregating the UML models and work products produced during the related activities. Each phase is composed of one or more sub-phases each one responsible for designing or refining one or more artefacts that are part of the corresponding model (for instance the System Requirements model includes an agent identification diagram that is a kind of UML use case diagrams but also some text documents like a glossary and the system use scenarios).

Fragment Description

Consider the PASSI process (Figure 1) and the "Agent Society" phase with its outcome "Agent Society Model". Now, let us consider the "Communication Ontological Description" (red colored in Figure 3) activity and the consequent outcome (the "Communication Ontological Description" composite document).
This activity aims to model the social interactions and dependencies among the agents involved in the solution and to face the following agent society aspects are faced:

communication and role description. The activity and its main outcome has been considered for being extracted from PASSI and for becoming a process fragment.

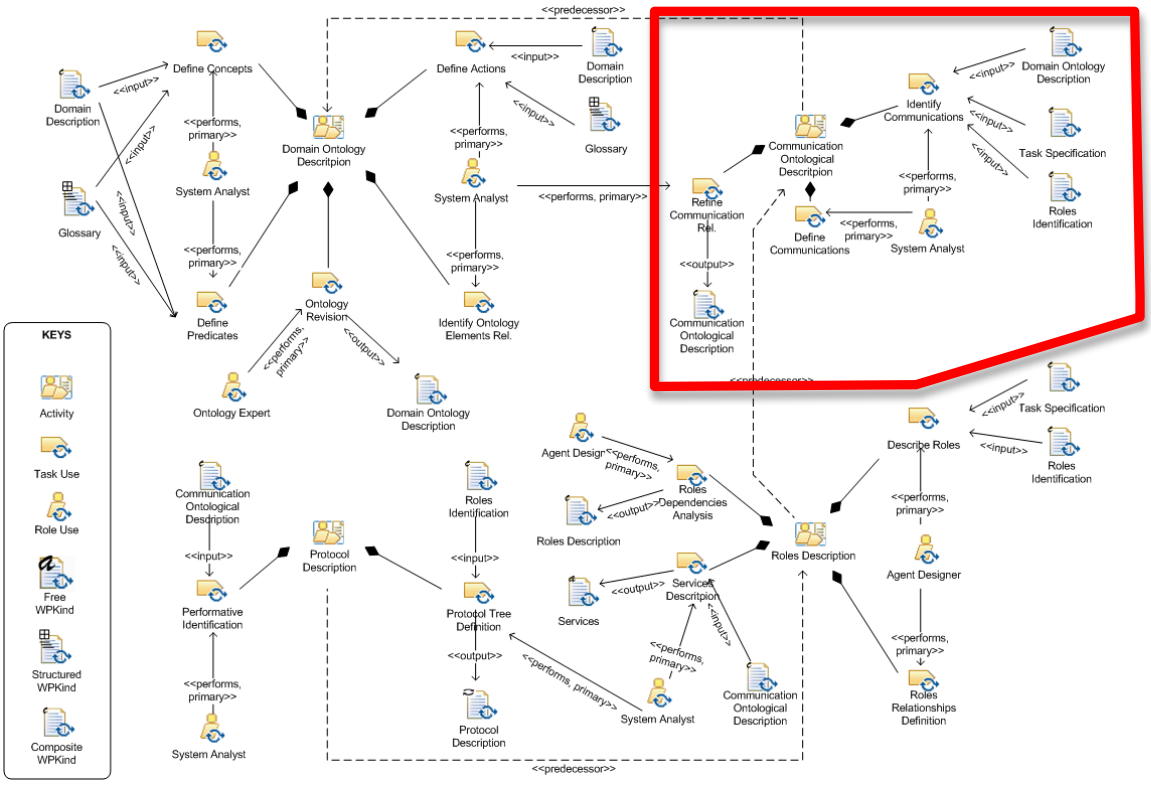

Figure 3. ***The Agent Society Phase (structural view)***

Portion of Process workflow

The process that is to be performed in order to obtain the result is represented in the following as a SPEM2.0 diagram.

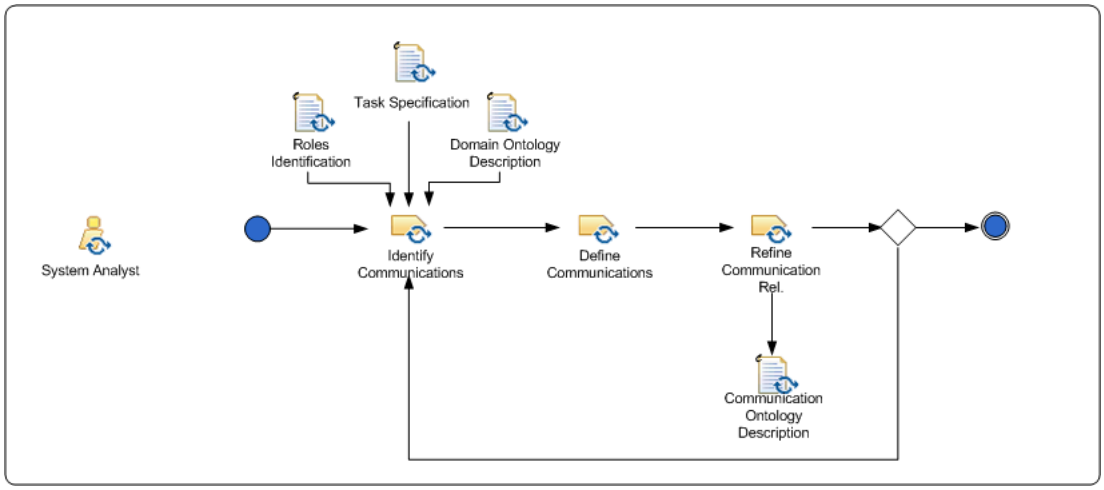

Figure 4. The flow of activity of this fragment

Fragment MAS metamodel

The portion of metamodel of this fragment is:

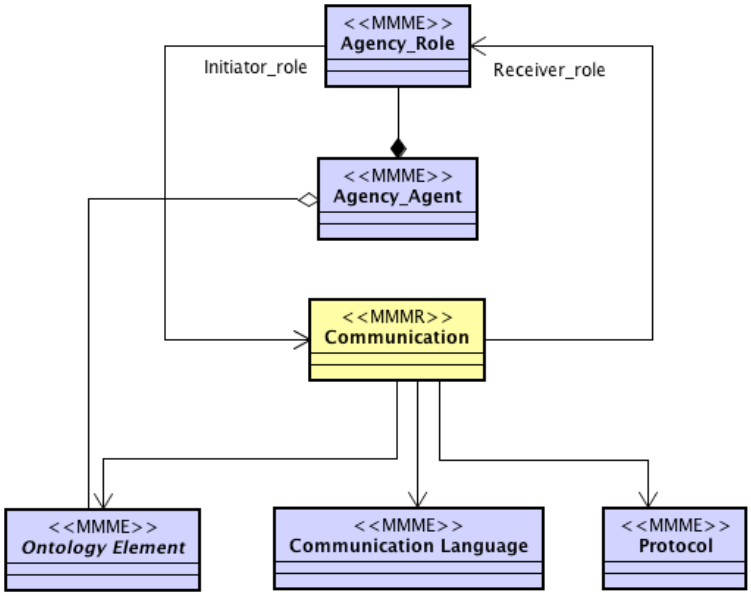

Figure 5. The fragment MAS metamodel

This fragment refers to the MAS meta-model adopted in PASSI and contributes to define and describe the elements reported in .

Glossary

Communication – An interaction among two agents, referring an Agent Interaction Protocol and a piece of the domain ontology (knowledge exchanged during the interaction). Usually it is composed of several messages, each one associated with one communicative act (or performative).
Message - an individual unit of communication between two or more agents that point out the standard FIPA message format. Usually a message is associated with a communicative act (or performative)

Activities description:

Activity/Task	Activity/Task Description	Roles involved	Input	Output
Define Agents and Roles	Agents and Roles defined during previous phase are now used to define Agency_Agent and Agency_Role according to a 1:1 transformation.	Automatic*	Agent, Role	(I)Agency_Agent, (I)Agency_Role

Identify Communications	It consists in defining communications among agents looking at exchanged messages in the scenario.	System Analyst (perform)	Scenario, Actor, Message_ RR, Message_ RA	(I)Communication
Define Communications	It consists in the description of agents' communications in term of ontology, content language and interaction protocol. Agents' knowledge structures necessary to deal with communication contents have to be introduced in agents.	System Analyst (perform), Ontology Expert (assist)	AA, CA, CC, CP, PP, PA (-OntoRel), Concept, Predicate	(R)Communication (C) AA, CA, CC, CP, PP, PA (-OntoRel), Concept, Predicate (I) Communication-Role (Association), Communication-Protocol (Association), Communication-Content Language (Association), Communication - Ontology Element (Association), Agent to Ontology Element (Aggregation)
Refine Communication Relationships	The identification of association classes in order to link each communication to the three fundamental element of communication itself (ontology, language and protocol).	System Analyst (perform)		

** Automatic means that no specific human activity is needed. The operation is usually performed by a tool.*

Deliverable relationships with the MMM

The following figure describes the structure of this fragment work products in relationship with the MAS model elements:

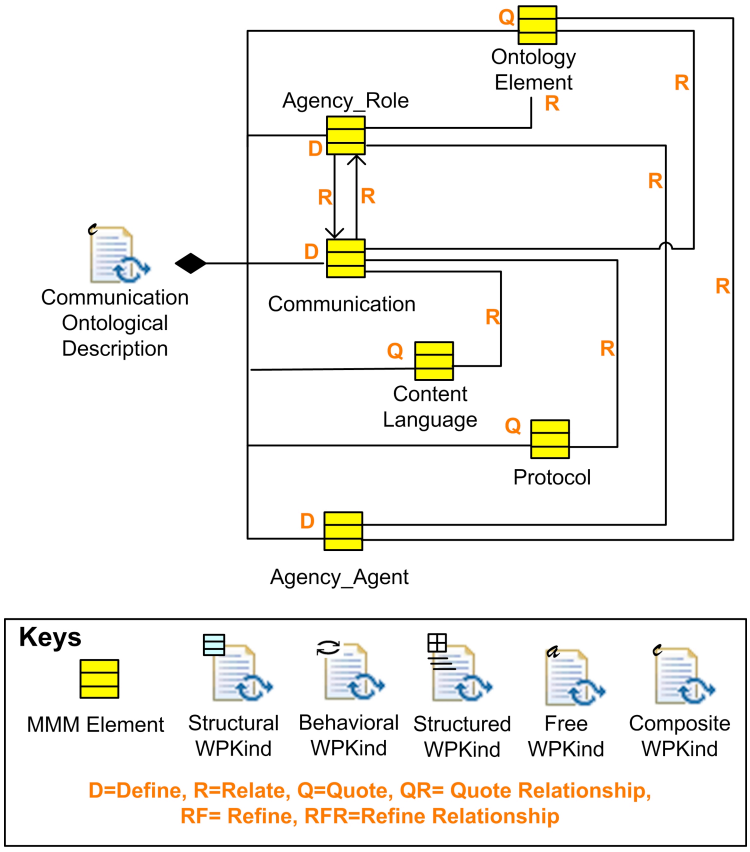

Fig.5. Structure of the fragment work-product in terms of MAS meta-model elements

Roles involved in this fragment are:

- Ontology Expert,
- System Analyst,
- Agent Designer.

They are described in the following subsections.

Ontology Expert

He is responsible for:

1. Assisting System analyst in defining communications.

System Analyst

He is responsible for:

1. Communications identification. It consists in introducing an association for each communication between two agents, looking at exchanged messages in the scenario.
2. Communications definition. The description of agents' communication in terms of ontology, content language and interaction protocol.
3. Communication relationships refinement. The identification of association classes in order to link each communication to the three fundamental element of communication itself (ontology, language and protocol).

Deliverables

Communication Ontological Description Document

This fragment produces a composite document composed by class diagram (whose classes represent agents and communications) and a text document describing the elements reported in the diagram.
The Communication Ontology Description (C.O.D.) diagram is a representation of the agents' (social) interactions; this is a structural diagram (for instance a class diagram) that shows all agents and all their interactions (lines connecting agents).
According to FIPA standards, communications consist of speech acts (Searle, 1969) and are grouped by FIPA in several interaction protocols that define the sequence of expected messages. As a consequence, each communication is characterized by three attributes, which we group into an association class. This is the characterization of the communication itself (a communication with different ontology, language or protocol is certainly different from this one) and its knowledge is used to uniquely refer this communication (which can have, obviously, several instances at runtime since it may arise more than once).

Agents' Knowledge

Agent	Knowledge piece	Data Type	Description

Communication details

From	To	Protocol	Content Language	Content (referred to ontology)	Description

Example of notation

Each agent (fill colour: yellow) is described in terms of its knowledge (pieces of the ontology described in the previous diagram). There is one relationship between two agents for each communication they are involved in. In each relationship the roles played by the agents during the communication are also reported.
Each communication (fill colour: white) is represented by the relationship among the two agents and it is detailed in the relationship attribute class. The class is identified by an unique name (also reported in the relationship among the two agents) and it is described by the ontology, language and protocol fields.
The ontology field refers to an element of the DOD (Domain Ontology Description); the language addresses for the content language of the communication while the protocol points out the adopted FIPA Interaction Protocol.

In Figure 6, the *PurchaseManager* agent starts a conversation (see *QueryForAdvice* association class) with the *PurchaseAdvisor* agent. The Conversation contains the *Course* ontology, the *Query* protocol and the RDF language. This means that the *PurchaseManager* wants to perform a speech act based on the FIPA's query protocol in order to ask the *PurchaseAdvisor* advice on how to purchase (supplier, number of stocks, number of items per each, purchase-money) provided the Course information.

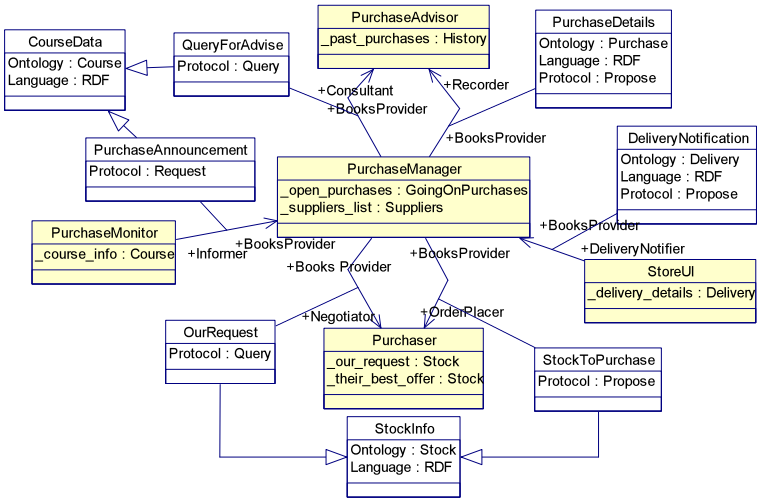

Figure 6. An example of Communication Ontological Description diagram

Preconditions and concepts to be defined

Input, output and elements to be designed in the fragment are detailed in the following tables.

As regards documents:

Input	Output
Roles Identification Document	Communication Ontological Description Document
Domain Ontology Description document	
Agents Identification document	

As regards MAS metamodel elements:

Input	To Be Designed	To be related (relation name)	To be refined	To be quoted	Relationships to be quoted
Scenario	Communication	Communication-Role (Association)		Ontology Element	Role - Agent
Actor	Agency_Agent*	Communication-Protocol (Association)		Protocol	
Message_RR	Agency_Role*	Communication-Content Language (Association)		Content Language	
Message_RA		Communication - Ontology Element (Association)			
AA,CA, CC, CP, PP, PA-OntoRel		Agent to Ontology Element (Aggregation)			

Agent					
Role					
Functional Requirement					
Non Functional Requirement					

* Obtained by 1:1 transformation from Problem Domain Agent and Role

Guideline

In designing this diagram we start from the results of the A.Id. (Agent Identification) phase. A class is introduced for each identified agent, and an association is then introduced for each communication between two agents. Clearly, it is also important to introduce the proper knowledge structure (selected from elements of the Domain Ontology Description) in each agent in order to store exchanged information. The association line that represents each communication is drawn from the initiator of the conversation to the other agent (participant) as can be deduced from the description of their interaction performed in the Role Identification (R.Id.) phase.
Roles played by agents in the interaction (as derived from the R.Id. diagrams) are reported at the beginning and the end of the association line.

Composition Guideline

--

Dependency Relationships with other fragments

None