Identify Communications

Atomic Process Fragment

Author(s): M. Cossentino, V. Seidita

Last saved on: 23/11/10 15:41

Index

Fragment Description

Fragment Goal

Identify communications among Agency_Roles starting from scenarios and agent interactions in functional-oriented models.

Fragment Origin

The presented fragment has been extracted from *PASSI* (Process for Agent Societies Specification and Implementation) design process.

PASSI (Process for Agent Societies Specification and Implementation) 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.

PASSI has been conceived in order to design FIPA-compliant agent-based systems, initially for robotics and information systems applications.

Systems designed by using the PASSI process are usually composed of peer-agents (although social structures can be defined). According to FIPA specifications agents are supposed to be mobile, and they can interact by using semantic communications referring to an ontology and an interaction protocol.

PASSI is suitable for the production of medium-large MAS (up to a hundred agent-kinds each one instantiated in an unlimited number of agents in the running platform).

The adoption of patterns and the support of specific CASE tools (PTK) allows a quick and affordable production of code for the JADE platform. This encourages the use of this process even in time/cost-constrained projects or where high quality standards have to be met.

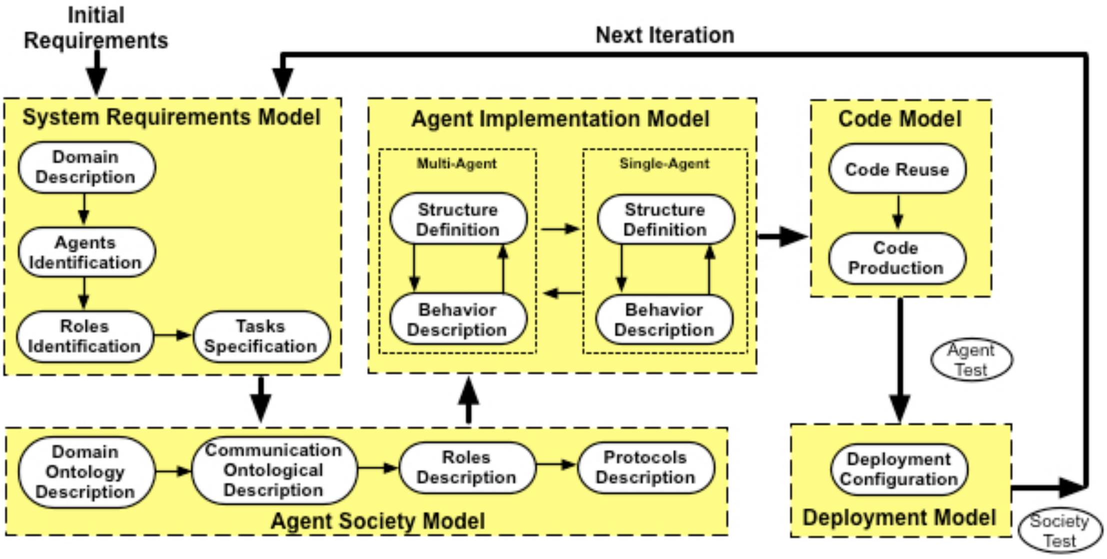


Figure 1. The PASSI design process

The design process is composed of five models (see Figure 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 (i.e. agents) across hardware processing units, and their movements across the different available platforms.

Useful references about the PASSI process are the following:

- M. Cossentino. From Requirements to Code with the PASSI Methodology. In Agent-Oriented Methodologies, B. Henderson-Sellers and P. Giorgini (Editors). Idea Group Inc., Hershey, PA, USA. 2005.
- M. Cossentino, S. Gaglio, L. Sabatucci, and V. Seidita. The PASSI and Agile PASSI MAS Meta-models Compared with a Unifying Proposal. Lecture Notes in Computer Science, vol. 3690. Springer-Verlag GmbH. 2005. pp. 183-192.
- M. Cossentino and L. Sabatucci. Agent System Implementation in Agent-Based Manufacturing and Control Systems: New Agile Manufacturing Solutions for Achieving Peak Performance. CRC Press, April 2004.
- M. Cossentino, L. Sabatucci, and A. Chella. Patterns reuse in the PASSI methodology. In Engineering Societies in the Agents World IV, 4th International Workshop, ESAW 2003, Revised Selected and Invited Papers, volume 3071 of Lecture Notes in Artificial Intelligence. Springer-Verlag, 2004. pp. 294-310
- M. Cossentino, L. Sabatucci, A. Chella - A Possible Approach to the Development of Robotic Multi-Agent Systems - IEEE/WIC Conf. on Intelligent Agent Technology (IAT'03). October, 13-17, 2003. Halifax (Canada)
- Chella, M. Cossentino, and L. Sabatucci. Designing JADE systems with the support of case tools and patterns. Exp Journal, 3(3):86-95, Sept 2003.

The Process lifecycle

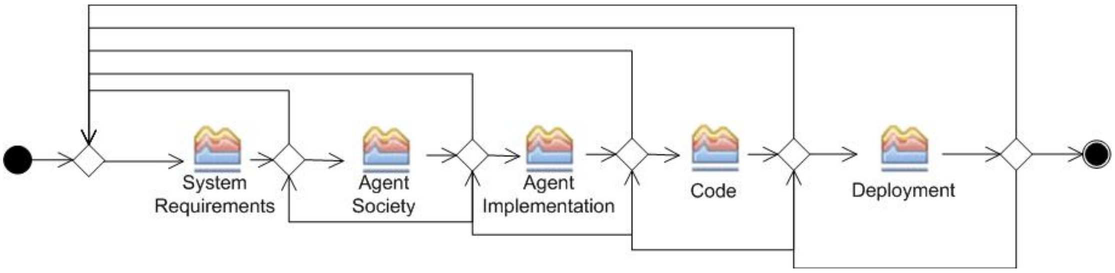


Figure 2. The PASSI process phases

PASSI includes five phases (see Figure 2) arranged in an iterative/incremental process model:

- 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.

Each phase produces a document that is usually composed aggregating 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 Overview

This fragment is positioned in the “Agent Society” phase of the PASSI process (see Figure 2). It is a component of the “Communication Ontological Description” (red colored in Figure 3) activity.

This task aims to identify the communications involving agents of this model. The task and its outcome (instantiation of some elements) has been considered for being extracted from PASSI and for becoming an atomic fragment.

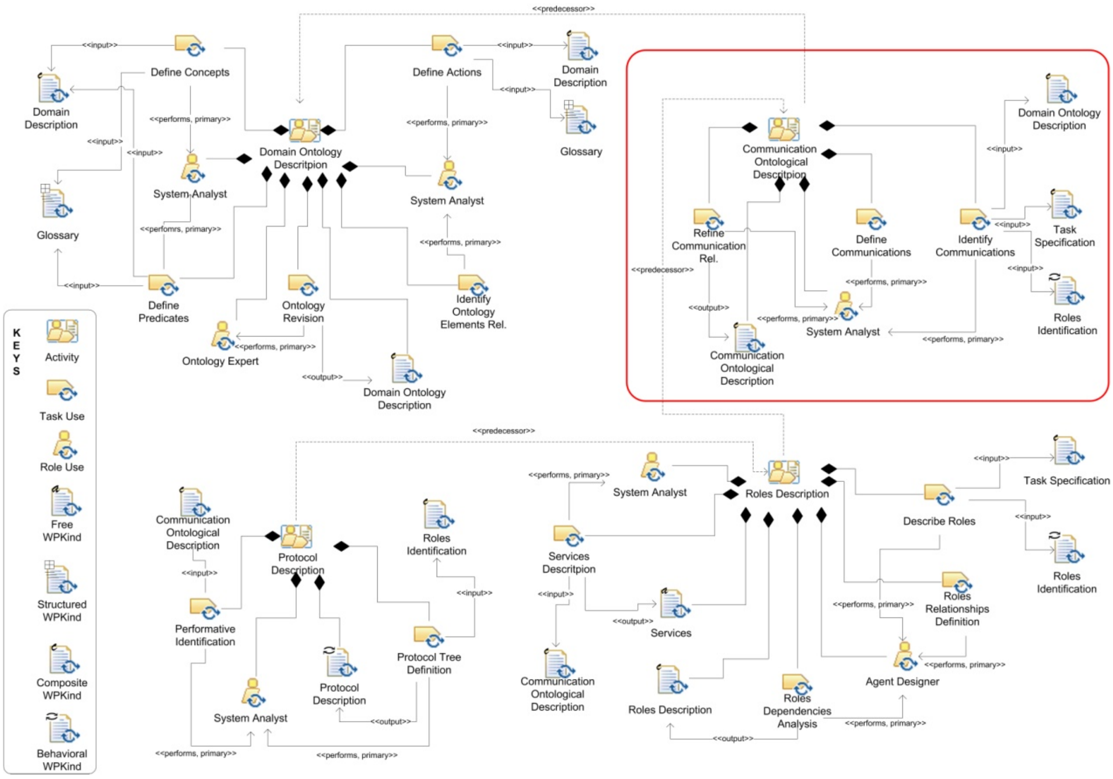


Figure 3. The Identify communications fragment within the PASSI Agent Society model structural view

Fragment System metamodel

The portion of metamodel of this fragment is:

Definition of System metamodel elements

This fragment underpins the following model elements:

Definition of System metamodel relationships

This fragment underpins the following relationships among the model elements:

System metamodel Input/Output

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

As regards system metamodel elements:

Input		**To Be Designed**		**To Be Refined**		**To Be Quoted**	
SMME	**SMMR**	**SMME**	**SMMR**	**SMME**	**SMMR**	**SMME**	**SMMR**
		Agency_Role	Communication				

Definition of input system metamodel elements and relationships

Stakeholders

Roles involved in this fragment are:

- Role 1

Their responsibilities are described in the following subsections.

Role 1

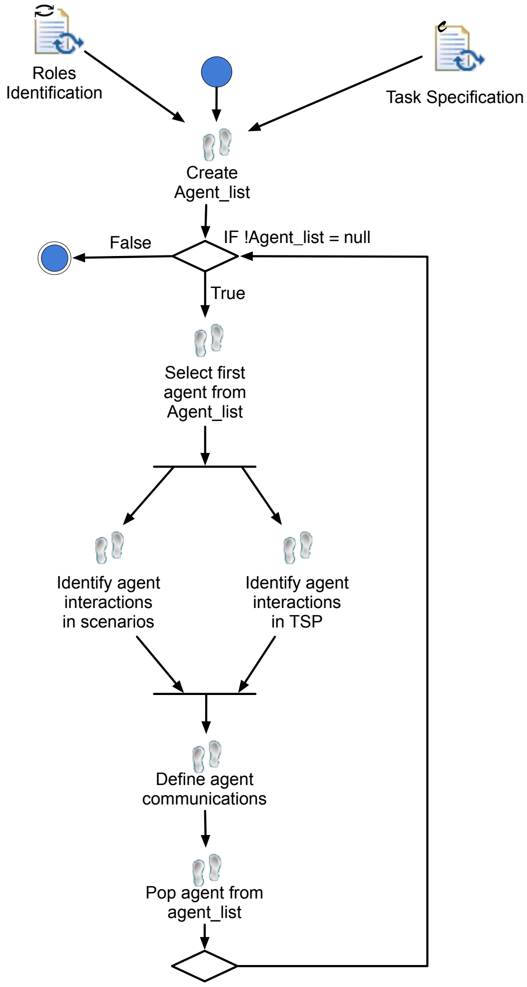
(S)He is responsible for:

1. activity

Fragment workflow

Workflow description

The process that is to be performed in order to obtain the result is represented in the following as a SPEM 2.0 diagram.


Roles Identification
Task Specification
Create Agent_list
False
IF !Agent_list = null
True
Select first agent from Agent_list
Identify agent interactions in scenarios
Identify agent interactions in TSP
Define agent communications
Pop agent from agent_list

Activity description

The fragment encompasses the following *work breakdown elements*:

Name	Kind	Description	Roles involved
Create Agent_List	Step	The list of agents is defined by looking at input documents of this fragment.	System Analyst
Select first agent from Agent_list	Step		System Analyst
Identify agent interactions in scenarios	Step		System Analyst
Identify agent interactions in TSP	Step		System Analyst
Define agent communications	Step	Interactions found in considered documents are grouped in communications according to their aim, knowledge exchanged and other main parameters of communications	System Analyst
Pop agent from Agent_list			

System metamodel elements and relationships input/output

The above described *work breakdown elements* have the following input/output in terms of system metamodel components.
In the Input column, system metamodel components utilization is completed by the name of the input document reporting them in the original design process.

	Input		Output	
Activity/Task Name	SMME	SMMR	SMME	SMMR

WP Input/Output

Input, output work products to be designed in the fragment are detailed in the following tables.

Input	Output

Deliverable

Document's name

Domain Requirements Description Diagram notation

Example

Deliverable relationships with the system metamodel

Guidelines

Enactment Guidelines

--

Reuse Guidelines

Composition

--

Dependency Relationship with other fragments

Glossary

References