Roles Identification

Process Fragment

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

Last saved on: 04/11/10 16:19

Index

Fragment Description

Fragment Goal

Identifying the roles played by agents and their main interactions.

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.

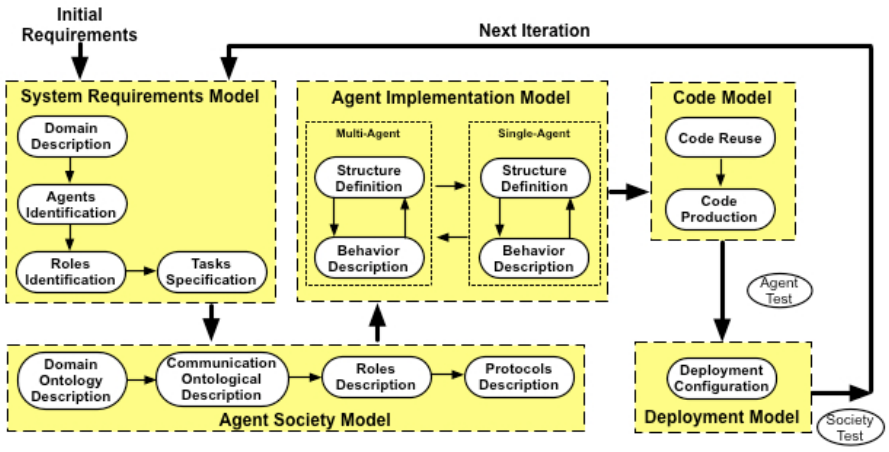


Figure 1. The PASSI design process

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

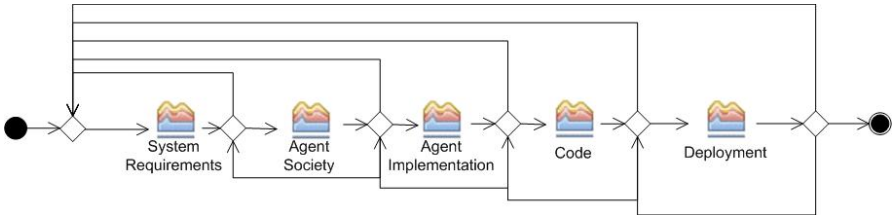


Figure 2. The PASSI process phases

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

Located inside the PASSI System Requirement phase reported in the following Figure 3, this fragment includes the activity “Roles Identification” (red box).
The aim is to identify the roles played by agents and to roughly describe their main interactions by starting from the requirements assigned to each agent and a text description of scenarios.

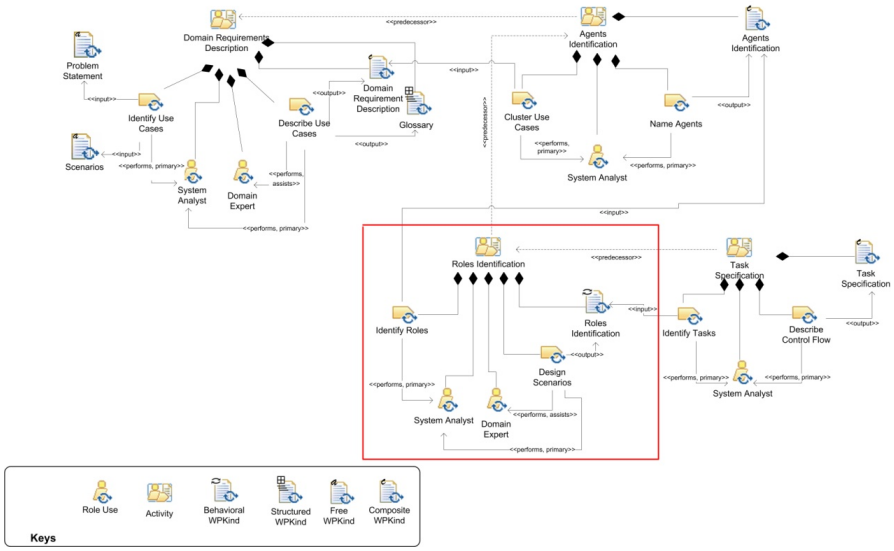
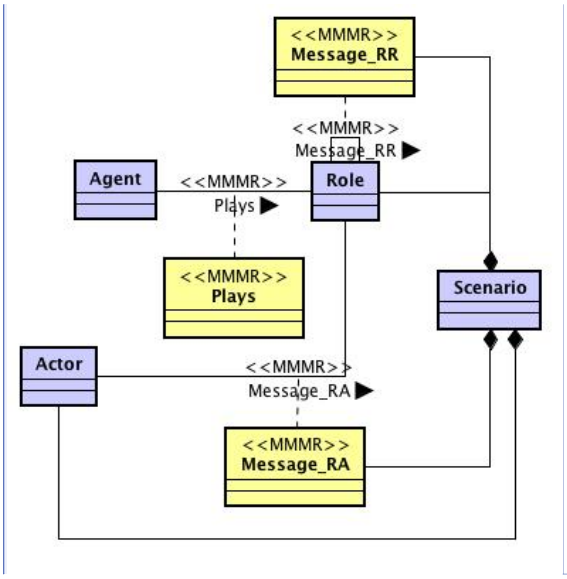


Figure 3. ***The System Requirements Phase (structural view)***

Fragment System metamodel

The portion of metamodel of this fragment is:



Valeria 4/11/10 16:19
Comment: aggiornare

Figure 4. 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 Figure 4.

Definition of System metamodel elements

Element	Type	Definition
Role	MMME	A portion of the behaviour of an agent that is characterized by a goal (accomplishing some specific functionality) and/or provides a service.
Scenario	MMME	An instance of a use case describing a concrete set of actions. A scenario is composed of the following fields: - Name: used to identify the scenario - Participating actors: the list of participating actors (frequently actor instances are used) - Flow of events: describing the flow of events step by step.
Message_RR	MMMR	A message exchanged between two roles. See glossary for message definition
Message_RA	MMMR	An interaction between an agent and a role. This can be a message as well as another kind of interaction (for instance GUI-based)
Plays	MMMR	It specifies which agent plays a role

Valeria 4/11/10 16:19
Comment: dividere MMMR e MMME

Keys:
MMME= MAS Metamodel Element
MMMR= MAS Metamodel Relationship

Definition of System metamodel relationships

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 Defined		To be refined		To be quoted	
MMME	**MMMR**	**MMME**	**MMMR**	**MMME**	**MMMR**	**MMME**	**MMMR**
Functional Requirement	Functional Requirement – Functional Requirement (UC_Relationship)	Role	Role-Role (Message_RR)			Agent	
Actor			Role-Actor (Message_RA)			Actor	
Agent			Role-Agent (Plays)				

Definition of input system metamodel elements and relationships

Functional requirement - Functional requirements describe the functions that the software is to execute. (from IEEE SEBOK 2004)
Actor - An external entity (human or system) interacting with the multi-agent system.
Agent – PASSI considers three different levels of abstraction in the agent definition:
1) the system requirements domain agent is a responsibility center; this means that each agent will rationally act to achieve its goals (usually defined in terms of functionalities it should ensure).
2) during the Agency Domain design phases, the agent is an autonomous entity capable of pursuing an objective through its autonomous decisions, actions and social relationships.
3) In the solution domain phases, an agent is a significant software unit, each agent is an instance of an agent class.

Generally speaking, an Agent is an entity which:
- is capable of actions in an environment;
- can communicate directly with other agents typically using an Agent Communication Language;
- is driven by a set of functionalities it has to accomplish;
- possesses resources of its own;
- is capable of perceiving its environment;
- has only a partial representation of this environment in form of an instantiation of the domain ontology (knowledge);
- can offer services;
- can play several, different (and sometimes concurrent or mutually exclusive) roles.

Stakeholder

Two roles are involved in this fragment: the System analyst and the Domain Expert. They are described in the following sub-sections:

System Analyst

He is responsible for:

1. Roles identification. The System Analyst studies (textual) scenarios and system requirements and he/she identifies the roles played by agents.
2. Designing scenarios in the chosen notation (usually UML sequence diagrams).

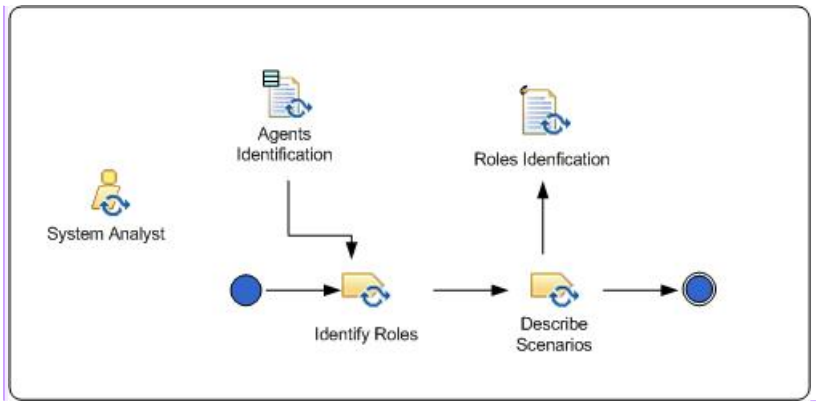
Domain Expert

He supports the system analyst during the description of scenarios.

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 diagram.



Valeria Seidita 28/9/10 14:08

Comment: non è la figura giusta, non c'è il domain expert e l'icona dell'agent identification è sbagliata, l'ho messa su dropbox , vorrei che la cambiassi tu visto che word non è stabile!

Figure 5. The flow of activity of this fragment

Activity description

The fragment encompasses the following *work breakdown elements*:

Name	Kind	Description	Roles involved
Identify Roles	Task	The System Analyst studies (textual) scenarios and system requirements (as defined in the previous phase) and identifies the roles played by agents	System Analyst (perform)

Design Scenarios	Task	Each scenario in designed in form of sequence diagram thus depicting the details of agents interactions	System Analyst (perform) Domain Expert (assist)

System metamodel elements and relationships input/output

The above described *work breakdown elements* have the following input/output in terms of system metamodel components:

	Input		Output	
Activity/Task Name	**MMME**	**MMMR**	**MMME**	**MMMR**
Identify Roles	Agent, Actor, Functional Requirement (from Agent Identification Document)	UC_Relationship (Generalize, Include, Extend, Communicate), Is_ResponsibleFor, Association (From Agent Identification Document	Role	
Design Scenarios			Scenario	Message_RR, Message_RA

Valeria Seidita 28/9/10 14:09

Comment: non avevamo detto di non usare acronimi di questo tipo? poi ci dimentichiamo cosa volevamo dire.

WP Input/Output

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

Input	**Output**
Agents Identification diagram	Roles Identification Diagrams

Deliverable

Roles Identification Diagrams

This fragment delivers one composite document. It is composed of one or more Roles Identification Diagram(s), a text description of the scenario reported in each diagram and one or more tables describing role features. Usually diagrams are UML sequence diagrams reporting roles played by agents, actors and the message they exchange.

Sequence diagrams describe all the possible communication paths between agents. A path describes a scenario of interacting agents working to achieve a required behaviour of the system. Each agent may belong to several scenarios, which are drawn by means of sequence diagrams in which objects are used to symbolize roles.

Roles Identification Diagram: example of notation

Common UML sequence diagrams are used to identify roles.
The name of each class is in the form: <role name>:<agent name>
Role names are not underlined because they are not instances of agent classes but rather they represent a part of the agent behaviour.

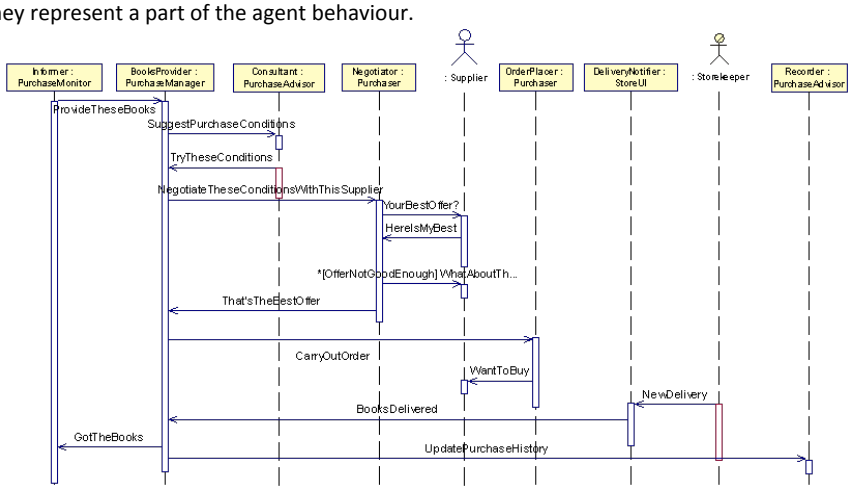


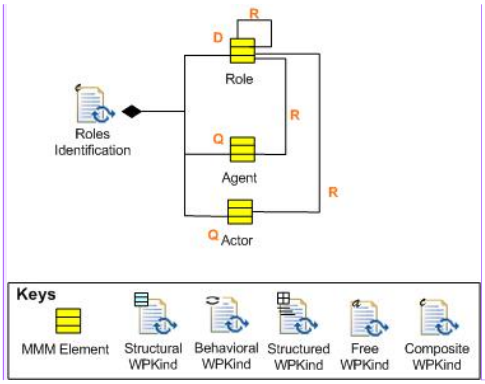
Figure 6. An example of Role Identification Diagram

Structure of the table used to describe roles:

Name	Description	Responsibilities

Deliverable relationships with the MMM

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



Valeria Seidita 28/9/10 14:13
Comment: ma questa l'hai cambiata? la risoluzione è pessima, almeno nel mio monitor.

Figure 7. Structure of the RID fragment work-product with respect to the MAS meta-model

Guidelines

Enactment Guidelines

Roles can be identified by describing scenarios arising from the instantiation of use case flows of activities.
This flow naturally involves agents (already identified in a previous fragment) and in so doing they play roles.

Reuse Guidelines

Composition

--

Dependency Relationship with other fragments

This fragment can be used when scenarios involving agents are available (for instance in terms of the outcome of a PASSI RID fragment)

Glossary

Message - an individual unit of communication between two or more agents that points out the standard FIPA message format. Usually a message is associated with a communicative act (or performative)

References

--