# Applying Process Document Standarization to INGENIAS

Alma Gómez-Rodríguez[1] and Juan C. González-Moreno[1]

Departamento de Informática (University of Vigo)
Ed. Politécnico, Campus As Lagoas,
Ourense E-32004 (SPAIN),
{alma,jcmoreno}@uvigo.es
http://gwai.ei.uvigo.es/

**Abstract.** The increasing interest on Agent Oriented Software Engineering in the last years is mainly due to its suitability for the design and implementation of huge, complex, distributed systems. In this field, special attention has been paid to development processes, because of direct correlation between the quality of the product and the process followed to obtain it. At the moment, there is neither a formal specification to define the activities to develop, the participants and the deliverables, nor guidance about the elements to introduce in the model or how these relate with each other. The use of standard notations may make it easier to describe the process, the resources and the mandatory deliverables. The IEEE FIPA Design Process Documentation and Fragmentation working group has proposed a template in order to cover this gap. This paper provides a first attempt at considering the results obtained by applying the proposed template to a well known development process proposed by the INGENIAS methodology for the construction of a multi-agent system.

## 1 Introduction

The software quality assurance discipline considers that the development process is very important because of the direct relation between process quality and final product quality. In particular, in Agent Oriented Software Engineering (AOSE) many methodologies and their associated processes of development have been proposed in the latest years [1–4]. All of them introduce all the conceptual abstractions that must be taken into account in any MultiAgent System (MAS) development.

Nevertheless, it is necessary to pay attention to the introduction of standards for the formal definition of these processes and methodologies.The use of standards provides better understanding and simplifies the task of sharing information among several groups of developers. At this moment, there is a ongoing work impelled by FIPA that proposes a template for standardizing methodological definition in AOSE field. The template provides a way of describing processes as well as some guidelines of how to use it. The detailed definition of the template is available at [5].

Following the key lines included in the template, this paper addresses the definition of a well known development process in the AOSE field. The process defined using the template is the one originally proposed by INGENIAS methodology [6–8] which is based on the Rational Unified Process (RUP)[9] also known as the Unified Development Process(UDP).

The remainder of the paper is organized according to the different sections of the template proposed by FIPA. This means that next section starts with an introduction to the methodology, with indication of its most relevant features. Section 3 details one of the phases defined by INGENIAS for MAS development and after, we introduce the definition of work-products dependencies. And finally, the paper ends with the conclusions obtained from template usage.

| | Analysis | Design |
|---|---|---|
| Phases | | |
| Inception | To generate use cases and identify actions of these use cases with the corresponding Interaction Model<br>To outline the system architecture with an Organisation Model<br>To generate Environment Models which reflects Requirement elicitation | To generate a prototype using RAD tools such as ZEUS or AgentTool |
| Elaboration | To refine use cases<br>To generate Agent Models that detail the elements of the system architecture<br>To continue with the Organisation Models, identifying workflows and tasks<br>To obtain Task and Goal Models to highlight control constraints (main goals, goal decomposition)<br>To refine the Environment Model including new elements | To focus the Organisation Model on workflow<br>To refine Tasks and Goal Models reflecting the dependencies and needs identified in workflows and the relationships with system's goals<br>To show how tasks are executed using Interaction Models<br>To generate Agent Models which show required mental state patterns |
| Construction | To study the remaining use cases | To generate new Agent models or refining existing ones<br>To study social relationships in order to refine the organisation |

**Table 1.** Phases and tasks for Ingenias Development Process

## 2 INGENIAS Process Documentation: Introduction

The INGENIAS methodology covers the analysis and design of MAS and it is intended for general use; that is, with no restrictions on application domain. It

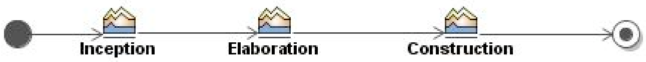

**Fig. 1.** Lifecycle for INGENIAS Methodology

has shown its capability and maturity as the supporting specification for the development of Multi-Agent Systems (MAS). The methodology provides the INGENIAS Development Kit (IDK), which contains a graphical editor for MAS specifications. Besides, the INGENIAS Agent Framework (IAF) [10], which is integrated in the IDK, enables a full model-driven development and transforms automatically specifications into code in the Java Agent DEvelopment (JADE) Framework.

Following the Rational Unified Process (RUP)[9], INGENIAS methodology distributes the tasks of analysis and design in three consecutive phases (see Fig. 1): *Inception*, *Elaboration* and *Construction*, with several iterations (where iteration means a complete cycle of development, which includes the performance of some analysis, design, implementation and proofs tasks). The sequence of iterations leads to the procurement of the final system.

The process of such development process is often represented by its authors in a tabular form (see Table 1). In the table, the three development phases are presented jointly with two different types of workflows for *Analysis* and *Design*. The methodology pays few attention, compared to RUP, to *Implementation* and *Test workflows*, because it provides some tools which automatically generate code, in parallel with system's specification. Attending this facility, these workflows are considered not to be modeled as a fundamental part of the process.

INGENIAS tries to follow a Model Driven Development (MDD) [11] approach, so it is based on the definition of a set of meta-models that describe the elements that could be used to specify a MAS following five viewpoints [12]:

1. **Agent**: It specifies the definition, control and management of each agent mental state
2. **Interaction**: The model is used to describe the agents' interactions
3. **Organization**: It details MAS architecture
4. **Environment**: This viewpoint is used to model the environment of the MAS
5. **Task and goals**: This meta-model present a detailed view of the tasks and goals assigned to each agent

The development process is supported by a set of tools, which are generated from the meta-models specification by means of a meta-modeling processor (which is the core of the IDK). MAS modeling is facilitated by a graphical editor and a verification tool. The methodology has been used in several examples from different domains, such as PC management, stock market, word-processor assistant, and specially collaborative filtering information systems.

Detailed references of the methodology from their authors can be found in [6, 7, 12].

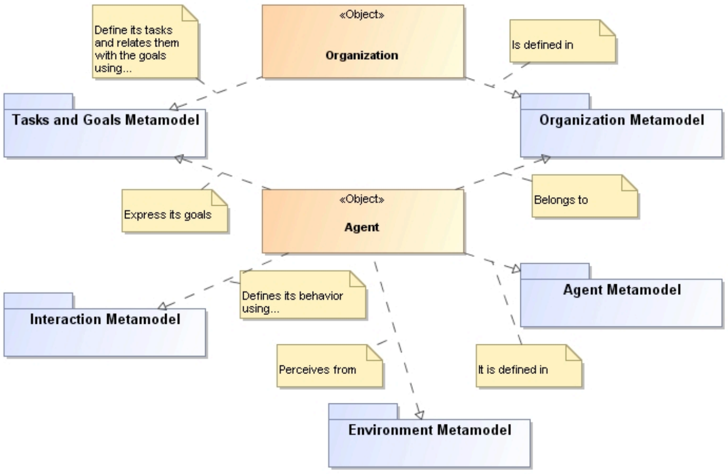

**Fig. 2.** Global meta-model of INGENIAS Methodology

### 2.1 The INGENIAS Process lifecycle

As pointed in the introduction INGENIAS methodology distributes the tasks of analysis and design in three consecutive phases: *Inception*, *Elaboration* and *Construction*. Each phase may have several iterations (where iteration means a complete cycle of development)[1].

Following the idea proposed by RUP to take the system architecture as guideline for development, INGENIAS propose the use of the Organization Model as basis for the MAS definition and construction (see Table 1)

### 2.2 The INGENIAS Meta-model

From the point of view of INGENIAS, a meta-model defines the primitives and the syntactic and semantic properties of a model. Following this idea the methodology provides five meta-models that define five different views of the system.

An important characterictis of INGENIAS meta-models is that they are quite detailed fine grained). This is due to that they are intended to be a precise definition of the system, and also because each meta-model is the basis for the automatic code generation provided by *the INGENIAS Development Kit* (IDK).

As an example of how meta-models are detailed in INGENIAS, Fig. 3 shows the graphical definition of of the Agent Meta-model. An agent is identified as an autonomous entity, with particular goals and a unique identity. Three fundamental elements are identified for each agent: *the roles* the agent must

[1] the definition of INGENIAS can be found in `http://www.pa.icar.cnr.it/cossentino/fipa-dpdf-wg/docs/INGENIAS.pdf`

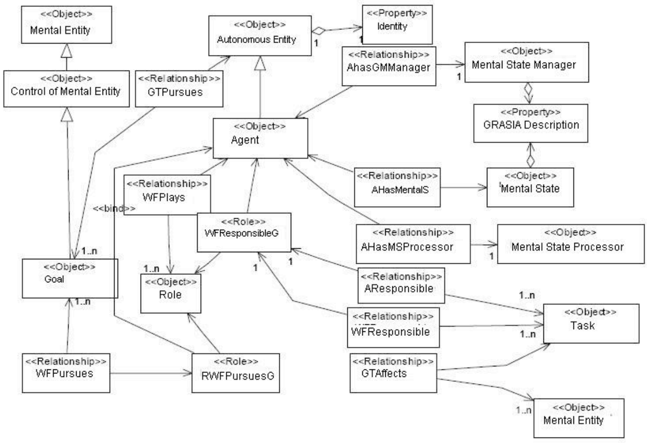

**Fig. 3.** Agent meta-model proposed by INGENIAS Methodology

play, *the tasks* the agent must accomplish and its *mental state*. The relationships among them show how an agent can pursue its goals and how it achieves that goals executing a particular tasks. This meta-model is selected because the entities (*Agents* and *Roles*) are also included in other MAS meta-models.

### 2.3 Definition of MAS meta-model elements

In table 2, the basic elements taken from the meta-model are introduced. As the meta-models of INGENIAS are very detailed, only the most important concepts have been defined. For further details, the original documentation of the methodology must be reviewed [6, 7, 12]

## 3 INGENIAS Process Documentation: Inception Phase

Following the recommendation of the template each phase must be specified in a section, due to space limitations this paper is focused on the first phase proposed by the methodology. According to INGENIAS, meta-models are a key issue in the MAS development, but these models must be associated to the activities done to obtain them. This integration is the key point covered on the specification introduced in this section.

Figure 4 gives a general view of the INGENIAS Inception Phase. The methodology considers that the development process is initiated from the document describing the problem, so that, this can be considered the initial input of the process. From this document, the Inception phase introduces several activities that are described in Fig. 4. Regarding the Analysis workflow at this level the activities are:

| Concept | Definition | Cross-References |
|---|---|---|
| Agent | An agent entity is an autonomous entity with identity, purposes and that performs activities to achieve its goals | Autonomous Entity |
| Application | An application is a wrapper to computational system entities. Computational represents a system having an interface and a concrete behavior | |
| Autonomous Entity | Root concept that represents an entity with identity and that pursues goals | Goal |
| Goal | According to the BDI model, a goal is a desired state that an agent wants to reach. In planning, a goal is represented by a world state. Here a goal is an entity by itself, however it can be related with a representation of the world state using satisfaction relationships with tasks. This relationships contains references to descriptions of mental states of agents, so they refer to the image of the world that agent have | Agent |
| Interaction | Interaction represents an exchange between two or more agents or roles. There can be only one initiator and at least one collaborator. An interaction also details the goal that pursues. This goal should be related with the goals of the participants. | Agent, Role & Goal |
| MentalState | A mental state represents the information an agent has in a certain moment. A MentalState is an aggregate of mental entities. | Agent |
| Organisation | An organisation is a set of agents, roles and resources that get together to achieve one or several goals. Inside an organisation there are not other organisations, just groups. You can think of an organisation as an enterprise. Internally it is composed by departments that may be restructured without affecting the external image of an enterprise. | Agent |
| Resource | Resource describes a resource according to TAEMS notation. Opposite to TAEMS, there is no distinction between consumable and non-consumable resources. | |
| Role | A role is a self-contained grouping of functionalities. When an agent plays a role we want to express that you have to execute tasks associated to a role and participate in the same interactions that role | Agent |
| Task | Tasks is the encapsulation of actions or non-distributable algorithms. Tasks can use Applications and resources. Tasks generate changes in the mental state of the agent that executes them. Changes consist of: (a) modifying, creating or destroying mental entities; or (b) changes in the perception of the world by acting over applications (applications act over the world producing events, that are perceived by the agent). Though tasks can be also assigned to roles, at the end, it will belong to an agent | Role |
| Workflow | A workflow is an abstraction to a process that has been automatised using activities and identifying responsibility relationships | |

**Table 2.** Definition of MAS meta-model Elements

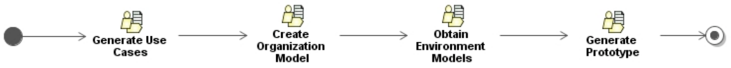

**Fig. 4.** Activities and workflows of Inception phase proposed by INGENIAS Methodology

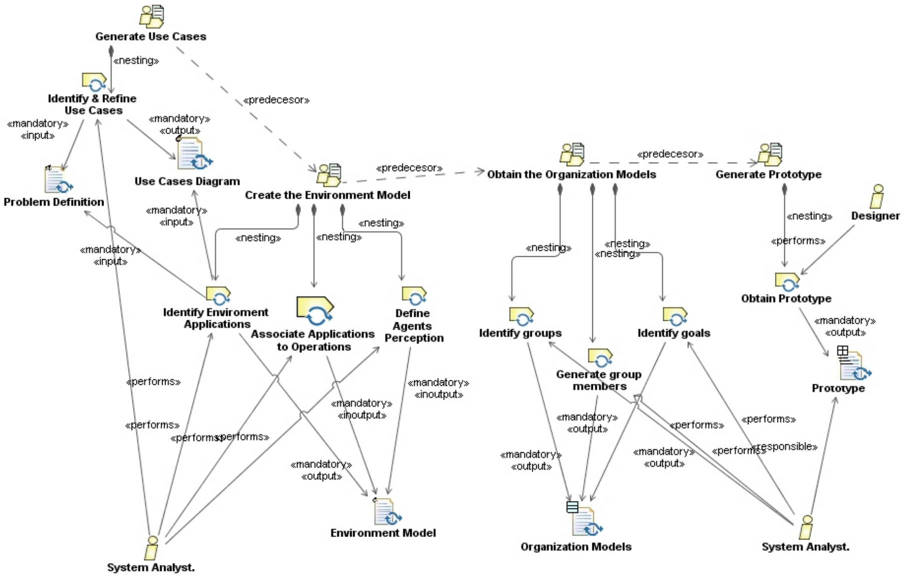

**Fig. 5.** Detailed tasks of Inception activities

- Generate Use Cases
- Initiate the architecture using the Organization Model
- Generate the Environment Model

In what respects to Design only the construction of a rapid prototype must be addressed.

All these activities and the tasks associated to each of them are shown in Fig. 5. From this figure, we can identify the different tasks proposed by INGENIAS for Inception and the produced work-products. Moreover, the roles responsible of each task as well as the kind of responsibility they assume are also shown.

### 3.1 Process roles

The template says that the roles that are responsible of each task must be identified. Nevertheless, INGENIAS methodology makes no explicit reference to the roles implied in the development. We consider roles identification proposed by the template very helpful because it solves a problem previously detected when using the methodology in real environments. In some cases, the team members have difficulties to known what activity or task they must done and what their responsibilities where according to the process.

To state the roles involved in this phase, it has been take into consideration that INGENIAS follow RUP development and it has been considered also the activities to be done and the level of abstraction of such activities. From this analysis, we propose only two roles to participate in this phase: the *System Analyst* and the *Designer*.

The System Analyst is responsible or performs the most part of the activities proposed in this phase. In particular, he will:

- *Identify the Use Cases* and *construct and refine the Use Cases Diagram*. From the initial description of the problem to solve, the analyst must obtain the use cases that will guide after the creation of the Interaction Model.
- Define the Environment Model, showing the interaction of the system with its environment. This will imply to: *identify applications* (in INGENIAS, all the software and hardware that interact with the system and can't be designed as an agent will be considered an application); *associate operations to particular applications* and *define agents perception on applications*.
- *Obtain the Architectural view* of the System using the Organization Model. This means to generate a structural definition of the system by *identify groups* in the organization, *generate group members* and *identify goals*.

The second role identified: the *Designer* must be responsible of *generating the prototypes*. According to INGENIAS literature, this will be done using a rapid application development tool such as ZEUS, Agent Tool or others.

### 3.2 Activity Details

Following the template recommendation [5], this section details the activities previously outlined for Inception Phase.

**Generate Use Cases**.- The generation and refining of Use Cases has been identified as a unique task. The goal of this task will be to identify the intended functioning of the system. Knowing the functionalities the system must provide, will allow to identify interaction collaborators and initiators and also to discover the nature of such interactions that will affect the type of control applied to the agent: *planning*, *cooperation*, *contract-net* or *competition*.

**Generate the Environment Model**.- The Environment Model tries to show the elements that constitute the environment of the system, and in consequence, the perceptions of the agents. The elements defined in this model are of three basic kinds: *agents*, *resources* and *applications*.

Figure 6 shows the task to be accomplished for obtaining an Environment Model of the system to construct. These task are further explained in Table 3.

**Initiate the architecture**.- One of the key activities in Inception Phase is to start the definition of system architecture. This is done by constructing the Organization Model, which reflects mainly the system's workflows.

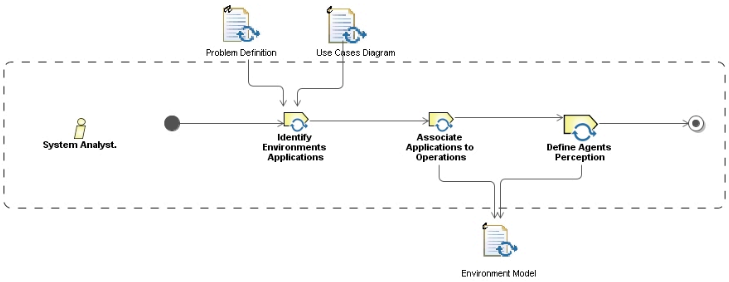

**Fig. 6.** Obtaining the Environment Model in the Inception Phase of INGENIAS

| Activity | Task | Description | Roles Involved |
|---|---|---|---|
| Generate the Environment Model | Identify the Environment Applications | All the software and hardware that interact with the system and that can not be designed following an agent oriented approach will be considered an application | System Analyst |
| Generate the Environment Model | Associate the Applications and Operations | Operations are associated to the applications defined by requirements. These operations have a signature, a precondition and a postcondition. The identification of operations is a conventional engineering task. | System Analyst |
| Generate the Environment Model | Define Agents Perception | The main aim of this task is to define agents perception on environment applications, at this moment of process it is enough to relate agents and applications | System Analyst |

**Table 3.** Task of Activity Generate the Environment Model of Inception Phase of INGENIAS

In Figure 7 the basic tasks related with the procurement of Organization Model in Inception activity are shown. These activities try to obtain an organizational view of the system, attending its structural, functional and social aspects. The detailed definition of tasks are addressed in Table 4.

**Construction of a prototype**.- The generation of a prototype is a unique and simple task, that is supposed to be done using a RAD tool.

### 3.3 Work Products

The last aspect to be covered to complete the specification of a phase following the template recommendation [5] is to detail the Work products used. The INGENIAS Inception Phase produces as result four basic work-products: a *Use Cases diagram*, an *Environment Model*, one or more *Organization Models*

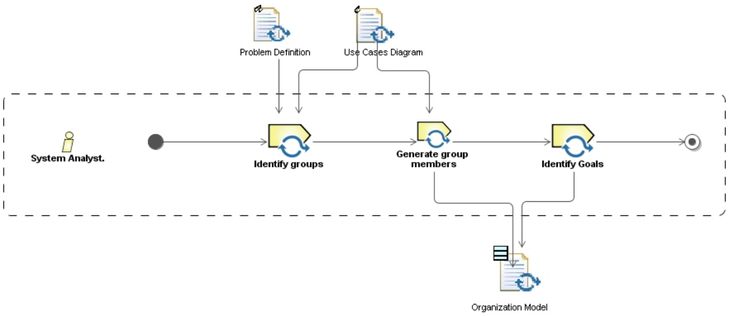

**Fig. 7.** Obtaining the Organization model in the Inception Phase of INGENIAS

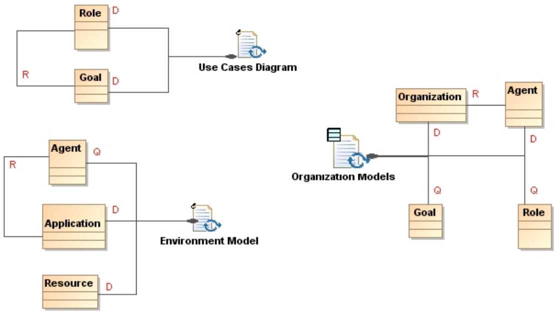

**Fig. 8.** Structure of Inception Work-products

and a *prototype of the system* to be built. The relationships among the models and the the meta-model elements are shown in Figure 8. Organization model, for instance, defines the organization meta-model element and the agents and uses the roles and goals previously defined. In this particular case, organization concept includes also the groups within the system (see organization definition in table 2). On the other hand, the Environment Model defines the internal and external applications the system interacts with, as well as the resources available.

## 4 INGENIAS Process Documentation: Work-product dependencies

Following the template [5], a final aspect that must be detailed after the specification of the process phases is the Work-product dependencies. Figure 9 introduces a global view of INGENIAS work-products, as well as their dependencies. As shown in the Figure, *Agent Model* depends on *Organization* and *Environment Models*, while the *Interaction Model* shows dependencies from *Agent* and *Task/Goal Models* among others.

| Activity | Task | Description | Roles Involved |
|---|---|---|---|
| Obtain the Organization Model | Identify groups | The groups in the system must be identified. In this way the participants in a particular work flow will be organized. | System Analyst |
| Obtain the Organization Model | Generate group members | Members (agents, roles, resources and applications) are assigned to groups creating the corresponding relationships. If needed, the groups can be decomposed in order to reduce complexity. | System Analyst |
| Obtain the Organization Model | Identify groups | The organization has a set of goals that must justify collaboration between agents. The goals identified in this task will after be assigned to individual agents or roles in the Task and Goals Model. | System Analyst |

**Table 4.** Task of Activity Initiate Architecture of Inception Phase of INGENIAS

## 5 Conclusions and Further Work

Most times a methodology proposes a particular development process in its description. This process may be common to different methodologies, but it is not specified using a common notation. This gap is being covered by the IEEE FIPA Design Process Documentation and Fragmentation working group with the proposal of a template for its definition. This paper provides a first attempt at the application of the proposed template to model the original development process of INGENIAS methodology.

The standard has been very useful for the definition an the results have been satisfactory. Thanks to the use of the the template jointly with the standard notation, we have been able to improve some how the definition of INGENIAS RUP based process. For instance, we have identified roles and responsibilities for each of the tasks that were not previously defined. Moreover, the dependencies among work-products indicate some relationships that must be taken into account when adapting the methodology to Agile development processes.

**Acknowledgements** This work has been supported by the project *Novos entornos colaborativos para o ensino* supported by Xunta de Galicia with grant 08SIN009305PR.

## References

1. Cuesta, P., Gómez, A., González, J., Rodríguez, F.J.: The MESMA methodology for agent-oriented software engineering. In: Proceedings of First International Workshop on Practical Applications of Agents and Multiagent Systems (IWPAAMS'2002). (2002) 87–98
2. Bernon, C., Cossentino, M., Pavón, J.: Agent-oriented software engineering. Knowl. Eng. Rev. **20** (2005) 99–116

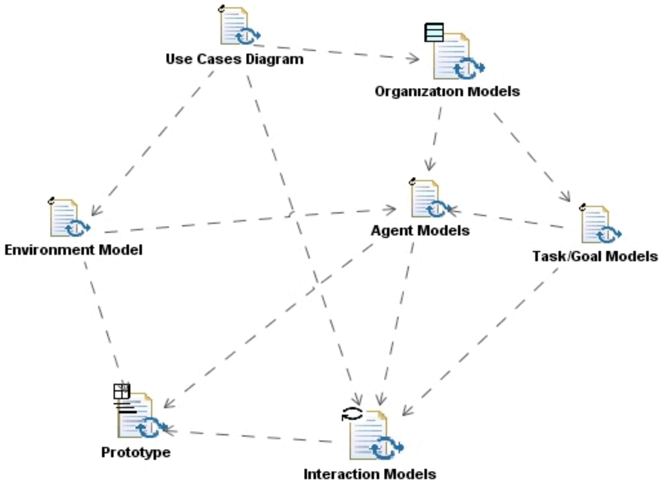

**Fig. 9.** Dependencies among INGENIAS Work-products

3. Federico Bergenti, Marie-pierre Gleizes, F.: Methodologies And Software Engineering For Agent Systems: The Agent-oriented Software Engineering Handbook. Springer (2004)
4. Henderson-Sellers, B., Giorgini, P.: Agent-oriented methodologies / Brian Henderson-Sellers, Paolo Giorgini. Idea Group Pub., Hershey, PA (2005)
5. IEEE FIPA Design Process Documentation and Fragmentation: IEEE FIPA Design Process Documentation and Fragmentation Homepage. http://www.pa.icar.cnr.it/cossentino/fipa-dpdf-wg/ (2009)
6. Gómez-Sanz, J.: Modelado de Sistemas Multi-agente. PhD thesis, Universidad Complutense de Madrid. Facultad de Informática (2002)
7. Gómez-Sanz, J.J., Pavón, J.: Meta-modelling in agent oriented software engineering. In: Advances in Artificial Intelligence - IBERAMIA 2002, 8th Ibero-American Conference on AI, Seville, Spain, November 12-15, 2002, Proceedings. Volume 2527 of Lecture Notes in Computer Science. (2002) 606–615
8. Pavón, J., Gómez-Sanz, J.: Agent Oriented Software Engineering with INGENIAS. Multi-Agent Systems and Applications III **2691** (2003) 394–403
9. Rational Software: Rational Unified Process: White Paper (1998)
10. Gómez-Sanz, J.: Ingenias Agent Framework. Development Guide V. 1.0. Technical report, Universidad Complutense de Madrid. (2008)
11. Atkinson, C., Kuhne, T.: Model-driven development: a metamodeling foundation. Software, IEEE **20** (2003) 36–41
12. Grupo de Investigación en Agentes Software: Ingeniería y Aplicaciones. INGENIAS Section. `http://grasia.fdi.ucm.es/main/?q=es/node/61` (2010)