ROMAS Methodology

Emilia Garcia, Adriana Giret, and Vicente Botti

Abstract

This chapter presents the ROMAS methodology. ROMAS is an agent-oriented methodology that guides developers on the analysis and design of *regulated open multiagent systems*. These kinds of systems are composed of heterogeneous and autonomous agents and institutions, which may need to coexist in a complex social and legal framework that can evolve to address the different and often conflicting objectives of the many stakeholders involved. Contracts and norms are used to formalize the normative context and to establish restrictions on the entities' behaviors.

1 Introduction

ROMAS methodology defines a set of activities for the analysis and design of *regulated open multiagent systems*. The most relevant characteristics of systems of this kind are that they are *social*, *open*, and *regulated*. First, they are social in the sense that autonomous and heterogeneous entities interact between themselves to achieve global and individual objectives. Besides, the entities of the system can be structured as institutions or groups of agents with similar characteristics, functionality, or that interact with the rest of the system as a single entity. Second, they are open in the sense that, dynamically at runtime, external parties can interact and become part of the system. Third, they are regulated in the sense that every entity or institution in the system can have associated a set of norms that must fulfill. Besides, the expected behavior of each entity should be clearly specified by means of defining its rights and duties inside the system. In ROMAS, we consider the *normative context* of a system to be the set of norms that regulates the behavior of

E. Garcia • A. Giret • V. Botti (✉)
Universitat Politecnica de Valencia, Valencia, Spain
e-mail: mgarcia@dsic.upv.es; agiret@dsic.upv.es; vbotti@dsic.upv.es

M. Cossentino et al. (eds.), *Handbook on Agent-Oriented Design Processes*, DOI 10.1007/978-3-642-39975-6__11,
© Springer-Verlag Berlin Heidelberg 2014

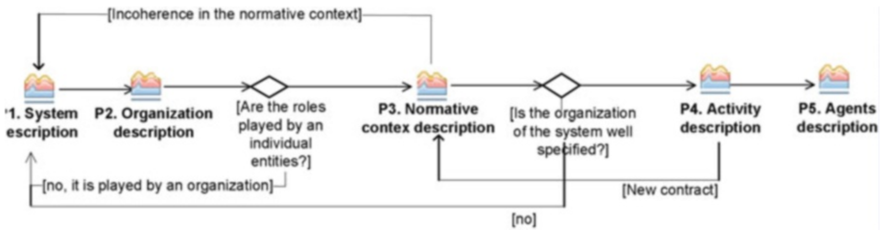

[Incoherence in the normative context]
'1. System escription
P2. Organization description
[Are the roles played by an individual entities?]
P3. Normative contex description
[Is the organization of the system well specified?]
P4. Activity description
P5. Agents description
[no, it is played by an organization]
[New contract]
[no]

Fig. 1 ROMAS design process

each entity and the set of contracts that formalizes the relationships between these entities. The *normative context* of each entity is specified by the set of norms that directly affects the behavior of a particular entity. ROMAS catchs the requirements of the design of systems from the global system's purposes to the specification of the behavior of each individual entity. The rest of the chapter presents the ROMAS metamodel and phases. Following some useful references about ROMAS:

- Emilia Garcia, A. Giret and V. Botti *Regulated Open multiagent Systems based on contracts* The 19th International Conference on Information Systems Development (ISD 2010) pp. 235–246. (2010)
- Emilia Garcia, A. Giret and V. Botti *A Model-Driven CASE tool for Developing and Verifying Regulated Open MAS* Journal Science of Computer Programming (2011)
- Emilia Garcia, G. Tyson, S. Miles, M. Luck, A. Taweel, T. Van Staa and B. Delaney *An Analysis of Agent-Oriented Engineering of e-Health Systems* 13th International Workshop on Agent-Oriented Software Engineering (AOSE–AAMAS) (2012)

1.1 The ROMAS Process Lifecycle

ROMAS methodology is composed of five phases that help developers to analyze and design the system from the highest level of abstraction to the definition of individual entities. As shown in Fig. 1, this is not a linear process but an iterative one. The identification of a new element of functionality during one phase may imply the revision of previous phases. For example, during the second phase, when a role that can be played by an organization as a whole is detected, it is necessary to go back to the first phase of the methodology to analyze the characteristics, global objectives, and structure of this organization.

1.2 ROMAS Metamodel

The analysis and design of these systems are formalized by means of several diagrams that are instances of the ROMAS metamodel. Table 1 describes the ROMAS metamodel elements. In order to facilitate the modeling tasks, this unified metamodel can be instantiated by means of four different views that are described below:

Organizational View

In this view, the global goals of organizations and the functionality that organizations provide and require from their environment are defined (Fig. 2). The static components of the organization, that is, all elements that are independent of the final executing entities, are defined too. More specifically, it defines the following:

– The entities of the system (*Executer*): *AAgents* and *Roles*. The classes *Executer* and *AAgents* are abstractions used to specify the metamodel, but neither of them are used by designers to model systems.
– An *AAgent* is an abstract entity that represents an atomic entity (*Agent*) or a group of members of the organization (*Organizational Unit*), which is seen as a unique entity from outside.
– The *Organizational Units* (OUs) of the system can also include other units in a recursive way as well as single agents. The *Contains relationships* include conditions for enabling a dynamical registration/deregistration of the elements of an OU through its lifetime.
– The global *Objectives* of the main organization. The objectives defined in this view are nonfunctional requirements (soft goals) that are defined to describe the global behavior of the organization.
– The *Roles* defined inside the OUs. In the *contains* relationship, a minimum and maximum quantity of entities that can acquire a particular role can be specified. For each role, the *Accessibility* attribute indicates whether a role can be adopted by an entity on demand (external) or it is always predefined by design (internal). The *Visibility* attribute indicates whether entities can obtain information from this role on demand. This attribute can take the value “public” if anyone can obtain information of this role, and it takes the value “private” if only members of this organizational unit (i.e., private role). A hierarchy of roles can also be defined with the *InheritanceOf* relationship.
– The organization’s social relationships (*RelSocialRelationship*). The type of social relationship between two entities is related to their position in the structure of the organization (i.e., information, monitoring, supervision), but other types are also possible. Some social relationships can have a *ContractTemplate* associated with them, which can formalize some predefined commitments and rights that must be accepted or negotiated during the execution time. Each *Contract Template* is defined using the Contract Template view.
– The *Stakeholders* interact with the organization by means of publishing offers and demands of *Products* and *Services* on the *Bulletin Board*.
– The *Bulletin Board* can be considered as an information artifact for Open MAS. This artifact allows the designer to define the interaction with external entities and

Table 1 Definition of ROMAS metamodel elements

Concept	Definition	Metamodel views
Objective	An objective is a specific goal that agents or roles have to fulfill. It can be refined into other objectives.	Organizational, internal view
Organizational Unit (OU)	A set of agents that carry out some specific and differentiated activities or tasks by following a predefined pattern of cooperation and communication. An OU is formed by different entities throughout its lifecycle, which can be either single agents or other organizational units that are viewed as a single entity.	Organizational, internal, contract template
Role	An entity representing part of the functionality of the system. Any entity that plays a role within an organization acquires a set of rights and duties.	Organizational, internal, contract template, activity
Agent	An entity capable of perceiving and acting into an environment, communicating with other agents, providing and requesting services/resources, and playing several roles.	Organizational, internal, contract template, activity
Norm	A restriction on the behavior of one or more entities.	Organizational, internal, contract template, activity
Contract template	A set of predefined features and restrictions that all final contract of a specific type must fulfill. A contract represents a set of rights and duties that are accepted by the parties.	Organizational, internal, contract template, activity
Bulletin Board	A service publication point that offers the chance of registering and searching for services by their profile.	Organizational, internal, contract template, activity
Product	An application or a resource.	Organizational, internal, contract template, activity
Service Profile	The description of a service that the agent might offer to other entities.	Organizational, internal, activity
Service Imple-mentation	A service-specific functionality that describes a concrete implementation of a service profile.	Internal, activity
Task	An entity that represents a basic functionality that consumes resources and produces changes in the agent's mental state.	Organizational, internal, contract template, activity
Stakeholder	A group that the organization is oriented toward and interacts with the OUs.	Organizational
Believe	A claim that an agent thinks that it is true.	Internal
Fact	A claim that is true at the system's domain.	Internal
Event	The result of an action that changes the state of the system when it occurs.	Internal
Interaction	An entity defining an interaction between agents.	Activity
Interaction Unit	A performative employed during the interaction.	Activity
Translation Condition	An artifact that allows to define the sequence of tasks depending on a condition.	Activity
Executer	A participant in an interaction. It can be an Organization, an Agent, or a Role.	Organizational, internal, contract template, activity

facilitates trading processes. When an agent wants to trade, they can consult or publish their offer on the Bulletin Board. Each offer or demand can be associated with a Contract Template. It means that this offer or demand has some predefined restrictions that are specified in this Contract Template view.

Internal View

This view allows defining the internal functionality, capabilities, belief, and objectives of each entity (organizations, agents, and roles) by means of different instances of this model (Fig. 3). More specifically, it defines the following features of each entity:

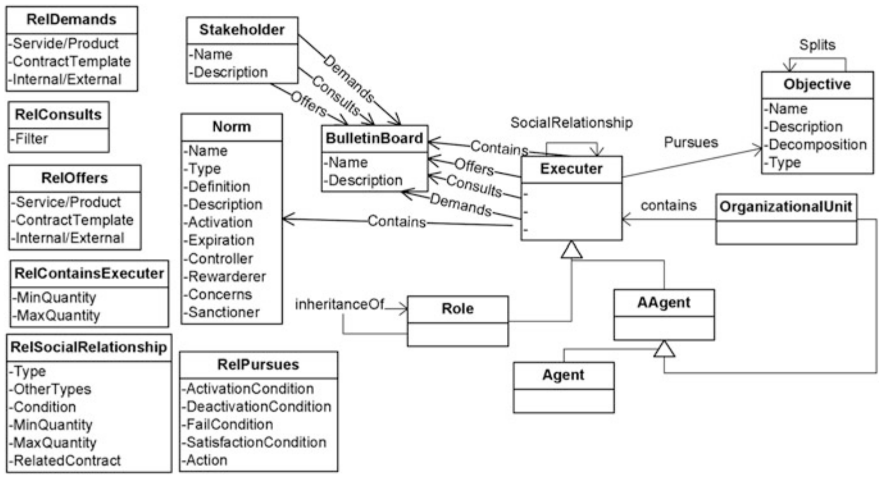


Fig. 2 Organizational view (the class *RelXXX* represents the attributes of the relationship *XXX*)

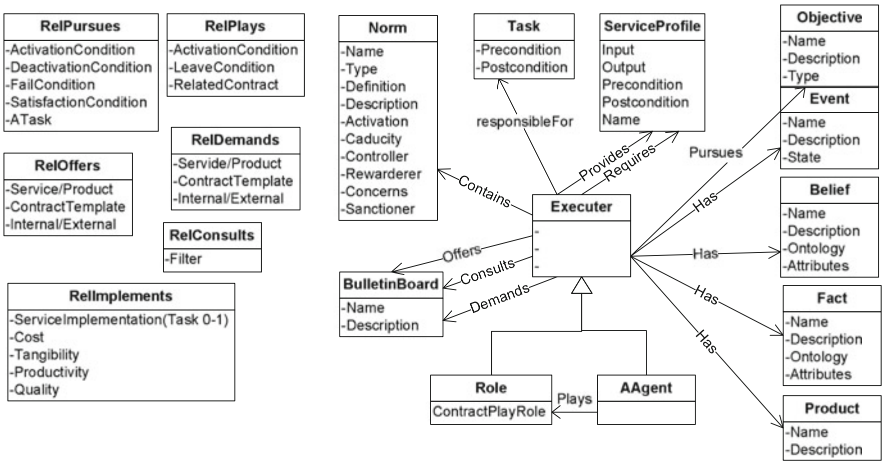


Fig. 3 Internal view (the class *RelXXX* represents the attributes of the relationship *XXX*)

– The *Objectives* represent the operational goals, that is, the specific goals that agents or roles have to fulfill. They can also be refined into more specific objectives. They might be related to a Task or Interaction that is needed for satisfying this objective.
– The *Mental States* of the agent, using belief, events, and facts.
– The *products* (resources/applications) available by an OU.
– The *tasks* that the agent is responsible for, that is, the set of tasks that the agent is capable of carrying out. A task is an entity that represents a basic functionality that consumes resources and produces changes in the agent's Mental State.

- the *Implements Service Profile*
- Internal entities can publish offers and demands in a *BulletinBoard*, as external stakeholders can do by means of the organizational view. This publication can also have an associated Contract Template to describe some predefined specifications.
- the *roles* that an agent or an organizational unit may play inside other organizational units (*Plays* relationship). *ActivationCondition* and *LeaveCondition* attributes of this relationship indicate in which situation an OU acquires or leaves a role.
- the *roles* played by each agent. *ActivationCondition* and *LeaveCondition* attributes of this *play* relationship indicate in which situation an agent can acquire or leave a role.
- the *Norms* specify restrictions on the behavior of the system entities. The relationship *Contains Norm* allows defining the rules of an organization and which norms are applied to each agent or role. *Norms* control the global behavior of the members of the OU.

Contract Template View

This view allows defining *Contract Templates*. Contracts are inherently defined at runtime. Despite this, designers represent some predefined restrictions that all final contracts of a specific type should follow by means of a *contract template*. Contract templates can be used at runtime as an initial point for the negotiation of contracts and to verify if the final contract is coherent with the legal context. The syntax of a contract template is defined in Fig. 4. More specifically, it defines the following:

- The relationship *Signants* indicates who is allowed to sign this type of contracts. It could be a specific agent, an agent who plays a specific role, or an organization. A *ThirdPart* could be anyone who participates in the negotiation protocol or who is affected by the final execution of the Contract.
- The relationship *Protocol* indicates which protocols are recommended to negotiate this type of contract.
- After the negotiation, the *Notary* is responsible for verifying the correctness and coherence of the final contract definition. He should check if any term of a contract violates any norm of the regulated environment.
- Each type of contract can define which *Receipts* will be generated during the execution time. Receipts are proof of facts; for example, a receipt can be generated when an agent successfully provides a service.
- In case of conflict, the *Judge* has to evaluate the Complaints and the generated Receipts following the *ConflictResolution* protocol. If he decides that there has been a violation of a norm, the *RegulationAuthority*, who is the main authority in the context of a contract, can punish or reward the agent behaviors.
- The relationship *Hard clause* indicates that any instance of this type of contract has to include this norm. *Soft clauses* are recommendations, so during the negotiation stage, *Signants* will decide whether this norm will be included or not in the final contract.

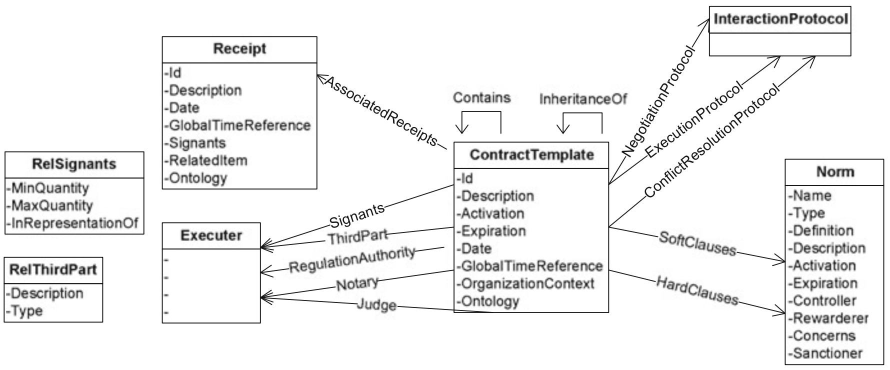


Fig. 4 Contract template view (the class *RelXXX* represents the attributes of the relationship *XXX*)

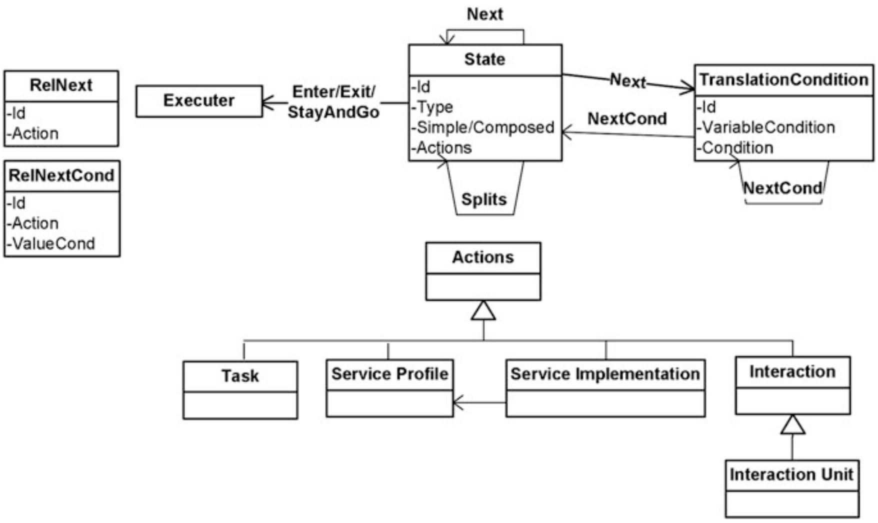


Fig. 5 Activity view (the class *RelXXX* represents the attributes of the relationship *XXX*)

Activity View

This view allows defining the sequence of actions in which a task, a service, or a protocol can be decomposed (Fig. 5). Each *state* represents an action or a set of actions that must be executed. An *action* is a first-order formula that indicates which task or service is executed or which message is interchanged between the agents that participate in this state. The relationship *next* indicates the sequence of states. These sequences can be affected by a *translation condition* that indicates under which circumstances the a state is going to be the next step of the process.

1.2.1 ROMAS Metamodel Notation

ROMAS diagrams use an UML-like graphical notation called GOPPR [6]. A caption to understand the graphical elements of the diagram is shown in Fig. 6. This notation

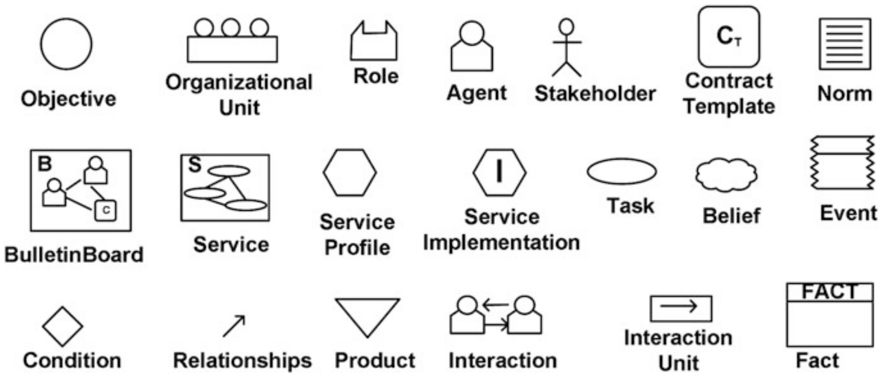


Fig. 6 Entities from the ROMAS graphical notation

is also used to define diagrams on INGENIAS, GORMAS, and ANEMONA methodologies. Some metamodel constructions proposed by ROMAS, such as *contract templates*, have been added to the notation.

2 Phases of the ROMAS Process

In this section, the phases that compose the ROMAS methodology are described. ROMAS offers a set of guidelines to analyze and formalize the requirements of the system, its functionality, the social relationship between the entities, and their normative context. These guidelines also guide designers to specify the interactions and service interchanges between the entities of an organization and between external entities.

2.1 PHASE 1: System Description

During this phase, the analysis of the system requirements, global goals of the system, and the identification of use cases are carried out. Besides, the global goals of the organization are refined into more specific goals, which represent both functional and nonfunctional requirements that should be achieved. Finally, the suitability of the ROMAS methodology for the specific system to develop is analyzed.

2.1.1 Process Roles

There are two roles involved in this phase: the system analyst and the domain expert. The *domain expert* is responsible for: (1) describing the system requirements, by means of identifying the system's main objectives, the stakeholders, the environment of the organization, and its restrictions; (2) supporting the system analyst in

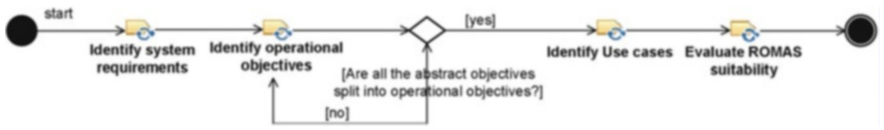

start
Identify system requirements
Identify operational objectives
[yes]
[Are all the abstract objectives split into operational objectives?]
[no]
Identify Use cases
Evaluate ROMAS suitability

Fig. 7 Phase 1: activity tasks

Table 2 Phase 1: activity tasks

ID.task	Task	Description	Roles involved
1.1	Identify system requirements	Following the guideline *system description document*, the requirements of the system are analyzed, including global objectives of the system, stakeholders that interact with the system, products and services are offered and demands to/from stakeholders, external events that the system handles and normative documents such governmental laws attached to the system.	Domain expert
1.2	Identify Operational Objectives	Following the guideline *objective description document*, the global objectives of the system are analyzed and split into operational objectives, i.e., into more low level objectives that can be achieved by means of the execution of a task or a protocol.	System analyst and domain expert
1.3	Identify use cases	Using the information obtained in the previous task, the use cases of the system regarding the tasks and protocols associated to the operational objectives identified are defined.	System analyst and domain expert
1.4	Evaluate ROMAS suitability	Following the guideline *ROMAS suitability guideline*, the suitability of the ROMAS methodology for the development of the system to be developed regarding its specific features is evaluated.	System analyst

the analysis of the objectives of the system; and (3) supporting the system analyst in the description of the use cases of the system. The *system analyst* is responsible for: (1) analyzing the objectives of the system; (2) identifying the use cases; and (3) evaluating the suitability of the ROMAS methodology for the system to be developed regarding its requirements.

2.1.2 Activity Details

The flow of tasks of this phase is reported in Fig. 7, and each task is detailed in Table 2.

2.1.3 Work Products

The following section describes the products generated on the *System definition* phase and the guidelines used to define them: (1) *System definition document*; (2) *Objectives description document*, (3) *Use case diagram*, (4) *ROMAS suitability guideline*. Figure 8 shows graphically the products used and produced by each task.

System Description Document

This document is employed to identify the main features of the system and its relationship with the environment. It is a structured text document whose template is shown in Table 3. Table 4 presents the analysis of CMS case study using this document.

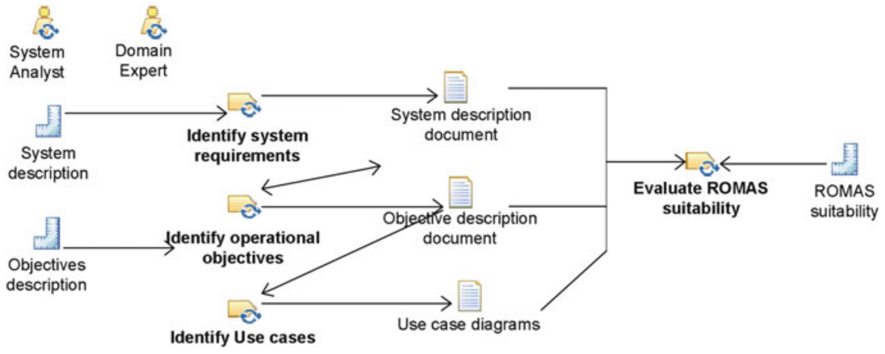

System Analyst
Domain Expert
System description
Identify system requirements
System description document
Evaluate ROMAS suitability
ROMAS suitability
Objectives description
Identify operational objectives
Objective description document
Identify Use cases
Use case diagrams

Fig. 8 Phase 1: resources and products used

Objectives Description Document and Objective Decomposition Diagram

This document analyzes the global objectives of the system and decomposes them into operational objectives. It is a structured text document whose template is shown in Table 5. Every global objective specified in the *system description document* is described using this document. The global objectives of the systems are refined into more specific ones that should also be described using this document. The document will be completed when all the global objectives are decomposed into operational objectives, that is they are associated to tasks, protocols, or restrictions that must be fulfilled in order to achieve these objectives. It is recommended to create one table for each global objective. The first column of each table will contain the property's name, the second the description of the global objective, and the following columns the descriptions of the objectives in which this global objective has been decomposed.

It is recommended to graphically represent the decomposition of the objectives by means of a diagram in order to provide a general overview of the purpose of the system that can be easily understood by domain experts. The graphical overview of the CMS case study objectives is shown in Fig. 9, where A means abstract objective and O means operational objective. As an example of the decomposition of a global objective into operational ones, Table 6 shows the decomposition of the global objective *Conference registration*.

Use Case Diagram

These diagrams are UML graphical representations of workflows of stepwise activities and actions with support for choice, iteration, and concurrency. The actions identified in the analysis of the operational objectives are related, forming activity diagrams in order to clarify the sequence of actions that will be performed in the system. Figure 10 shows the sequence of actions that can be performed in the CMS case study.

ROMAS Suitability Guideline

After analyzing the requirements of the system, it is recommended to use this guideline in order to evaluate the suitability of the ROMAS methodology for the

Table 3 Template for system description document

Property	Description		Guideline
System identifier	General name of the system to be developed.		It is recommended to select a short name or an abbreviation.
System description	Informal description of the system.		There is no limitation on the length of this text. - What is the motivation for developing such a system? -Is there any system requirement that specify if the system must be centralized or decentralized? - Which is the main objective of this system?
Domain	Domain or domains of application.		If this system must be able to be applied in different domains, it is recommended to add a text that explains each domain and whether it is necessary to adapt the system to each domain.
Kind of environment	Identify and specify the kind of environment of the system.		- Can the functionality of the system be distributed between different entities? - Are the resources of the system distributed in different locations? - Are here external events that affect the internal state and behavior of the system? Is it a reactive system? - Is it a physical or a virtual environment? Is there any physical agent or robot that plays a role in the system? - Is there any human interaction with the system? - Should the results of the system be presented graphically? Is there any graphical environment?
Global objectives	Functional and non-functional requirements (softgoals) that specify the desired-global behavior of the system.		- Which are the purposes of the system? - Which results should provide the system? - Should the system keep any parameter of the system between an specific threshold? (ex. the temperature of the room, the quantity of money in an account and so on)
Stakeholders	Identifier	An identifier for the stakeholder.	Are there external entities or applications that are able to interact with the system?
	Description	Informal description of the stakeholder.	
	Type	Indicate if the stakeholder is a client, a provider or a regulator.	
	Contribution	To point out what the organization obtains from its relationship with the stakeholder.	
	Requires	A set of products and/or services that the stakeholder consumes.	
	Provides	A set of products and/or services that the stakeholder offers to the organization.	
	Frequency	To point out whether this stakeholder contacts with the organization frequently, occasionally or in an established period of time.	
Resources	Resources and applications available by the system.		- Is there any application or resource available by the system? - Is this resource physical or virtual?
Events	External events that produce a system response.		Which events can produce an effect on the system? How the system capture these events and how response to them?
Offers	A set of products or services offered by the organization to its clients.		Is there any product or service that the system should provide to an external or internal stakeholder?
Demands	A set of products or services demanded by the organization to its clients.		Are there any requirements that the system cannot provide itself? Is it important who provide this service or product?

(continued)

Table 3 (continued)

Property	Description	Guideline
Restrictions	An overview about which types of restrictions the system should imposed on its entities.	- Behavioral restrictions: Is there any system requirement that specify limits on the behavior of the members of the system? - Critical restrictions: Is there any action whose inadequate usage could be dangerous for the system? - Usage restrictions: Is there any restriction on the usage of the system resources? Is there any restriction on the usage of the services and products offered by the system? Is there any restriction on who is an appropriate stakeholder to provide a service or product to the system? - Legal restrictions: Is there any normative document, such as governmental law or institutional internal regulations, that affects the system's entities behavior?

Table 4 Phase 1—Case study: system description document

Case study: ***System description document***	
System identifier	CMS (Conference Management System)
System description	This system should support the management of scientific conferences. This system involves several aspects from the main organization issues to paper submission and peer review, which are typically performed by a number of people distributed all over the world.
Domain	Research
Kind of environment	Virtual and distributed environment with established policies and norms that should be followed.
Global objectives	- Management of user registration - Management of conference registration - Management of the submission process - Management of the review process - Management of the publication process
Stakeholders	There is no external entity that interact with the system. Every entity that wants to interact with the system should be registered and logged in the system.
Resources	Database: it should include personal information and affiliation and information about which users are registered as authors, reviewers or publishers for each conference. Also it should include information about each conference, i.e., its status, its submitted papers and reviews,. . .
Events	Non external events are handled by the system.
Offers	- NewUsers_registration(); - Log_in();
Demands	
Restrictions	- The system should follow the legal documentation about the storage of personal data. - Each conference should describe its internal normative.

development of the analyzed system. Table 7 shows the criteria used to evaluate whether ROMAS is suitable.

ROMAS is focused on the development of regulated multiagent systems based on contracts. ROMAS is appropriate for the development of distributed systems, with autonomous entities, with a social structure, with the need of interoperability standards, regulation, and trustworthiness between entities and organizations. ROMAS is not suitable for the development of centralized systems or non-multiagent systems. Although non-normative systems could be analyzed using ROMAS, it is not recommended.

The analysis of the CMS case study features following this guideline shows that ROMAS is suitable for the development of this system. It is a distributed system, composed of intelligent systems with social relationships between them. The entities of the system should behave following the regulations of the system.

Table 5 Phase 1: Objectives description

Property	Description	Guideline
Identifier	Objective name identifier.	It is recommended to select a short name or an abbreviation.
Description	Informal description of the objective that is pursued.	There is no length limitation on this text. It should clearly describe this objective.
Activation Condition	First order formula that indicates under which circumstances this objective begins being pursued.	- Does the organization pursue this objective from the initialization of the system? - Is there any situation that activates this objective? Common circumstances that can activate objectives are: when an event is captured, the failure of other objective, the violation of a restriction, when an agent plays a specific role, and so on. - If this objective is deactivated, is there any situation that forces the objective to be pursued again?
Deactivation Condition	First order formula that indicates under which circumstances this objective stops being pursued.	- Is this objective pursued during the whole lifecycle of the system? - Is there any situation that deactivates this objective? Common circumstances that deactivate an objective are: when it is satisfied, when other objective is satisfied, when some restriction has been violated, and so on.
Satisfaction Condition	First order formula that indicates in which situation this objective is satisfied.	- Is the satisfaction of this objective measurable? - What results should be produced to claim that this objective has been satisfied?
Fail Condition	First order formula that indicates in which situation this objective has failed.	- Is there any situation that is contrary to this objective and that will invalidate it? - Is there any threshold that should not be exceeded?
Type	Objectives can be abstract or operational. An *abstract objective* is a non-functional requirement that could be defined to describe the global behavior of the organization. An *operational objective* is a specific goal that agents or roles have to fulfill.	If there is a task that can be executed in order to satisfy this objective, it is an operational objective, in other cases it is an abstract objective. Abstract objectives can be refined into other abstracts or operational objectives.
Decomposition	First order formula that specified how this objective is decomposed.	If this is an abstract objective it should be decomposed in several operational objectives which indicates which tasks should be executed in order to achieve this objective. Operational objectives can also be decomposed in order to obtain different subobjectives that can be pursued by different members of the organization. This fact simplifies the programming task and facilitates the distribution of responsibilities.
Related Action/ Restriction	Objectives can be related to a restriction on the behavior of the system, or to an action that must be executed in order to achieve this objective. Actions can be tasks, services or protocols.	The difference between a task and a protocol is that a task can be executed by one single agent, however a protocol is a set of tasks and interactions between two or more agents. Services are pieces of functionality that an entity of the system offers to the others, so the main difference between services and tasks or protocols is that they are executed when an entity request this functionality. At this phase it is not necessary to detail all the parameters of the task. You should describe in a high abstraction level what actions and activities are necessary to achieve this objective.

Type	Task, service or protocol.
Identifier	An identifier for the task, service or protocol.
Description	Informal description of the action.
Resources	Which applications or products are necessary to execute this task (for example, access to a database). This feature can be known at this analysis phase due to requirement specifications.

(continued)

Table 5 (continued)

Property	Description		Guideline
		However, if there is no specification, the specific implementation of each task should be defined in following steps of the methodology.	
	Activation condition	First order formula that indicates under which circumstances this action will be activate.	
	Inputs	Information that must be supplied.	
	Precondi-tion	A set of the input conditions and environment values that must occur before executing the action in order to perform a correct execution.	
	Outputs	Information returned by this action and tangible results obtained.	
	Postcondi-tions	Final states of the parameters of the environment, by means of the different kinds of outputs.	

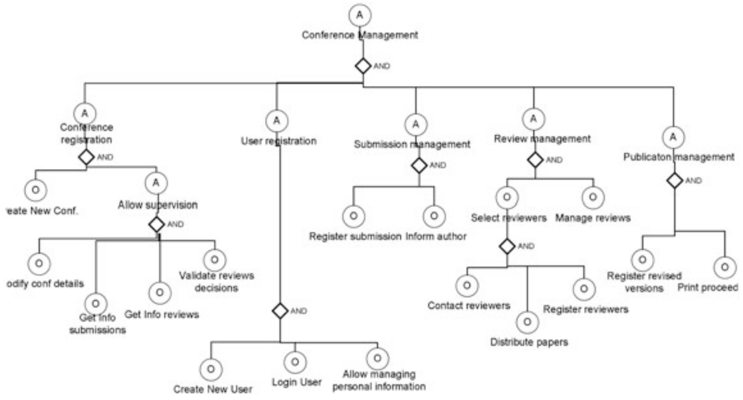


Fig. 9 Case study: objective decomposition diagram

The rights and duties that an entity acquires when it participates in the system should be formalized. For example, reviewers should know, before they acquire the commitment of reviewing a paper, when its revision must be provided. Therefore, a contract-based approach is recommendable.

Table 6 Phase 1—Case study: objective description document

Case study: *Conference registration objective* decomposition						
Identifier	Create new conference	Allow supervision	Modify conf details	Get Info submissions	Get Info reviews	Validate reviews decision
Description	The system should allow the registration of new conferences. The entity who registers the conference should be the chair of it.	The authorized entities should be able to supervise the status of the conference and modify its details.	The description details such as deadlines, topics and general description can be modified by the chair or vice-chair of the conference.	The system should provide information about the submitted papers.	The system should provide information about the reviews that has been uploaded in the system.	The chair should validate the decisions about acceptance or rejection of papers performed by the reviewers.
Activation Condition	True (always activated)	Conference_status= activated	Conference_status= activated	Conference_status= activated	Conference_status= activated	Conference_status= revision
Deactivation Condition	False	Conference_status= canceled	Conference_status= canceled	Conference_status= canceled	Conference_status= canceled	Conference_status= canceled
Satisfaction Condition						
Fail Condition						
Type	Operational	Abstract	Operational	Operational	Operational	Operational
Decomposition		Modify conf details AND Get info submissions AND Get info reviews AND Validate reviews decision				
Related Action/Restriction						
Identifier	Create_new_conference()		Modify_conf_details()	Get_Info_ submissions()	Get_Info_ reviews()	Validate_reviews_ decision()
Type	Service		Service	Service	Service	Task/Protocol

(continued)

Table 6 (continued)

Case study: ***Conference registration objective*** **decomposition**					
Description	The registration must be performed by means of a graphical online application.	After checking that the user that is trying to modify the conference details is authorized to do that, the system will provide a graphical online application to update the details. The information is shown by means of a graphical online application.	Only pc members can access to the information about submissions. The information is shown by means of a graphical online application	Only pc members that are not authors of the paper can access to the reviews of a specific paper. The information is shown by means of a graphical online application	The chair should validate one per one the decision for each paper. If the chair performs the action by itself this objective would be pursued by means of a task. If the final decision is negotiated between the PC members this objective should be pursued by means of a protocol.
Resources	Access to the conferences database	Access to the conferences database	Access to submitted papers database	Access to reviews database	Access to reviews database.
Activation condition	Registered user demand				After the review deadline is finished
Inputs	Deadlines, topics of interests and general information				
Precondi-tion	The entity that executes the task should be a registered user.				
Outputs					
Postcondi-tions	The user that executes the task becomes the chair of the conference.				After the validation the decision is considered final and authors should be notified.

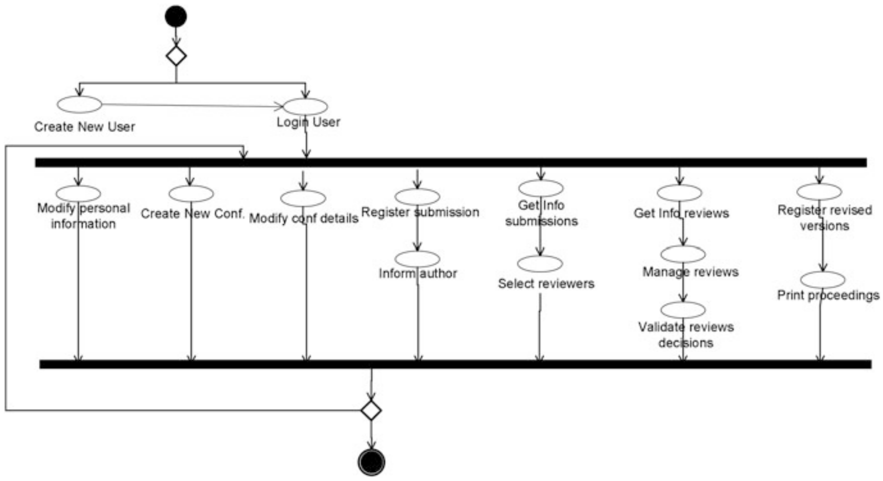


Fig. 10 Case study: use case

2.2 PHASE 2: Organization Description

During this phase, the analysis of the structure of the organization is carried out. In the previous phase of the methodology, the operational objectives are associated to specific actions or restrictions. In this phase, these actions and restrictions are analyzed in order to identify the roles of the system. A *role* represents part of the functionality of the system, and the relationships between roles specify the structure of the system.

2.2.1 Process Roles

The roles involved in this phase are the same as in the previous phase: the *system analyst* and the *domain expert*. The domain expert is in charge of supporting the system analyst facilitating information about domain requirements and restrictions.

2.2.2 Activity Details

The flow of tasks of this phase is reported in Fig. 11, and each task is detailed in Table 8.

2.2.3 Work Products

This phase uses the work products produced in the previous phase (*System definition*), and it produces the following work products: one *role description document* for each role; one *internal view diagram* for each role; and one *organizational view diagram*. Figure 12 shows graphically the products used and produced by each task.

Some of the work products generated are instances of the ROMAS metamodel. Figure 13 describes the relation between these work products and the metamodel elements in terms of which elements are *defined* (D), *refined* (F), *quoted* (Q),

Table 7 ROMAS suitability guideline

DISTRIBUTION: It is recommendable to use a distributed approach to develop the system if any of these questions is affirmative.
- Composed system: Is the system to be developed formed by different entities that interact between them to achieve global objectives? Are there different institutions involved in the system? - Subsystems: Is the system composed by existing subsystems that must be integrated? - Distributed data: Is the required data spread widely in different locations and databases? Are there any resources that the system uses distributed in different locations?
INTELLIGENT ENTITIES: It is recommendable to use an agent approach to develop the system if any of these questions is affirmative.
- Personal objectives: Do the entities involved in the system have individual and potentially different objectives? - Heterogenous: Is possible that entities of the same type had been implemented with different individual objectives and implementations? - Proactivity: Are the entities of the system able to react to events and also able to act motivated only by their own objectives? - Adaptability: Should be the system able to handle dynamic changes in its requirements and conditions?
SOCIAL STRUCTURE: It is recommendable to use an organizational approach to develop the system if any of these questions is affirmative.
- Systems of systems: Does the system needs the interaction of existing institutions between which exist a social relationship in the real-world that must be taken into account? - Social relationships: Do the entities of the system have social relationships, such as hierarchy or submission, between them? - Departments: Is the functionality of the system distributed in departments with their own objectives but that interact between them to achieve common objectives? - Regulations: Are there different regulations for different parts of the system, i.e., is there any regulation that should be applied to a group of different entities but not to the rest of them? - Domain-like concepts: Is the domain of the system in the real-world structured by means of independent organizations?
INTEROPERABILITY: The system must implement interoperable mechanism to communicate entities if any of these questions answers is affirmative.
- Technical Interoperability: Is possible that different entities of the system use different (potentially incompatible) technologies? - Process Interoperability: Is possible that different entities of the system employ divergent (potentially incompatible) processes to achieve their goals? - Semantic Interoperability: Is possible that different entities of the system utilize different vocabularies and coding schemes, making it difficult to understand the data of others?
REGULATIONS: If the system has regulations associated it is recommended to apply a normative approach to develop the system. Only in the unlikely possibility that the norms of the system were static (no possibility of changing over time) and all the entities of the system are implemented by a trustworthy institution taking into account the restrictions of the system a non normative approach could be used.
- Normative documents: Is the system or part of it under any law or institutional regulation? - Resources restrictions: Are there specific regulations about who or how system resources can be accessed? - Dynamic regulations: Should the system be adapted to changes in the regulations? - Openness: Is the system open to external entities that interact and participate in the system and these entities should follow the regulations of the system? - Risky activities: Is there any action that if it is performed the stability of the system would be in danger?
TRUSTWORTHINESS: It is recommended to use a contract-based approach if any of these questions is affirmative.
- Formal interactions: Are there entities that depend on the behavior of the others to achieve their objectives and whose interactions terms should be formalized? - Contractual commitments: Should the entities of the system be able to negotiate terms of the interchanges of products and services and formalize the results of these negotiations? - Social commitments: Are the entities of the system able to negotiate their rights and duties when they acquire a specific role? Could the social relationships between agents be negotiated between them? - Control system: Is the system responsible of controlling the effective interchange of products between entities? - Openness: Is the system open to external entities that interact and participate in the system acquiring a set of rights and duties?

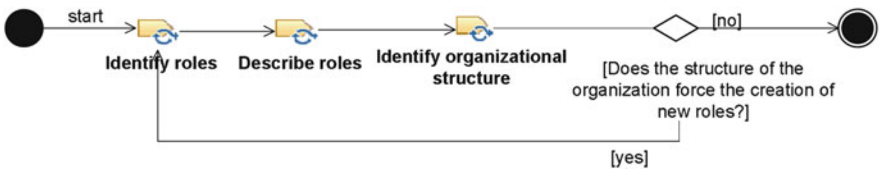

start
Identify roles
Describe roles
Identify organizational structure
[no]
[Does the structure of the organization force the creation of new roles?]
[yes]

Fig. 11 Phase 2: activity tasks

Table 8 Phase 2: activity tasks

ID.task	Task	Description	Roles involved
2.1	Identify roles	Following the guideline *Role identification guideline* the roles of the system are identified and associated to different parts of the system functionality.	System analyst and domain expert
2.2	Describe roles	Following the guideline *Role description document* each identified role is analyzed. The details about each role are graphically represented by means of instances of the *internal view diagram*.	System analyst and domain expert
2.3	Identify organizational structure	Identify how the members of the organization interact between them, i.e., which social structure has the organization and graphically represent that using an *organizational view diagram*	System analyst and domain expert

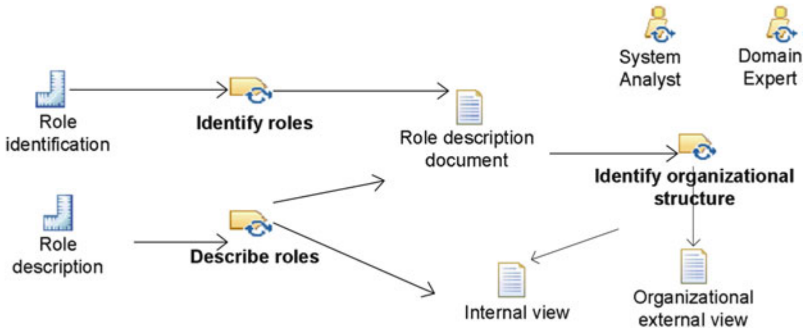


Fig. 12 Phase 2: resources and products used

related (R), or *relationship-quoted* (RQ). These terms are described in the specification of the FIPA template [2].

Role Identification Guideline

A role is an entity representing a set of goals and obligations, defining the services that an agent or an organization could provide and consume. The set of roles represents the functionality of the system; therefore, the roles that a system should have are defined by the objectives of the system and should also take into account previous system requirements. The relationships and interactions between

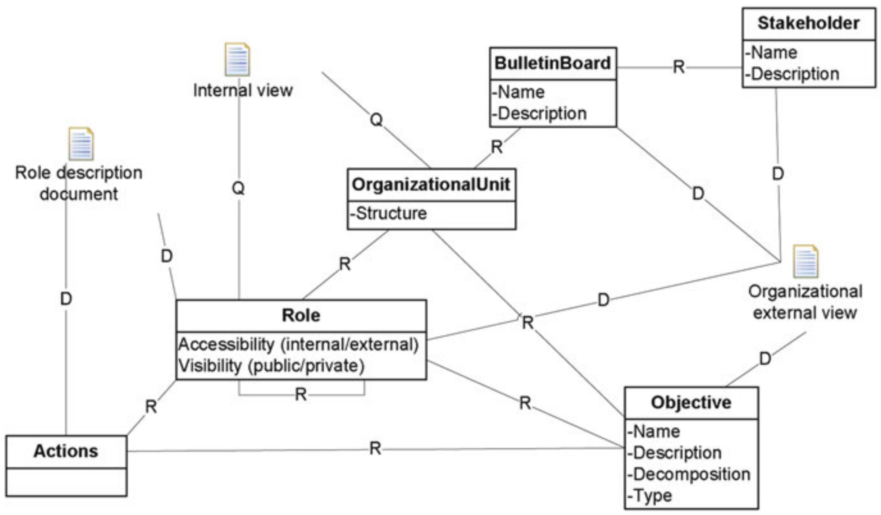


Fig. 13 Phase 2: relations between work products and metamodel elements

roles are the basis to define the structure of the organization. This guideline is designed to help the system analyst to identify the roles that are necessary in the system. Figure 14 represents de sequence of activities to do.

The first step of the process consists in asking the *domain expert* and checking in the *system description* document whether there is any pre-established role defined in the requirements of the system.

After that, every operational objective described in the *Objective description document* should be analyzed. It is recommended to analyze all the operational objectives obtained by the decomposition of an abstract objective before analyzing the next abstract objective.

If this operational objective is associated to a restriction, it would add a norm in the organization that pursues this objective. Besides, if this restriction is associated to an external event or a threshold, there must be an entity responsible for handling this event or measuring this threshold variable.

If this operational objective is associated to a protocol, the system analyst should revise the sequence of actions necessary to perform this protocol in order to obtain all the entities that participate in this protocol.

Usually, each task and functionality are associated to a role in order to create a flexible and distributed system. However, decomposing the system into too many entities can increase the number of messages between entities, the number of restrictions, and the complexity of each activity. Although the system analyst is responsible for finding the balance taking into account the specific features of the domain, here we present some general guidelines:

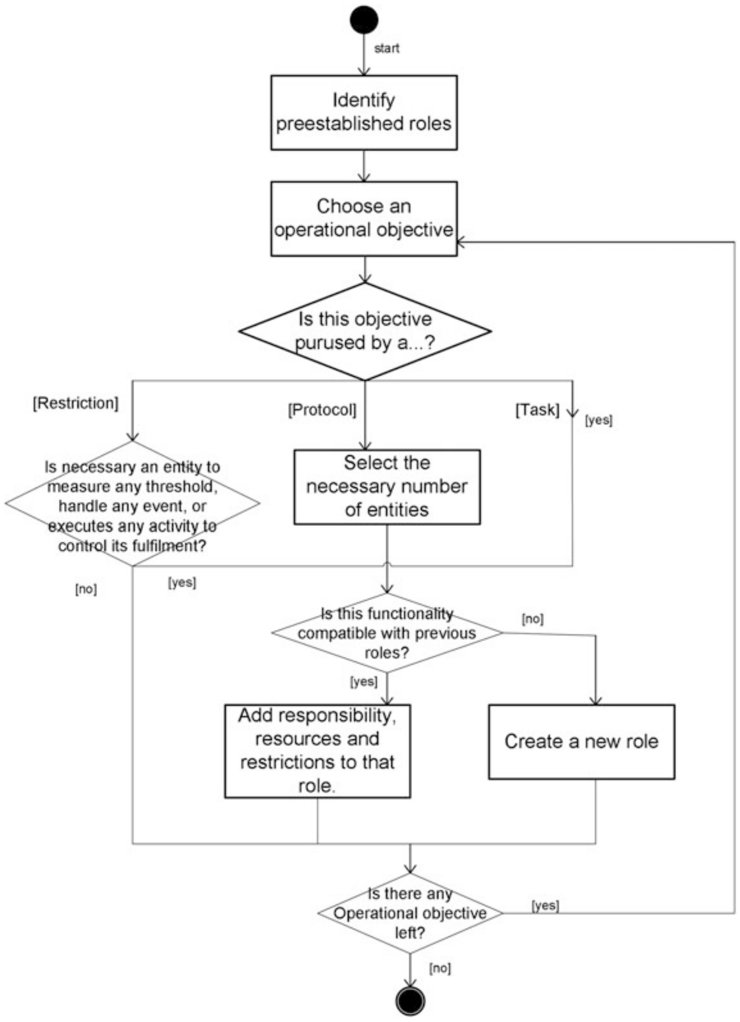

start
Identify preestablished roles
Choose an operational objective
Is this objective purused by a...?
[Restriction]
[Protocol]
[Task]
[yes]
Is necessary an entity to measure any threshold, handle any event, or executes any activity to control its fulfilment?
Select the necessary number of entities
[no]
[yes]
Is this functionality compatible with previous roles?
[no]
[yes]
Add responsibility, resources and restrictions to that role.
Create a new role
Is there any Operational objective left?
[yes]
[no]

Fig. 14 Phase 2: role identification guideline

It is not recommended to assign two functionalities to the same role when

- These functionalities have different physical restrictions, that is, they must be performed in different places.
- These functionalities have temporary incompatible restrictions, that is, when they cannot be executed at the same time by the same agent. For example, it is usual that an entity was able to buy and sell, but as far as he is not able to sell and buy the same item at the same time, it is recommended to create one role Buyer and one role Seller. Remember that roles represent functionalities, so any final entity of the system could be able to play several roles at the same time.

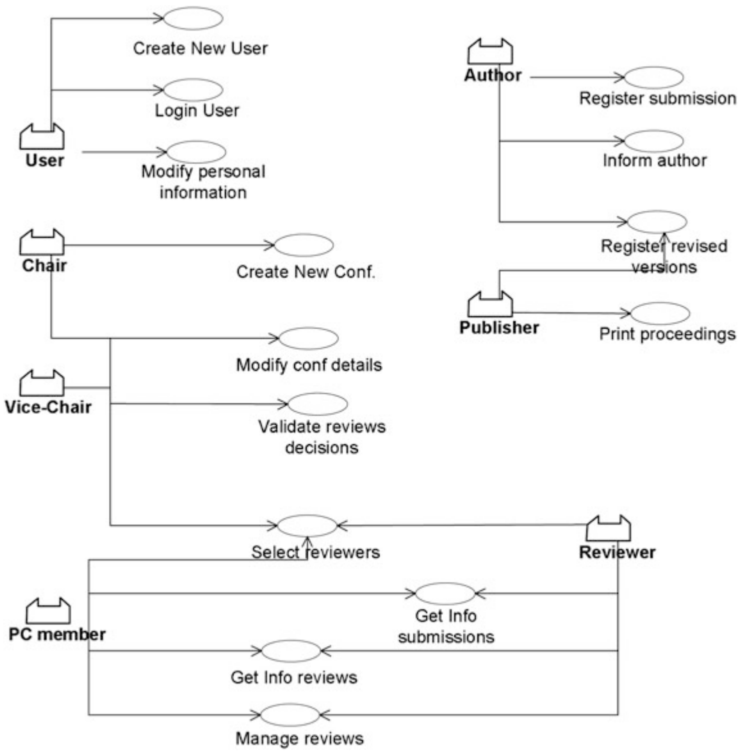


Fig. 15 Case study: roles overview

- These functionalities involve the management of resources that are incompatible. For example, the functionality of validating who is able to access a database should not be joint to the functionality of accessing the database. The reason is that if the entity who is accessing the database is responsible for validating its own access, the security can be compromised.

  It is recommended to assign two functionalities to the same role when
- These functionalities cannot be executed concurrently and they are part of a sequence.
- These functionalities access the same resources and have the same requirements.
- These functionalities can be considered together as a general functionality.

In order to provide a fast and general overview, it is recommended to create a graphical representation of the relationships between the identified roles and the tasks and protocols. A relationship between a role and an action in this diagram means that the role is responsible for this action, participates in it, or is affected by its results. Figure 15 gives an overview of the results obtained when applying this guideline to the CMS case study. As is shown, seven roles have been identified:

Table 9 Phase 2: role description document

Property	Description	
Identifier	General name of role. It is recommended to select a short name or an abbreviation.	
Description	Informal description of the role. There is no length limitation on this text.	
List of objectives in which the role participates	**Objective's identifier**	The identifier of the objective that this role is going to contribute to its satisfaction
	Description of its contribution	An informal text describing how this role contributes to the satisfaction of this objective.
	Task/Protocol/ Service	Which task is responsibility of this role or in which protocol this role participates. If this task should be activated as a reaction of a petition of other entity, this task should be published as a service.
	Responsibility shared with	A text explaining if this role pursue this objective alone or if he collaborates with others to achieve it.
Resources: Used	A list of the resources (products, services and applications) that this role requires to develop its functionality. This text should specify which type of access the role needs (reading, executing, writing, partial or full access)	
Resources: Provided	A list of the resources (products, services and applications) that this role provides.	
Events	A list of the events that this role handles and with which task.	
Restrictions	A list of the restrictions that are inherent to the functionality that this role executes.These restrictions are mainly derived from the information provided by the Domain Expert.	
Other memberships	A text explaining if this role in order to executes a task inside the organization it must be part of other different organization. If it is know, its rights and duties in the other organization must be detailed in order to ensure the coherence its objectives, rights and duties. However, due to privacy restrictions it is probable that many details cannot be shared between organizations.	
Personal objectives	A role can pursue an objective not directly related to the functionality required by the organization. For example, it can pursue an objective in order to maintain its integrity.	
Who plays the role	Is this role played by a single entity or by an organization? If it is played by an organization this organization must be analyzed following each step of the methodology.	

- The *User* role is an entity of the system that must be registered in order to access the system. On the contrary of the rest of the roles, this role is not related to any specific conference.
- The *Author* role is an entity attached to a specific conference in which this role can submit papers and receive information about the status of its papers.
- The *Chair* role is the main role responsible for a conference. This role is able to create a conference and share the responsibility of selecting the reviewers, validate the revisions, and update the conference details with the *Vice-Chair* role.
- The *PC member* role is responsible for managing the reviews, can participate in the selection of the reviewers, and have access to the information about submissions and reviews for a specific conference.
- The *Reviewer* role is responsible for submitting the reviews to the system.
- The *Publisher* role is responsible for managing the revised versions of the papers and printing the proceedings.

Role Description Document and Internal View Diagram

Each role should be described by means of the guideline offered in Table 9. This guideline allows the analysis of the roles and also the analysis of the relationships between them. After this analysis, this information is graphically represented by means of an *internal view diagram* for each role.

Table 10 Phase 2: Case study—reviewer role description document

Property	Description		
Identifier	Reviewer		
Description	This role is responsible from submit the reviews to the system. It is attached to an specific conference. It is responsible from submit a review from a paper within the established deadline and in the specific format that the conference specifies.		
List of objectives in which the role participates	**Objective's identifier**	Select reviewers	Manage reviews
	Description of its contribution	Reviewers should negotiate with *PC members* which papers are they going to review, when they are supposed to provide the reviews and which specific format these reviews must have.	Reviewers should send to the system their reviews. PC members would validate the reviews and contact the reviewers if there is any doubt in the information supplied
	Task/Protocol/ Service	Reviewers participate in the protocol Select_reviewers()	Reviewers are responsible from the protocol *Manage_reviews()* and they offer the service *Execute_review()*
	Responsibility shared with	PC members	
Resources: Used	- Reviews and papers database. -They use the service *Get_info_submissions()*		
Resources: Provided			
Events	Conference details modification event		
Restrictions	- The same entity cannot be the author and the reviewer of the same paper. - Reviewers only have access to the information about their own reviews. They do not have access to other reviews or to the authors's personal details.		
Other memberships	Any entity that wants to play the role reviewer should be previously registered in the system as a *user*.		
Personal objectives	In general, there is no personal objectives for reviewers in the system. However, some conferences can encourage the productivity of their reviewers by offering rewards for each revised paper or for presenting the reviews before a specific date.		
Who plays the role	This role is played by a single entity.		

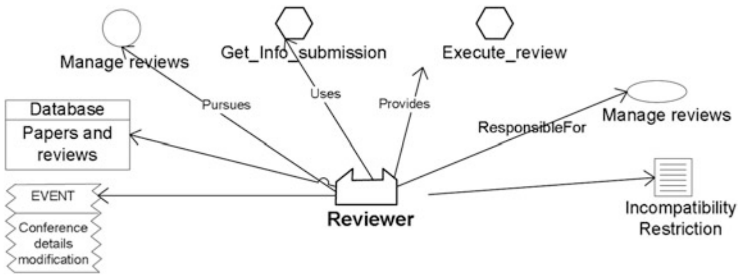


Fig. 16 Case study: reviewer role diagram

As an example, Table 10 shows the description of the role *reviewer* from the case study. Figure 16 shows its graphical representation using a ROMAS internal view diagram.

Organizational View Diagram

One organizational view diagram is created to graphically represent the structure of the system. Besides, this diagram also describes the overview of the system by

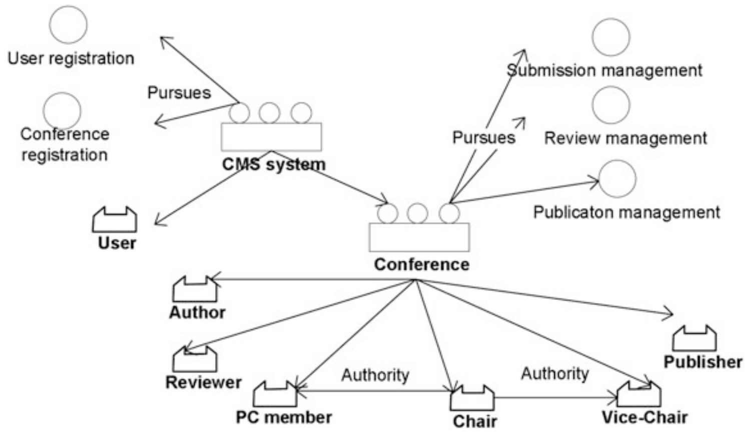


Fig. 17 Case study: organizational diagram

means of its global *objectives* and how the system interacts with the environment of the system (which *services* offers and demands to/from the *stakeholders* and which *events* the system is able to handle). The necessary information to fulfill these diagrams is obtained from the *System description document*. Due to the fact that in the literature, there are several well-defined guidelines to identify the organizational structure of a system, ROMAS does not offer any new guideline. Instead, the use of the guideline defined by the GORMAS methodology in [1] is recommended.

Figure 17 shows the organizational view diagram of the CMS case study. Inside the main system, the *Conference organization* represents each conference that is managed by the system. Each conference is represented as an organization because using this abstraction, each one can define its own internal legislation and can refine the functionality assigned to each entity of the system.

2.3 PHASE 3: Normative Context Description

During this phase, the normative context of the system is analyzed by means of identifying and formalizing the norms and the social contracts that regulate the entities' behavior inside the system. Norms are formalized using the following syntax: *(normID,Deontic,Target,Activation,Expiration,Action,Sanction,Reward)*.

2.3.1 Process Roles

The system analyst is responsible for performing the activities of this phase. The domain expert will support the system analyst during the identification of the norms that regulate the system.

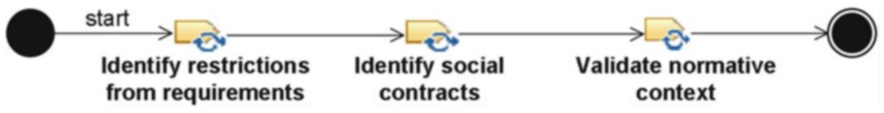

start
Identify restrictions from requirements
Identify social contracts
Validate normative context

Fig. 18 Phase 3: activity tasks

Table 11 Phase 3: activity tasks

ID.task	Task	Description	Roles involved
3.1	Identify restrictions from requirements	Following the guideline *Organizational norms*, the system analyst formalizes the norms described in the requirements that regulate the agent behavior. This norms refine the *organizational view* diagram of the organization associated to these norms.	System analyst and domain expert
3.2	Identify social contracts	Following the guideline *Social contracts*, the social contracts of the system are identified and formalized by means of the *contract template view diagram*.	System analyst and domain expert
3.3	Validate normative context	Following the guideline *Normative context validation*, the coherence among system's norms and between them and the social contracts of the system is validated.	System analyst

2.3.2 Activity Details

The flow of tasks inside this activity is reported in Fig. 18, and the tasks are detailed in Table 11. Figure 19 describes the relation between these work products and the metamodel elements in terms of which elements are *defined* (D), *refined* (F), *quoted* (Q), *related* (R), or *relationship-quoted* (RQ).

2.3.3 Work Products

Figure 20 shows graphically the products used and produced by each task. The remainder of this section details the following work products: (1) *Organizational norms guideline*, (2) *Social contracts guideline*, (3) *Normative context validation guideline*, (4) *Contract template view diagram*. The *organizational view* diagram of the organization associated to the identified norms is refined by means of adding these norms to the diagram, in the same way that the *internal view* diagram of each role is refined by adding the social contracts and norms attached to this role.

Organizational Norms Guideline

This guideline specifies a process to identify and formalize restrictions on the behavior of entities gained from the analysis of system requirements. These normative restrictions are associated with specific features of the system and are usually well known by domain experts but not formally expressed in any document. This guideline helps the system analyst identify these restrictions with the support of the domain expert.

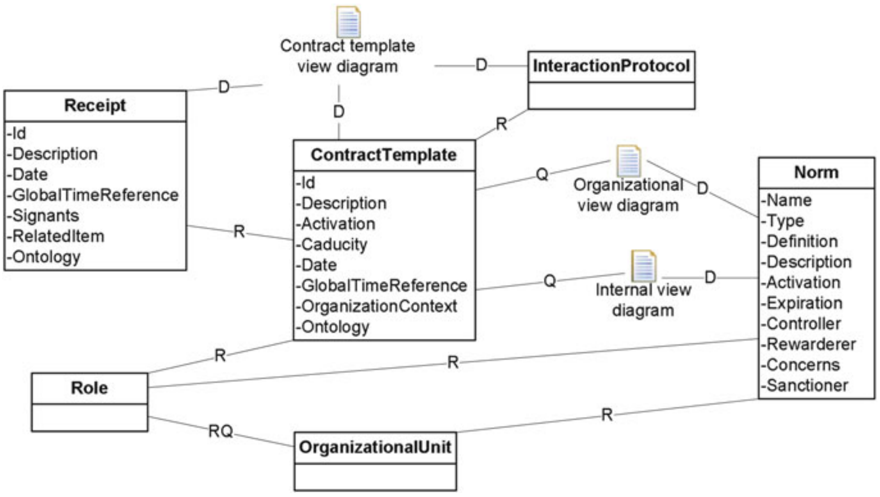


Fig. 19 Phase 3: relations between work products and metamodel elements

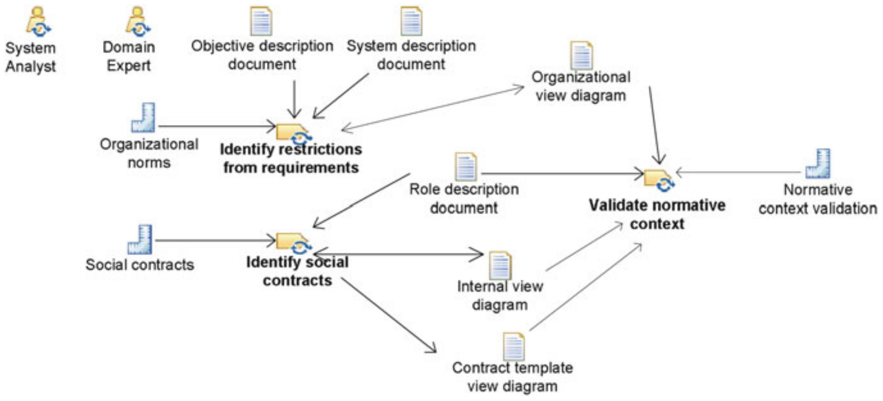

System Analyst
Domain Expert
Objective description document
System description document
Organizational view diagram
Organizational norms
Identify restrictions from requirements
Role description document
Validate normative context
Normative context validation
Social contracts
Identify social contracts
Internal view diagram
Contract template view diagram

Fig. 20 Phase 3: resources and products used

Step 1 *Analysis of system description documents* These documents contains in plain text the requirements of the system, and if the system is composed of several organizations, there will be one system description document for each organization. The norms that arise from a document only affect the entities inside the organization that this document describes. The following steps are executed:

1.1 *Analysis of the resources* For each resource of the system, such as a database or an application, it is analyzed who has access to the resource, who cannot access it, and who is responsible for its maintenance. Therefore, permission, prohibition, and obligation norms are associated with these resources. For example, the analysis of the *Conference* database highlights the norm that only the *chair* of the conference can modify the details of the conference (NModifyDetails, FORBIDDEN, !Chair, Modify(ConferenceDB),-,-,-,-).

1.2 *Analysis of the events:* For each event that the organization must handle, an obligation norm to detect this event is created. If the organization should react to this event by executing a task, another obligation norm is specified. The activation condition of this norm is the event itself.

1.3 *Analysis of the offers/demands:* External stakeholders can interact with the organization, offering and demanding services or resources. If the system is obliged to offer any specific service, an obligation norm is created. If there are specific entities that are allowed to offer, demand, or use a service, a related permission norm is created. On the other hand, if there are specific entities that are not allowed to offer, demand, or use it, a related prohibition norm is created.

1.4 *Analysis of domain restrictions* The last attribute of the system description document analyzes if there are normative documents attached to the organization or if there are specific domain restrictions that must be taken into account. In both cases, the restrictions are specified in plain text, so the system analyst should analyze and formalize these restrictions using a formal syntax. For example, in the CMS case study, the domain expert has claimed that *"Each conference should describe its internal normative"*. This restriction is formalized as (confNormative, OBLIGED, Conference, Define(Normative),-,-,-,-). This norm will be attached to every conference; therefore, the task of defining the internal normative should be added to a role inside the conference. In this case, this task has been added to the *chair* responsibilities.

Step 2 *Analysis of the objectives description document* We can differentiate two types of objectives: the objectives attached to restrictions and the objectives attached to specific actions. First, for each objective that pursues the stability of any variable of the system in a threshold, a forbidden norm should be created to ensure that the threshold is not exceeded. A variable of a system is anything that the system is interested in measuring; for example, the temperature of a room or the quantity of money that a seller earns. Second, for each objective that is attached to an action, an obligation norm is created in order to ensure that there is an entity inside the system that pursues this objective. The activation and expiration conditions of the created norms are determined by the activation and expiration conditions of the related objective.

Social Contracts Guideline

This guideline specifies a process to identify and formalize social contracts inside a specific organization regarding the information detailed in the *role description document*, the roles' *internal view diagrams*, and the structure of the organization. Social contracts are used to formalize two kinds of social relationships: (1) *play role contract template*, which specifies the relationship between an agent playing a role and its host organization; and (2) *social relationship contract template*, which specifies the relationship between two agents playing specific roles. Social order thus emerges from the negotiation of contracts over the rights and duties of participants.

One *play role contract template* should be defined for each role of the organization in order to establish the rights and duties that any agent playing this role should fulfill. Therefore, in the CMS case study, seven *play role contract templates* should be formalized: one for role *user* of the main organization and six for each role described inside the *Conference* organization (*author*, *reviewer*, *PC member*, *Chair*, *Vice-chair*, *Publisher*). That means that the rights and duties of an agent that tries to play a role inside a conference can be different, depending on how each conference negotiates these contracts. For example, one conference can establish that a PC member cannot submit a paper to this conference while other conferences do not add any restriction like that. Since every agent that intends to play a specific role inside the system must sign a *play role contract*, every agent will be aware of its rights and duties inside the organization in advance.

One *social relationship contract template* should be defined for each pair of roles that must interchange services and products as part of the social structure of the organization. Contracts of this kind should be negotiated and signed by the related entities and not by the organization as a whole. However, if the terms of the contract are not negotiated by the entities and the relationship between these agents is determined by their organization, it is not necessary to create a social relationship contract. Instead, the rights and duties of each role over the other are included in their respective *play role contracts*. In the CMS case study, there is an authority relationship between the *chair* role and the *vice-chair* role. The terms of this relationship are specified by each conference. Therefore, the rights and duties from one entity to the other are formalized in their respective *play role contracts*, and no social relationship contract is created.

Below, each step of the guideline that should be applied to each role of the system is described.

- *Adding identified norms:* Every restriction or norm identified during the application of the *Organizational norms guideline* that affects the role should be added to the contract. The norms that are attached to several roles but that include this specific role should be added. This can increase the size of the contract, so it is the responsibility of the domain expert to specify which norms should be communicated. For example, in the case of CMS case study, not all governmental norms related to the storage of personal data are included in the contracts; only a norm that specifies that any agent inside the system should follow this regulation is specified in the contracts.
- *Analysis of the organizational objectives:* In previous phases of the ROMAS methodology, the requirements of the system are analyzed by means of the analysis and decomposition of the objectives of the system. Each objective is associated with an action that must be performed in order to achieve it, and these actions are associated with specific roles that become responsible for executing them. Therefore, for each objective related to a role, an obligation norm must be created that ensures the execution of this action. If the action related to the objective is a *task*, the role is obliged to execute this task. If it is related to a *service*, the role is obliged to offer and register this service. The activation and expiration of the norm match the activation and expiration of the objective.
- *Analysis of offers/demands:* The description of each role should specify which resources and services this role must offer and which ones it can use. For each

resource and service that this role is able to use, a permission norm is added. For each resource or service that this role cannot have access to, a prohibition norm is created. Also, for each resource and service that this role is supposed to provide, an obligation norm is added. In this sense, an agent would not be able to play a role unless it were able to provide all the services and resources that are specified in the *play role contract*.
- *Analysis of the events:* For each event that the role must handle, a norm that forces any agent that plays this role to detect this event is created. If the role should react to that event by executing a task, an obligation norm is created whose activation condition is that event and indicates that the role should execute this action.
- *Analysis of the relationships:* As is discussed earlier, the norms derived from the social relationships between roles should be included in the *play role contract template* when they cannot be negotiated by the entities playing these roles, that is, they are rigidly specified by the organization. In other cases, a *social relationship contract* should be created and these norms included in it. The norms that are derived from the social relationship should be activated only when the social relationship is active and their deontic attribute depends on the type of relationship between the parties. If two roles are incompatible, a prohibition norm is added specifying this fact. In the same way, if any agent playing one role is required to play another, an obligation norm is included in the contract. Usually, a social collaboration appears when several roles should interact to achieve a global goal of the organization. In such cases, a set of obligation norms specifies which actions and services are the responsibility of each entity. If the collaboration relationship indicates *information*, it means that one role is obliged to inform another when some conditions occur. An *authority/submission* relationship requires the specification of: (1) which services should provide the submitted party, (2) which actions the authority can force the other agent to do, and (3) which actions the submitted party cannot perform without the consent of the authority.
- *Analysis of personal objectives:* A personal objective of a role is a goal that is not directly related to the main goals of the system, but that all the agents that play this role will pursue. The system as an entity can establish some restrictions on the performance of personal objectives. An example of a personal objective in the *CMS* case study is that although the agents that play the role *author* pursue the objective of *Submitting as many papers as possible*, each conference can establish limits on the quantity of papers that an author can submit to the same conference.

Normative Context Validation Guideline

The validation of the normative context is understood as the verification that there are no norms in conflict, that is, that the normative context is coherent. As is presented in [3], conflicts in norms can arise for four different reasons: (1) the obligation and prohibition to perform the same action; (2) the permission and prohibition to perform the same action; (3) obligations of contradictory actions; (4) permissions and obligations of contradictory actions. Therefore, after the specification of the organizational norms and the social contract templates that define the structure of the organization, it is necessary to verify that the normative context as a whole is coherent.

Each organization can define its own normative context independently of the other organizations that constitute the system. The first step is analyzing the normative context of the most simple organizations, that is, the organizations that are not composed of other organizations. After that, we will analyze the coherence between this simple organization and the organization in which it is. This process will continue until analyzing the coherence of the main organization of the system.

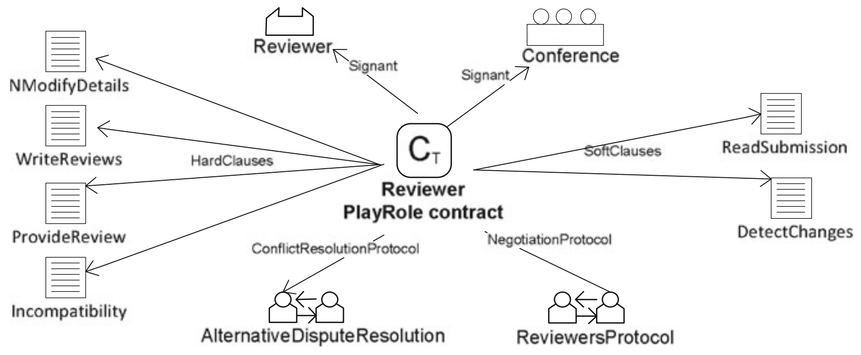
In order to analyze the coherence of a specific organization, it is necessary to verify that: (1) There is no state of the system in which two or more organizational norms in conflict are active. (2) There is no norm that avoids the satisfaction of an organizational objective, that is, there is no norm that is active in the same states as the objective is pursued and whose restriction precludes the satisfaction of this objective. (3) There is no social contract that specifies clauses that are in conflict with the organizational norms. (4) There is no pair of social contracts whose clauses are in conflict between them and, therefore, the execution of one contract would preclude the satisfactory execution of the other one. (5) There is no social contract in which a role participates whose clauses preclude the satisfaction of the roles objective.

The validation task can be performed manually or by means of automatic techniques such as model checking. In [4], we present a plug-in integrated in our case tool that allows a simple verification of the coherence between the organizational norms and the contracts by means of the SPIN model checker [5].

Contract Template View Diagram

One contract template diagram is created for each identified social contract. The recommended steps to specify a contract template are

– *Identify signants:* If it is a *play role contract template*, the signants are the entity that tries to pursue this role and the organization as a whole. If there is a specific role in charge of controlling the access to the organization, the entity playing this role will sign the contract on behalf of the organization. If it is a *social relationship contract template*, the signants are the entities playing the roles that have the relationship.
– *Attach clauses:* The norms that have been identified by means of the *social contract guideline* are included in the contract. If the norm to be included in the contract must be in any contract of this type, this norm is defined as a *hard clause*. On the contrary, if the norm to be included in the contract is a recommendation, this norm is defined as a *soft clause*.
– *Define receipts:* In order to monitor the correct execution of the contract, it is recommended to define specific artifacts that entities participating in the contract should provide in order to prove the fact that they have executed their required actions.
– *Define authorities:* Optionally, the designer can define who is responsible for verifying the coherence of the final contract (*notary*), for controlling the correct execution of the contract (*regulation authority*), and for acting in case of dispute between the signant parties (*Judge*).



NORM ID	NORM DESCRIPTION (Deontic,Target,Activation,Expiration,Action,Sanction,Reward)
NModifyDetails	(FORBIDDEN, !Chair, Modify(ConferenceDB),-,-,-,-)
WriteReviews	(PERMITTED, Reviewer, -,-, WriteAccess(Conference database),-,-)
ReadSubmission	(PERMITTED, Reviewer,-, - , UseService(Get Info Submission),-,-)
DetectChanges	(OBLIGED, Reviewer, -,-, Detect(ChangesConference event),-,-)
ProvideReview	(OBLIGED, Reviewer, -,-, ProvideService(Execute review),-,-)
Incompatibility	(FORBIDDEN, Reviewer, reviewer=author,-, UseService(Get Info Submission),-,-)

Fig. 21 Phase 3: Case study—reviewer play role contract template

– *Identify protocols:* Optionally, the designer can define specific negotiation, execution, and conflict resolution protocols. At this phase, only a general description of these protocols is provided. They will be completely specified in the next phase of the methodology.

Figure 21 shows the *play role contract template* that any entity that wants to play the role reviewer should sign. It is signed by the role that wants to play the role *reviewer* and by the *conference* in which the entity wants to participate. There are six clauses attached to this contract template that specify that an entity playing this role is not allowed to modify the details about a conference unless it is also the chair of this conference (*NModifyDetails* norm), neither to access the submission information about a paper in which he is also author (*Incompatibility* norm). This entity would have permission to access the reviews database (*WriteReviews* norm) and to use the service *Get Info Submission* (*ReadSubmission* norm). This entity would be obliged to detect when the conference details have changed (*DetectChanges* norm) and to provide the service *Execute review* (*ProvideReview* norm).

2.4 PHASE 4: Activity Description

During this phase, each identified task, service, and protocol is described by means of instances of the *activity model view*.

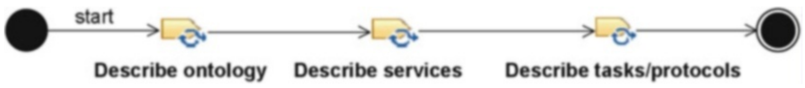


Fig. 22 Phase 4: activity tasks

Table 12 Phase 4: activity tasks

ID.task	Task	Description	Roles involved
4.1	Describe ontology	System domain concepts are analyzed. These concepts will be used to define the inputs, outputs and attributes of tasks, protocols and services.	System analyst and domain expert
4.2	Describe services	Define service profile attributes for each service. One *activity view* diagram is created for specifying each service implementation. If there are services that should be published to other members of the system or to external stakeholders, the *organizational view diagram* of the system should be refined by adding a *BulletinBoard*. This abstraction is an artifact where authorized entities can publish and search services.	System analyst
4.3	Describe tasks and protocols	Create one instance of the *activity view diagram* for each task and protocol to specify them. In addition to the protocols associated to objectives and roles, the contracts of the system should be completed by adding specific negotiation, execution and conflict resolution protocols.	System analyst

2.4.1 Process Roles

The domain expert should provide the domain ontology and should give support to the system analyst in the definition of the protocols, tasks, and services.

2.4.2 Activity Details

The flow of tasks inside this activity is reported in Fig. 22, and the tasks are detailed in Table 12. Figure 23 describes the relation between these work products and the metamodel elements in terms of which elements are *defined* (D), *refined* (F), *quoted* (Q), *related* (R), or *relationship-quoted* (RQ).

2.4.3 Work Products

Figure 24 shows graphically the products used and produced by each task. One *activity view diagram* is created for each task, protocol, or service identified in the previous phases of the methodology. Phase 2 shows the tasks, services, and protocols that each role should implement, and phase 3 identifies the negotiation, execution, and conflict resolution protocols for the contract templates.

An example is presented in Fig. 25. It shows the description of the *reviewer play role negotiation protocol*. First, the chair sends to the user that tries to play the role reviewer the details about the conference (deadlines, topics of interest, . . .). The user analyzes this information, and if necessary, proposes a change in the review

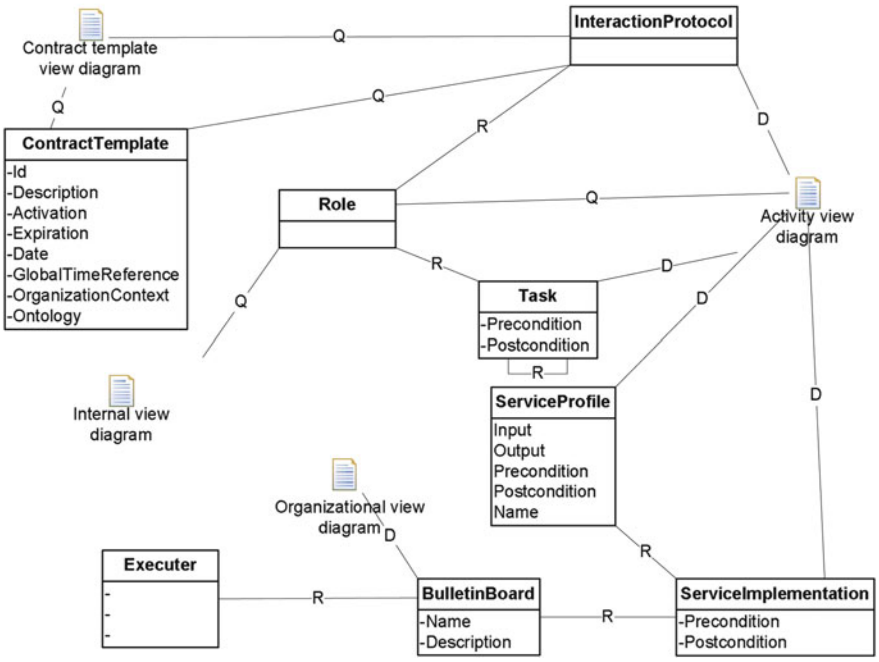


Fig. 23 Phase 4: relations between work products and metamodel elements

deadlines. This change can be accepted or rejected by the chair. If the chair rejects the change, he can finish the interaction or modify his proposal and send it again to the user. Once they have agreed on the conference details, the chair sends the user the specification of the contract, that is, the rights and duties that the user will acquire if he becomes a reviewer. This contract cannot be negotiated, so the user can reject it and finish the interaction or accept it and begin playing the role reviewer within this conference.

2.5 PHASE 5: Agents Description

During this phase, each identified agent is described by means of an instance of the *internal view* metamodel.

2.5.1 Process Roles

The tasks of this phase are executed by the collaboration between the system analyst and the domain expert. The domain expert should provide the information related to agent development requirements. The system analyst should formalize these requirements using the ROMAS diagrams.

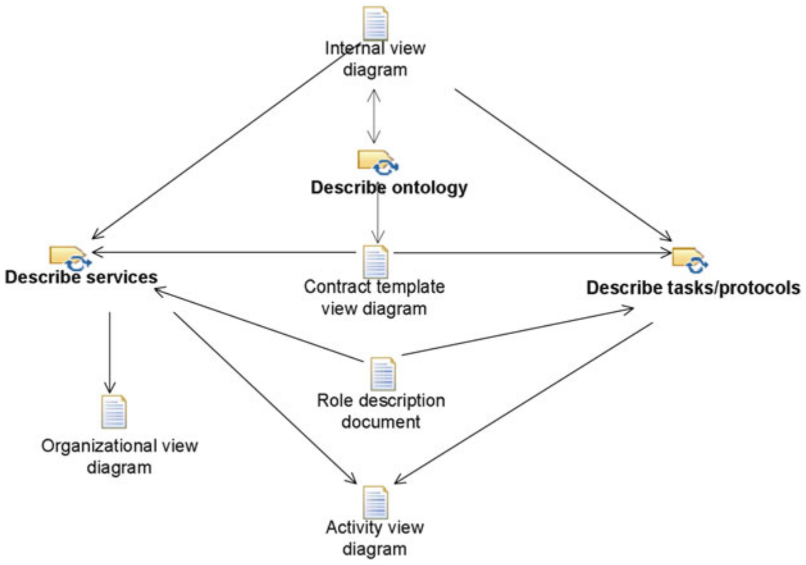


Fig. 24 Phase 4: resources and products used

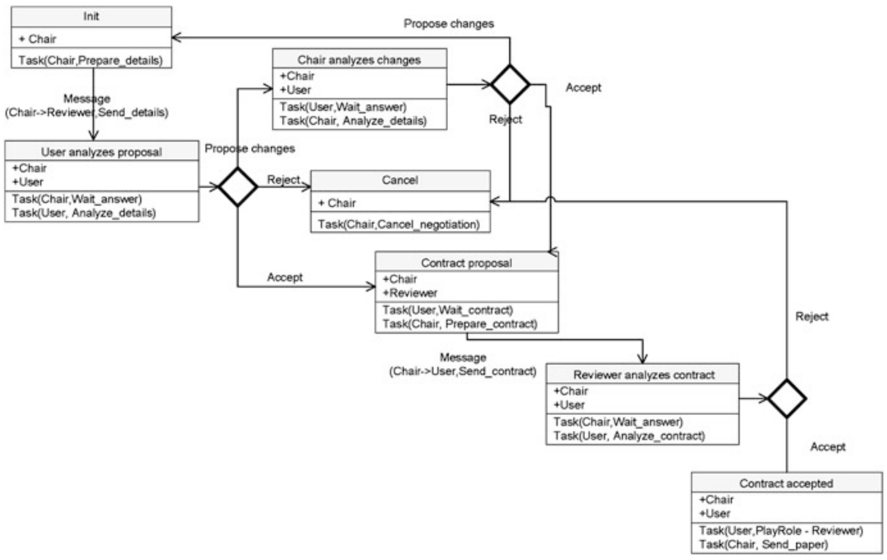


Fig. 25 Phase 4: Case study—reviewer play role negotiation protocol

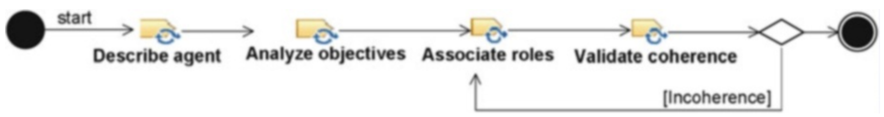


Fig. 26 Phase 5: activity tasks

Table 13 Phase 5: activity tasks

ID.task	Task	Description	Roles involved
4.3	Describe agent	Following the guideline *agent description document*, the development requirements of each agent are analyzed.	System analyst and domain expert
4.1	Analyze objectives	Following the guideline *objectives description document* detailed in Phase 1, the agent's objectives are analyzed and decomposed in operational objectives.	System analyst and domain expert
4.2	Associate with system roles	Identify which roles the agent must play in order to achieve its objectives. This analysis is performed by matching the agent objectives with the roles functionality. Therefore, the *objective description document* of the agent is compared with the analysis of the roles presented in the *roles description documents*.	System analyst
4.4	Validate coherence	Validar que puede cumplir sus objetivos y que puede cumplir las tareas de los contratos y que las normas del sistema no van contra ninguna de sus normas personales	System analyst

2.5.2 Activity Details

The flow of tasks inside this activity is reported in Fig. 26, and the tasks are detailed in Table 13. Figure 27 describes the relation between these work products and the metamodel elements in terms of which elements are *defined* (D), *refined* (F), *quoted* (Q), *related* (R), or *relationship-quoted* (RQ).

2.5.3 Work Products

Figure 28 shows graphically the products used and produced by each task. First, an *agent description document* is created for each agent. Table 14 shows the related guideline and an example from the CMS case study. After that, each identified objective is analyzed following the guideline *objective description document* described in Phase 1. The analysis of the objectives in our running example shows that the main objective of the Ph.D. student agent, *Improve CV*, is decomposed into: *Submit thesis draft*, *Increase number of publications*, and *Collaborate in conferences*. The first objective is not related to any objective of the system, so it cannot be achieved inside the conference management system. The second objective, *Increase number of publications*, could be achieved if the agent joined conferences as an author. The authors' play role contract templates establish that any agent that wants to join a conference as an author should submit an abstract of the paper. Since Bob has unpublished papers that he could submit to a conference, he can play the role *author*.

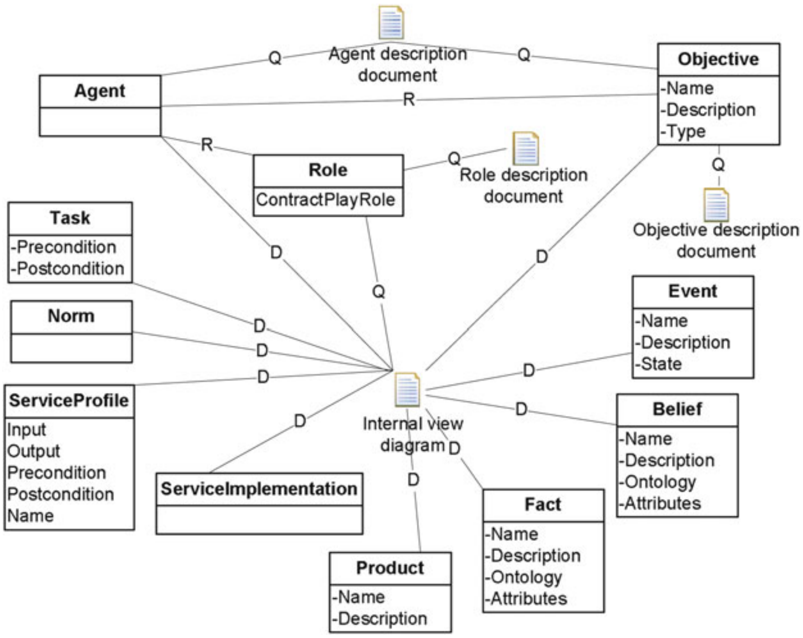


Fig. 27 Phase 5: relations between work products and metamodel elements

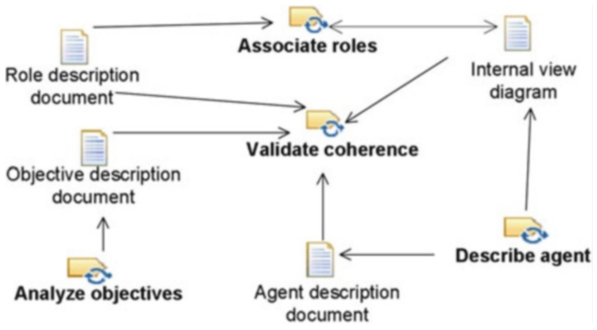


Fig. 28 Phase 5: resources and products used

The third objective, *Collaborate in conferences*, could be achieved by being the PC member of a conference. However, after the validation step, it is shown that Bob cannot play the role PC member because any agent that wants to play this role must be a doctor, and the agent is a Ph.D. student. One *internal view diagram* is created for each to specify the features of each agent. As an example, Fig. 29 shows the internal view diagram of the Ph.D. student agent.

Table 14 Phase 5: agent description document

Property	Description	Example
Identifier	General name of the agent. It is recommended to select a short name or an abbreviation.	Ph.D. student
Description	Informal description of the agent. There is no length limitation on this text.	It is a Ph.D. student who wants to participate in the system in order to improve its CV.
Objectives	Informal description of the agent's purposes of the agent.	- Improve its CV
Resources: Available for the agent	A list of the resources (products, services and applications) that the agent has or provides.	- Unpublished papers
Resources: Required by the agent	A list of the resources (products, services and applications) that the agent requires to develop its functionality. This text should specify which type of access the role needs (reading, executing, writing, partial or full access)	
Events	A list of the events that this agent handles.	
Other memberships	A text explaining if this agent is interacting with other active systems or organizations.	This agent plays the role Ph.D. student inside its college.
Restrictions	A list of the restrictions that are inherent to the agent.	This agent must follow the regulation of its college and that he is responsible of the maintenance of the research group database.

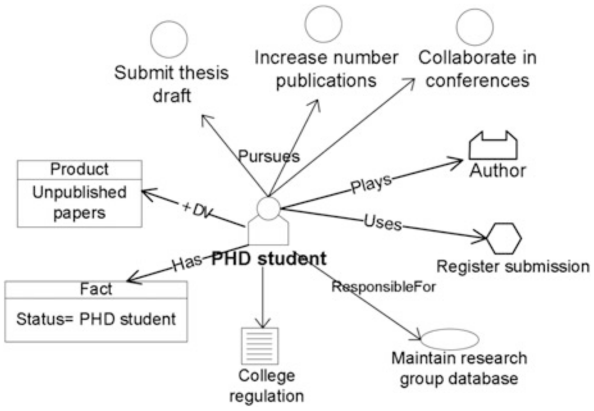


Fig. 29 Phase 5: Case study—Ph.D. student agent description

3 Work Product Dependencies

Figure 30 describes the dependencies among the different work products. For instance, the analysis of the *system description document* is necessary to define the *objective description document* and the *use case diagrams*.

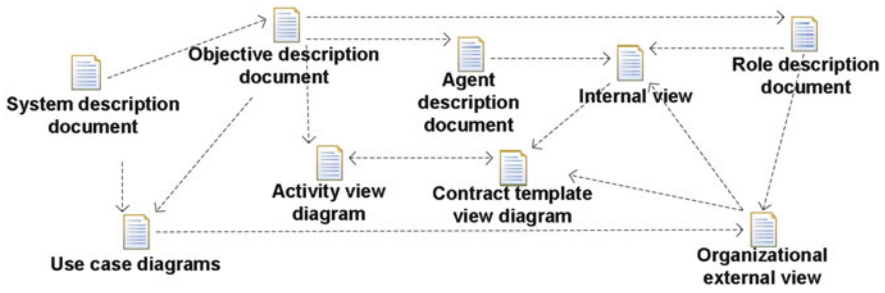

Objective description document
System description document
Agent description document
Internal view
Role description document
Activity view diagram
Contract template view diagram
Use case diagrams
Organizational external view

Fig. 30 Work product dependencies

Acknowledgements This work is partially supported by the TIN2008-04446, TIN2009-13839-C03-01, PROMETEO 2008/051 projects, CONSOLIDER INGENIO 2010 under grant CSD2007-00022 and FPU grant AP2007-01276 awarded to Emilia Garcia.

References

1. Argente, E., Botti, V., Julian, V.: GORMAS: an organizational-oriented methodological guideline for open MAS. In: Agent-Oriented Software Engineering, pp. 85–96. Springer, Berlin (2009)
2. Cossentino, M.: Design process documentation template. Technical Report, FIPA (2010)
3. Fenech, S., Pace, G.J., Schneider, G.: Automatic conflict detection on contracts. In: Proceedings of the 6th International Colloquium on Theoretical Aspects of Computing (ICTAC '09), pp. 200–214 (2009)
4. Garcia, E., Giret, A., Botti, V.: A model-driven CASE tool for developing and verifying regulated open MAS. Sci. Comput. Program. (2011). doi:10.1016/j.scico.2011.10.009
5. Holzmann, G.: Spin Model Checker, the: Primer and Reference Manual. Addison-Wesley Professional, Reading (2003)
6. Tolvanen, J.-P., Marttiin, P., Smolander, K.: An integrated model for information systems modeling. In: Proceedings of the 26th Annual Hawaii International Conference on Systems Science, pp. 470–476. IEEE Computer Society Press, USA (1993)