



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

GAIA

Process documentation



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

GAIA

Process documentation

Date:	28th May 2012
Version:	First Draft
Pages:	
Responsible:	Ambra Molesini
Authors:	Ambra Molesini Luca Cernuzzi Andrea Omicini
Contact:	ambra.molesini@unibo.it lcernuzz@uca.edu.py

Contents

1	Introduction	2
1.1	Process Lifecycle	3
1.2	Meta-model	4
2	Phases of the Gaia process	8
2.1	The Analysis	8
2.1.1	Process roles	8
2.1.2	Activity Details	11
2.1.3	Work products	16
2.2	The Architectural Design	20
2.2.1	Process roles	20
2.2.2	Activity Details	20
2.2.3	Work products	26
2.3	The Detailed Design	30
2.3.1	Process roles	30
2.3.2	Activity Details	31
2.3.3	Work products	32
3	Work products dependencies	36
	Bibliography	38

1

Introduction

The Gaia methodology was initially proposed by [5][6] and subsequently revised and improved by [8]. The original version of Gaia focuses on the analysis and design of closed Multi-Agent Systems (MAS) assuming the benevolence and predisposition to collaborate of the agents in the organizations. The extended version of Gaia is more oriented to designing and building systems in complex, open environments. The Gaia methodology was one of the first appeared in the AOSE field, and offered interesting insights for the development of different other Agent Oriented Software Engineering methodologies. It explicitly focuses on using organizational abstractions to drive the analysis and design of MAS usually characterizing complex and open environments. It models both the macro (social) aspect and the micro (agent internals) aspects of a MAS. Moreover, Gaia devotes a special effort to model the organizational structure and to specify the organizational rules that govern the global behavior of the agents in the organization avoiding conflict based on self-interest actions. An overview of the Gaia methodologies is presented in Figure 1.1.

Useful reference about the Gaia process are the following:

- M. Wooldridge, N. R. Jennings, D. Kinny. A methodology for agent-oriented analysis and design [5]
- M. Wooldridge, N. R. Jennings, D. Kinny. The Gaia methodology for agent- oriented analysis and design [6]
- F. Zambonelli, M. Wooldridge, N.R. Jennings. Developing Multiagent Systems: The Gaia Methodology [8].
- F. Zambonelli, N.R. Jennings, M. Wooldridge. Multiagent Systems as Computational Organizations: the Gaia Methodology [7].
- L. Cernuzzi, F.Zambonelli. Experiencing AUML in the Gaia Methodology [3].
- L. Cernuzzi, F. Zambonelli. Adaptive Organizational Changes in Agent-Oriented Methodologies [4].

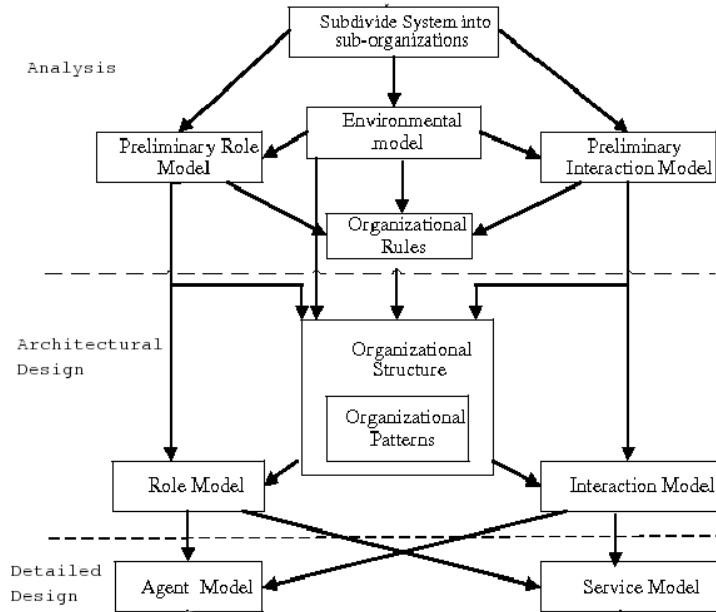


Figure 1.1: An overview of the Gaia process

- L. Cernuzzi, A. Molesini, A. Omicini, F. Zambonelli. Adaptable Multi-Agent Systems: the case of the Gaia Methodologies [2].

1.1 Process Lifecycle

In the original version [6] the development process underpinning Gaia was composed by only two phases: Analysis and Design. However, due to different limitation of the original version of Gaia [1], [8] introduced relevant improvements to better exploit the organizational perspective in the analysis and design activities.

These improvements led to a reformulation of the Gaia's process, including the introduction of a new phase - Architectural Design - and a set of new models. Therefore, actually Gaia proposes three mainly phases, namely, the Analysis, the Architectural Design, and the Detailed Design, as represented in Figure 1.2.

It is worth noting that Gaia does not directly deal with the requirement phase, although the difference with the analysis phase is not always obvious, and implementation issues. The authors believe that requirement elicitation activities should be neutral with respect to the adopted paradigm for the design and the eventual solution technology. Finally, the result of the design phase is assumed to be something that could be implemented in a technology neutral way, that is, by using a traditional method (such as object orientation or component-ware) or an appropriate agent-programming framework.

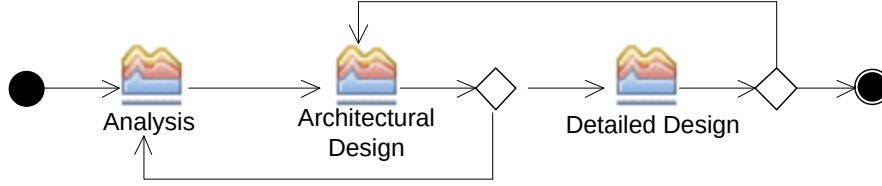


Figure 1.2: The Gaia process phases

The details of each step will be discussed in the following sections.

1.2 Meta-model

The MAS meta-model of Gaia is mostly devoted to represent a MAS system as a social organization and adopts as the basic building blocks organizations, agents, roles, protocols, organizational rules, and environment. The meta-model adopted by Gaia is represented in Figure 1.3, where Gaia abstract entities are depicted along with their mutual relations. In particular: the Gaia agent is an entity that plays one or more roles; a role represents specific behaviors to be played by an agent, defined in terms of responsibilities and activities, and of its interactions with other roles (i.e. protocols); an agent plays a role by actualising the behavior in terms of services.

The primary role of organizational abstractions is represented by mean the organization and the associated organizational rules elements. The MAS organization could be composed by different sub-organizations, each one with a specific structure (i.e. topology) and goals. In turn, each sub-organization is defined as a composition of agents and organizational rules.

The organizational structure describes the overall architecture of the system characterizing the position of each agent (better, of the agents playing specific roles) in it, and the structural relationships with the other agents. Organizational rules have the scope of specifying some constraints that agents in the organization (or sub-organization) have to observe. To some extent, they could be considered as the social laws that have to drive all interactions in the organization and the evolution of the organization itself. They may be global, affecting the behavior of the society as a whole, or concerning only specific roles or protocols.

Finally, the environment abstraction explicitly specifies all the entities and resources a MAS may interact with, according to defined permissions.

Analysis In the Analysis, Gaia covers different concepts of the meta-model. Gaia starts with the identification of the *Sub-Organizations* for the MAS with their respective *Goals*.

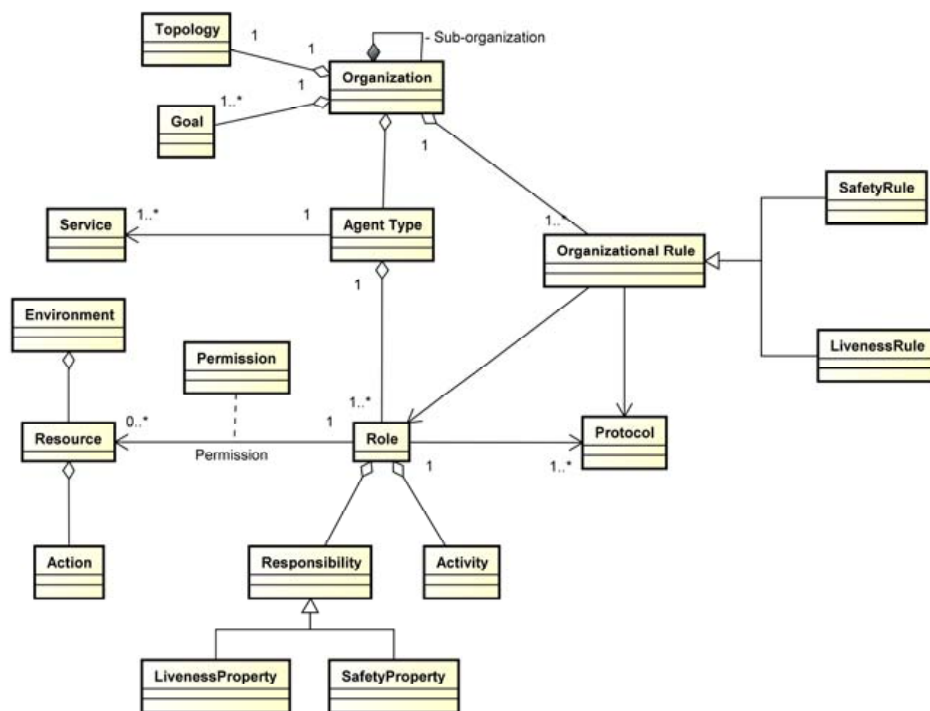


Figure 1.3: The Gaia Meta-model

Then, for each sub-organization it prescribes to specify the Preliminary Role Model identifying a set of Roles (i.e., the functional requirement to achieve the sub-organizational goals). Each *Role* includes a set of Activities and Responsibilities that are in turn composed by *LivenessProperty* and *SafetyProperty* (to state the active and safe behaviors of an agent playing such a role).

Roles are usually interacting among them according to different *Protocols* that are specified in the Preliminary Interaction Model. Moreover, to achieve their objectives *Roles* have to interact with the MAS operation *Environment* (i.e. the Environmental Model) that is composed of a set of computational *Resources*. Each resource enables the role to perform different *Actions* according to specific *Permission*. Finally, system analysts have to specify the Organizational Rules that govern the general behavior of the MAS. Such rules include *LivenessRules* and *SafetyRules* (to specify norms and constraints on the desirable and undesirable behaviors into the sub-organization or in the overall MAS).

Architectural Design The main goal of the Architectural Design phase is defining the most proper Organizational Structure for the MAS, i.e., the *Topology* of the overall MAS and/or of each sub-organization. The *Topology* complements the *Goals* and the *Organizational Rules* (i.e., the control regime) of the sub-organizations. It is possible that new *Roles* and *Protocols* arise according to the adopted organizational structure.

Detailed Design Finally, the Detailed Design covers the specification of the set of *Agent Types*, i.e., the agent classes in the MAS. Each instance of an *Agent Type* is an Agent playing one or more *Roles*. In addition, the set of activities an agent are able to perform are described in term of *Services*.

Table 1.1: The Gaia entities definitions

Concepts	Definition	Phase
Role	Entity representing specific behaviors to accomplish some tasks	Analysis, Architectural Design
Activity	action that a role can perform to reach some specific goal	Analysis, Architectural Design
Responsibility	is the key attribute associated with a role and determine functionality of the role	Analysis, Architectural Design
LivenessProperty	describe those states of affairs that an agent must bring about (intuitively states that “something good happens”, and hence that the agent carrying out the role is alive)	Analysis, Architectural Design

SafetyProperty	an acceptable state of affairs maintained across all states of execution (intuitively states that “nothing bad happens”)	Analysis, Architectural Design
Environment	involves determining all the entities and resources that the MAS, as a whole, can exploit, control or consume when it is working towards the achievement of the organizational goal	Analysis
Resource	elements in the environment for agents to interact with	Analysis
Action	an activity impacting resources of the environment according to roles objectives	Analysis
Permission	identifies the resources that are available to a role in order to realize its responsibilities (i.e. the “rights” associated with a role)	Analysis, Architectural Design
Protocol	define the way that a role can interact with other roles	Analysis, Architectural Design
OrganizationalRule	express general requirements for the proper instantiation and execution of a MAS in term of relationships and constraints between roles, between protocols, and between roles and protocols into the organization	Analysis, Architectural Design
SafetyRule	properties that the system must guarantee to prevent undesirable behaviors into the organization	Analysis, Architectural Design
LivenessRule	properties that the system must guarantee to enable desirable behaviors into the organization	Analysis, Architectural Design
Organization	a group of agents interacting according to a specific organizational structure (i.e., topology) and respecting some defined organizational rules	Architectural Design
Topology	describe the relationships between roles (and therefore their positions) into an organization	Analysis, Architectural Design
Goal	a single objective of an organization that have to be accomplished by one or more agents into the organization	Analysis, Architectural Design
AgentType	defines a class of pro-active, autonomous entity that plays one or more roles	Detailed Design
Service	a function of the agent; that is, simply a single, coherent block of activity that an agent will engage in	Detailed Design

2

Phases of the Gaia process

2.1 The Analysis

The Analysis phase in Gaia covers the requirements in term of functions and activities adopting the agent paradigm according to the organizational perspective. This involves firstly identifying which loosely couple sub-organizations possibly compose the whole systems and then, for each of these, to produce four basic abstract models: (i) the environmental model, to capture the characteristics and their computational representation of the MAS operational environment; (ii) a preliminary roles model, to capture the key task-oriented activities to be played in the MAS; (iii) a preliminary interactions model, to capture basic inter-dependencies between roles and their corresponding protocols; and (iv) a set of organizational rules, expressing global onstraints/directives that must underlie the MAS functioning. The different models can in principle be performed concurrently. However, there exist a strict interdependence between them, since the modification of one specific diagram (e.g. the introduction of a new preliminary role in the sub-organization) typically implies the modification of other diagrams, too.

Figure 2.1 presents the Analysis process, while Figure 2.2 presents the flow of activities, the involved roles and the work products.

2.1.1 Process roles

One role is involved in the Analysis step: the System Analyst.

System Analyst

System Analyst is responsible for:

- identification of the organization structure and its main characteristics such as the couple degree, sub-organizations, and organizational rules
- identification of the preliminary roles and their main characteristics such as skills, permissions, and responsibilities
- identification of the preliminary protocols

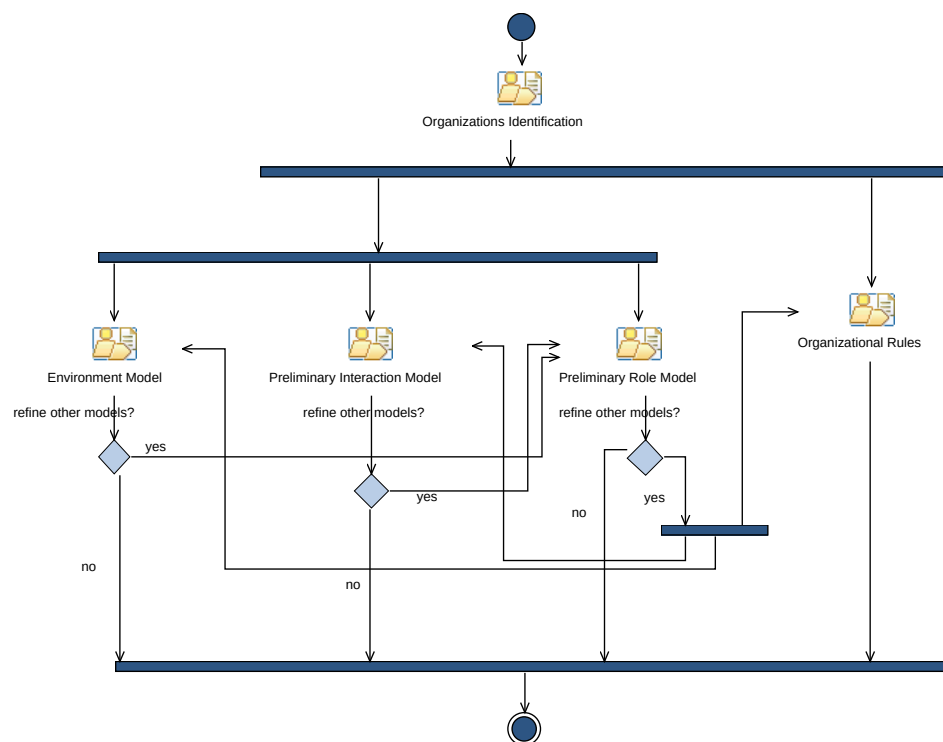


Figure 2.1: The Analysis process

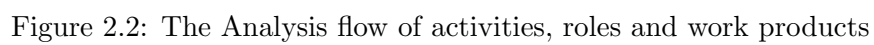


Figure 2.2: The Analysis flow of activities, roles and work products

- identification of the resources presented in the system environment

2.1.2 Activity Details

Organizations Identification activity

The flow of tasks inside this activity is reported in Figure 2.3 and the tasks are detailed in the following table.

Activity	Task	Task Description	Role Involved
Organizations Identification	Identify Organization	the MAS could be seen as a global organization with its general goals	System Analyst (perform)
Organizations Identification	Identify Couple Degree	identifying the couple degree of the sub-organizations	System Analyst (perform)
Organizations Identification	Identify Sub-organizations	identifying how to fruitfully decompose the global organization into loosely coupled sub-organizations	System Analyst (perform)
Organizations Identification	Identify sub-goals	each sub-organization has its own goals that contribute to the goals of the overall MAS	System Analyst (perform)

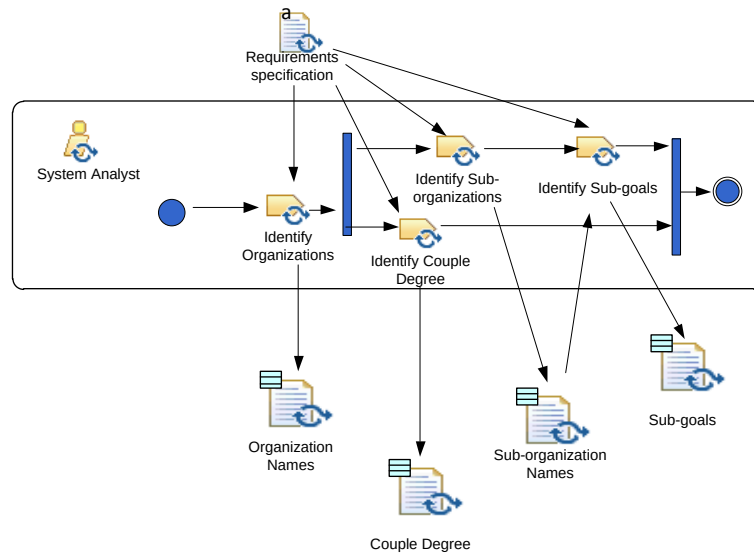


Figure 2.3: The flow of task in the Organizations Identification activity

Environment Model activity

The flow of tasks inside this activity is reported in Figure 2.4 and the tasks are detailed in the following table.

Activity	Task	Task Description	Role Involved
Environment Model	Identify Abstract Computational Resources	a set of variables or tuples, made available to the agents for sensing (e.g., reading their values), for effecting (e.g., changing their values) or for consuming (e.g., extracting them from the environment)	System Analyst (perform)
Environment Model	Describe Abstract Computational Resources	for each resource is described in term of a symbolic name, is characterized by the type of actions that the agents can perform on it, and could present additional textual comments and descriptions	System Analyst (perform)

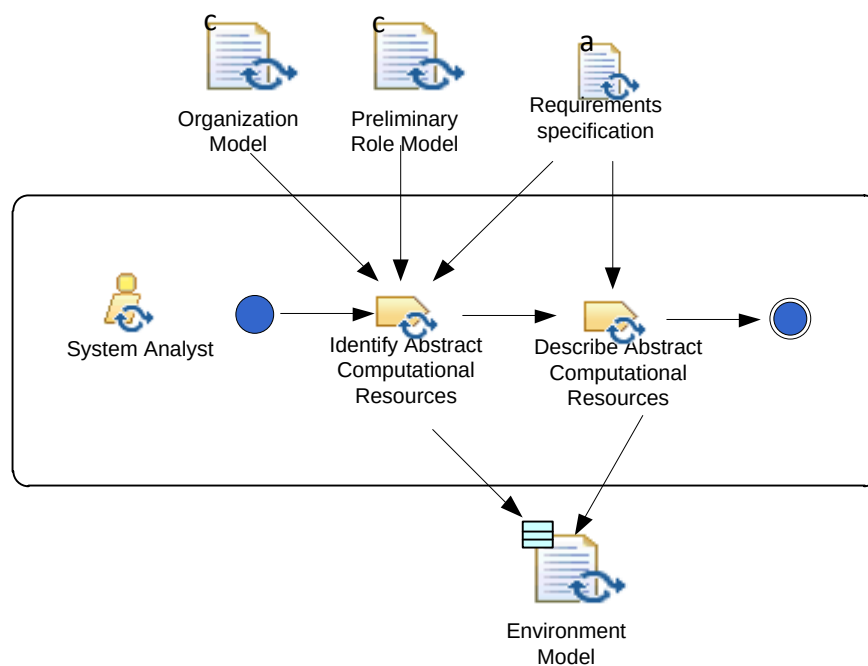


Figure 2.4: The flow of task in the Environment Model activity

Preliminary Role Model activity

The flow of tasks inside this activity is reported in Figure 2.5 and the tasks are detailed in the following table.

Activity	Task	Task Description	Role Involved
Preliminary Role Model	Identify Skills	some characteristics that are required by the organization to achieve its goals	System Analyst (perform)
Preliminary Role Model	Identify Preliminary Roles	basic skills of the organization. Each of them, with their activities, can be carried out from an agent	System Analyst (perform)
Preliminary Role Model	Identify Permissions	permissions relate agent roles to the environment identifying the resources that can be used to carry out the role and stating the resource limits within which the role must operate (e.g. possibly change or consume them)	System Analyst (perform)
Preliminary Role Model	Identify Responsibilities	determine the expected behavior of a role in term of liveness properties and safety properties	System Analyst (perform)
Preliminary Role Model	Create Role-Resource Diagram	a graphical diagram representing what agents playing the role must be allowed to do or must not be allowed to do on the available environmental resources	System Analyst (perform)

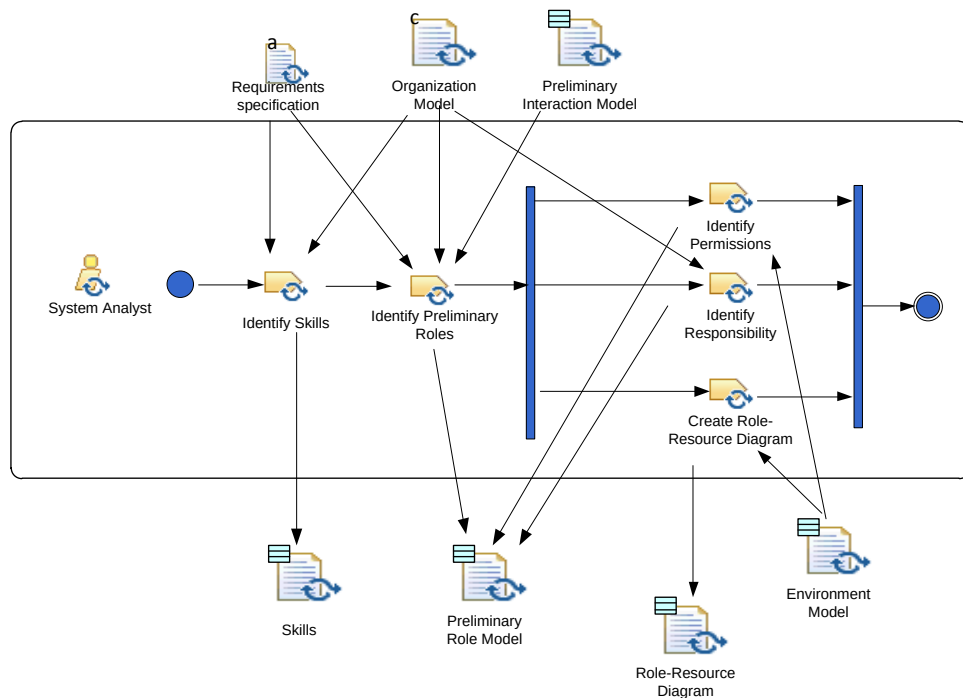


Figure 2.5: The flow of task in the Preliminary Role Model activity

Preliminary Interaction Model activity

The flow of tasks inside this activity is reported in Figure 2.6 and the tasks are detailed in the following table.

Activity	Task	Task Description	Role Involved
Preliminary Interaction Model	Identify Protocol	a pattern of interaction that has been formally defined and abstracted away from any particular sequence of execution steps	System Analyst (perform)
Preliminary Interaction Model	Identify Role Dependencies	relationships and interactions between the various roles in the MAS organization that allow roles to reach their own goals	System Analyst (perform)

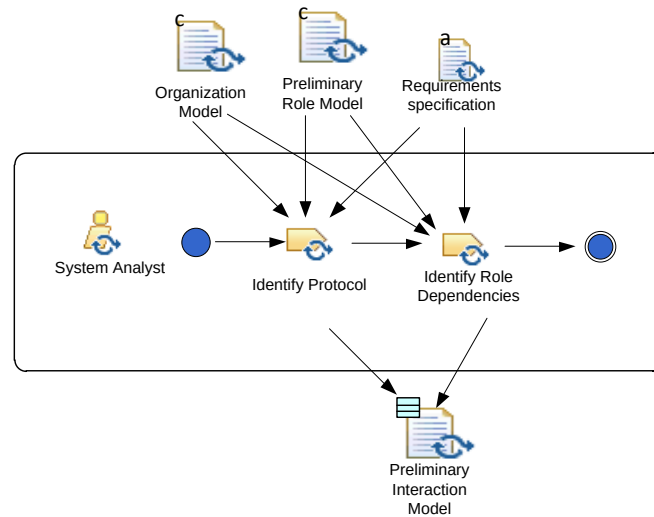


Figure 2.6: The flow of task in the Preliminary Interaction Model activity

Organizational Rules activity

The flow of tasks inside this activity is reported in Figure 2.7 and the tasks are detailed in the following table.

Activity	Task	Task Description	Role Involved
Organizational Rules	Identify Organizational Rules	organizational rules are considered as responsibilities of the organization as a whole and are represented in term of liveness and safety organizational rules	System Analyst (perform)
Organizational Rules	Identify Role-Role Dependencies	norms and constraints on relationships between roles (e.g. a role can be played only if another role has been played before; or that two roles can never be played concurrently by the same entity)	System Analyst (perform)
Organizational Rules	Identify Role-Protocol Dependencies	norms and constraints that relate roles and their interactions (e.g. the protocol must be executed only once by the role)	System Analyst (perform)
Organizational Rules	Identify Protocol-Protocol Dependencies	norms and constraints on the interactions among roles (e.g. the protocol P1 must necessarily be executed before the execution of protocol P2)	System Analyst (perform)
Organizational Rules	Identify Environmental Constraints	norms and constraints on the environmental resources (e.g. each paper will receive three reviews from different referees)	System Analyst (perform)

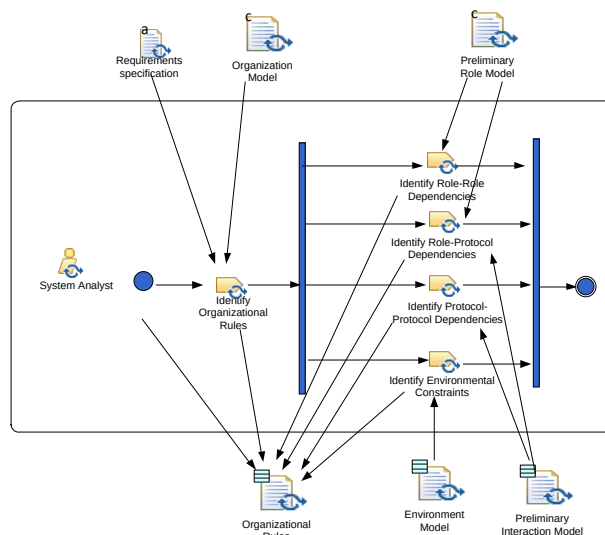


Figure 2.7: The flow of task in the Organizational Rules activity

2.1.3 Work products

Figure 2.8 reports the relationships among the work products of this phase and the MMMElements of the Analysis.

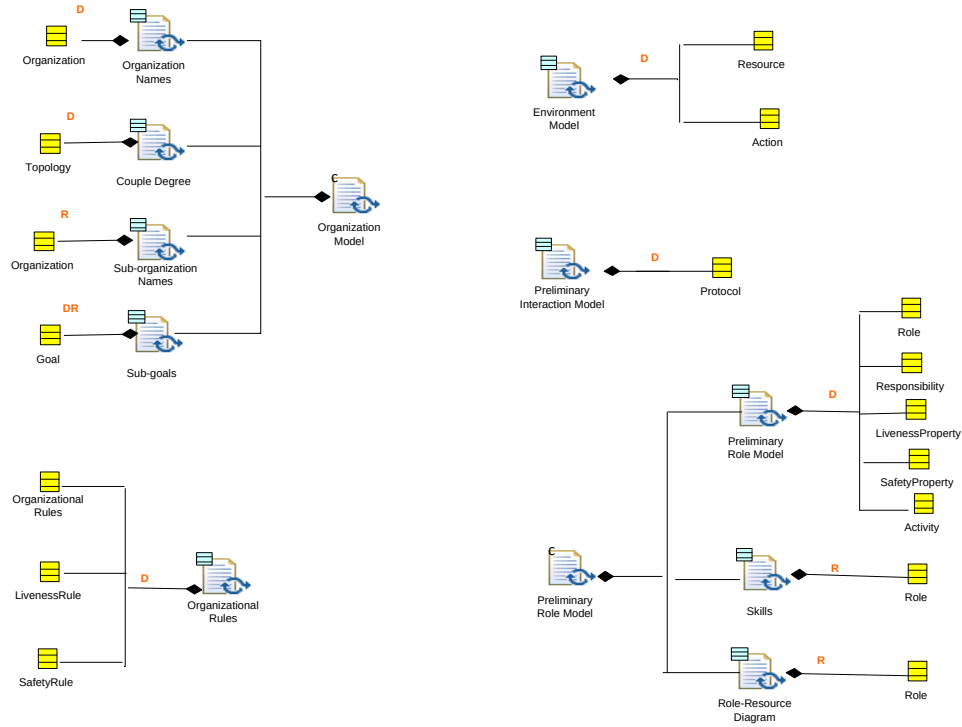


Figure 2.8: The Analysis work products

Work Products Kinds

Table 2.1 describes all the work products of the Analysis.

Table 2.1: Analysis work products kinds

Name	Description	Work Product Kind
Requirement Specification	A description of the problem to be solved (out of the scope of Gaia)	Free text
Organization Model	A composition of others work products that defines the MAS and its components	Composite (Structural)
Organization Names	The name of the overall organization	Structural
Couple Degree	The degree of relationships and dependencies among sub-organizations	Structural

Sub-organization Names	The name of each sub-organization	Structural
Sub-goals	The goals of a sub-organization	Structural
Environment Model	A set of computational resources with their description and the actions an agent can perform on them	Structural
Preliminary Role Model	A composition of others work products that defines the roles	Composite (Structured)
Preliminary Role Model	The basic skills of the organization that can be carried out from an agent	Structural
Skills	Set of basic characteristics that are required by the organization to achieve its goals	Structural
Role-Resource Diagram	Diagram representing the actions allowed or not acceptable on the available environmental resources	Structural
Organizational Rule	Set of norms and constraints that governs the organization (or a sub-organization) as a whole	Structural
Preliminary Interaction Model	Relationships and interactions between the various roles in the MAS	Structural

Work products Example

The following table presents an example of the Organization Model for the conference management case study.

Sub-organization	Couple degree	Sub-goals
Submission Org	loosely coupled	distribution of CFP; paper submission; papers classification; assignment of a submission number; authors notification
Review Org	loosely coupled	papers assignment; providing reviews; collections of reviews; acceptance/rejection; authors notification
Publication Org	loosely coupled	camera-ready production; collecting camera ready; program preparation; composing the proceedings

Table 2.2 presents an example of the Environment Model for the conference management case study.

Table 2.3 presents an example of the Preliminary Role Model for the conference management case study.

Table 2.4 presents an example of the Preliminary Interaction Model for the conference management case study.

Action	Environment Abstraction	Description
Reads	Paper Submitted	the Web site receives a paper
	Review Submitted	the Web site receives a review
Changes	DB Submission	insert in the data base the paper or the review received; one per each track
	DB Reviewer	insert in the data base the personal data of the reviewer, the topic of expertise and the maximum number of papers the referee accepted to review

Table 2.2: The Environment Model for the Review Sub-organization.

Role Name: ReviewCatcher		
Description: This role is in charge of selecting reviewers and distributing papers among them.		
Protocol and Activities: <u>GetPaper</u> , <u>CheckPaperTopic</u> , <u>CheckRefereeExpertise</u> , <u>CheckRefereeConstraints</u> , <u>AssignPaperReferee</u> , <u>ReceiveRefereeRefuse</u> , <u>UpdateDBSubmission</u> , <u>UpdateDBReferee</u>		
Permissions:		
Reads	<i>paper_submitted</i>	in order to check the topic and authors
	<i>referee-data</i>	in order to check the expertise and constraint (i.e. the referee is one of the authors, or belong to the same organization)
Changes	<i>DB Submission</i>	assigning a referee to the paper
	<i>DB Referee</i>	assigning the paper to the referee incrementing the number of assigned papers
Responsibilities:		
<i>Liveness:</i>	ReviewCatcher = (<u>GetPaper</u> . <u>CheckPaperTopic</u> . <u>CheckRefereeExpertise</u> . <u>CheckRefereeConstraints</u> . <u>AssignPaperReferee</u> . <u>ReceiveRefereeRefuse</u>] <u>UpdateDBSubmission</u> . <u>UpdateDBReferee</u>) ⁿ	
<i>Safety:</i>	$\forall \text{ paper: } \text{number_of_referees} \geq n$ Referee $\neq$ Author Referee_organization $\neq$ Author_organization	

Table 2.3: The ReviewCatcher functional role schema.

Protocol Name: AssignPaperReferee		
Initiator: ReviewCatcher	Partner: Reviewer	Input: <i>Submitted Paper</i>
Description: The ReviewCatcher, having checked the expertise of the Reviewer and that she/he is compliant, assigns the paper to the Reviewer which can refuse (in which case ReviewCatcher selects another Reviewer) or accept it (in which case the ReviewCatcher update the DB Submission and DB Referee).		Output: <i>The Reviewer accepts or refuses to review the paper</i>

Table 2.4: The *AssignPaperReferee* interaction protocol.

2.2 The Architectural Design

The Architectural Design phase defines the most proper organizational structure for the MAS, i.e., the topology of interactions in the MAS and the control regime of these interactions, which most effectively enables to fulfill the MAS goals. The definition of the organizational structure has to account for a variety of factors, including the characteristics of the environment, the need of reflecting the structure of the real-world organization, the need of respecting and enforcing the organizational rules, as well as the obvious need to keep the design as simple as possible. In choosing the most appropriate organizational structure it is strongly recommended to exploit well-known organizational patterns. Once the most appropriate organizational structure is defined, the roles and interactions models identified in the analysis phase (which were preliminary, in that they were organizational-independent aspects) can be finalized, to account for all possibly newly identified roles and interactions. These activities result in two models: the Complete Role Model and Complete Interaction Model. The Complete Role and Complete Interaction models are strongly related one to another since any modification of one of the models directly affects the other model.

Figure 2.9 presents the Architectural Design process, while Figure 2.10 presents the flow of activities, the involved roles and the work products

2.2.1 Process roles

One role is involved in the Architectural Design: the Architectural Designer.

Architectural Designer

Architectural Designer is responsible for:

- identify the best organizational structure according to a specific evaluation of organizational forces
- complete role model according to the chosen organizational structure
- complete interaction model according to the chosen organizational structure and the role model
- categorize roles and protocols

2.2.2 Activity Details

Organizational Structure activity

The flow of tasks inside this activity is reported in Figure 2.11 and the tasks are detailed in the following table.

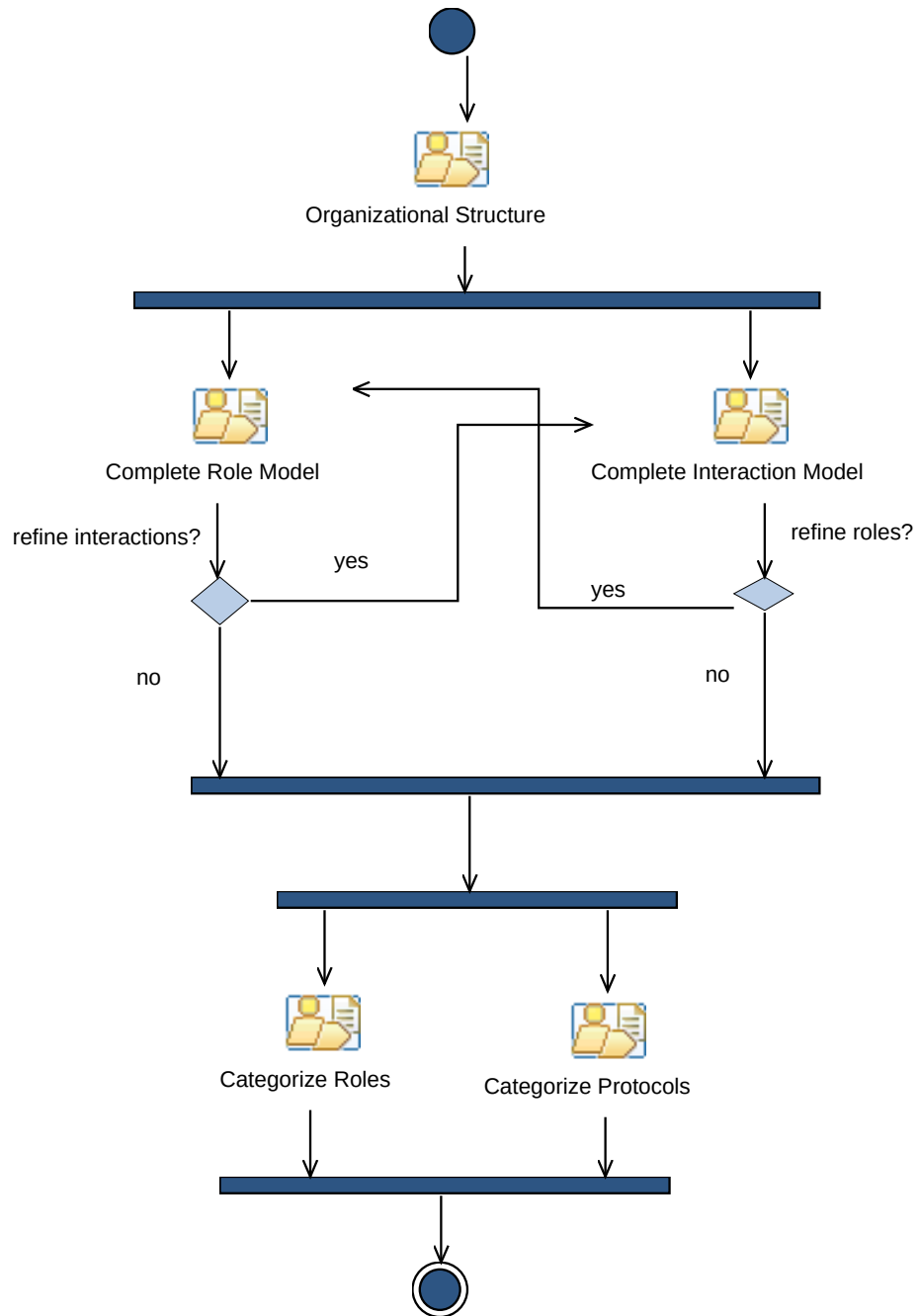


Figure 2.9: The Architectural Design process

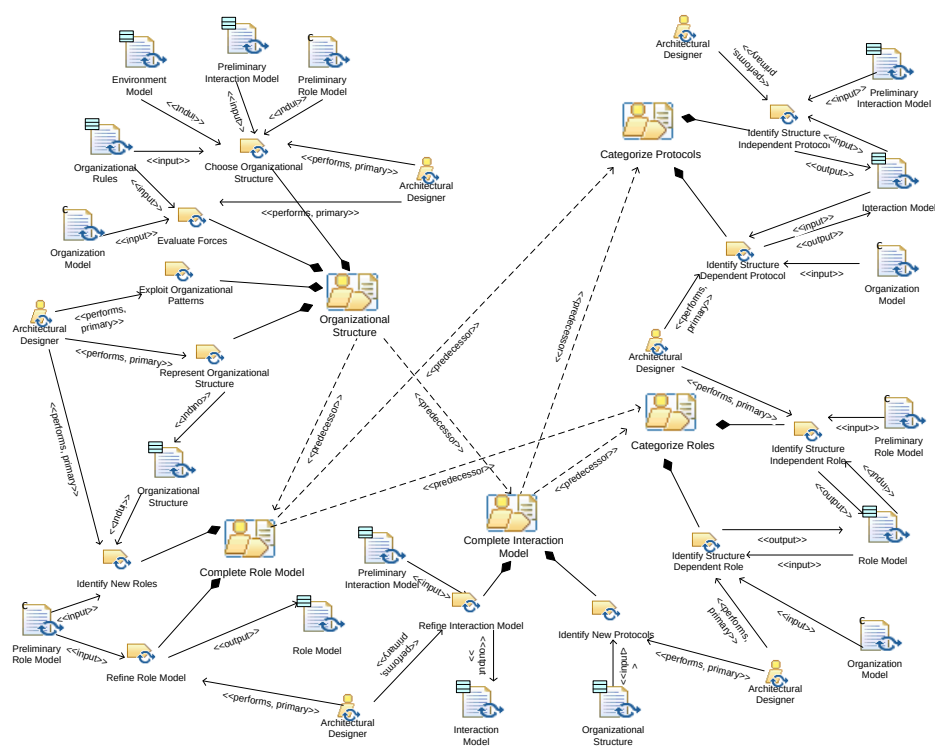


Figure 2.10: The Architectural Design flow of activities, roles and work products

Activity	Task	Task Description	Role Involved
Organizational Structure	Evaluate Forces	considering forces like the need to achieve organizational efficiency; the need to respect organizational rules (i.e. control regime); and the need to minimize the distance from the real-world organization	Architectural Designer (perform)
Organizational Structure	Choose Organizational Structure	identify an efficient and reliable way to structure the MAS organization (i.e. topology)	Architectural Designer (perform)
Organizational Structure	Exploit Organizational Patterns	analyze and possibly adopt well-documented patterns that already demonstrated how and when to exploit a specific organizational structure	Architectural Designer (perform)
Organizational Structure	Represent Organizational Structure	using a diagram to represent the topology of the MAS organization	Architectural Designer (perform)

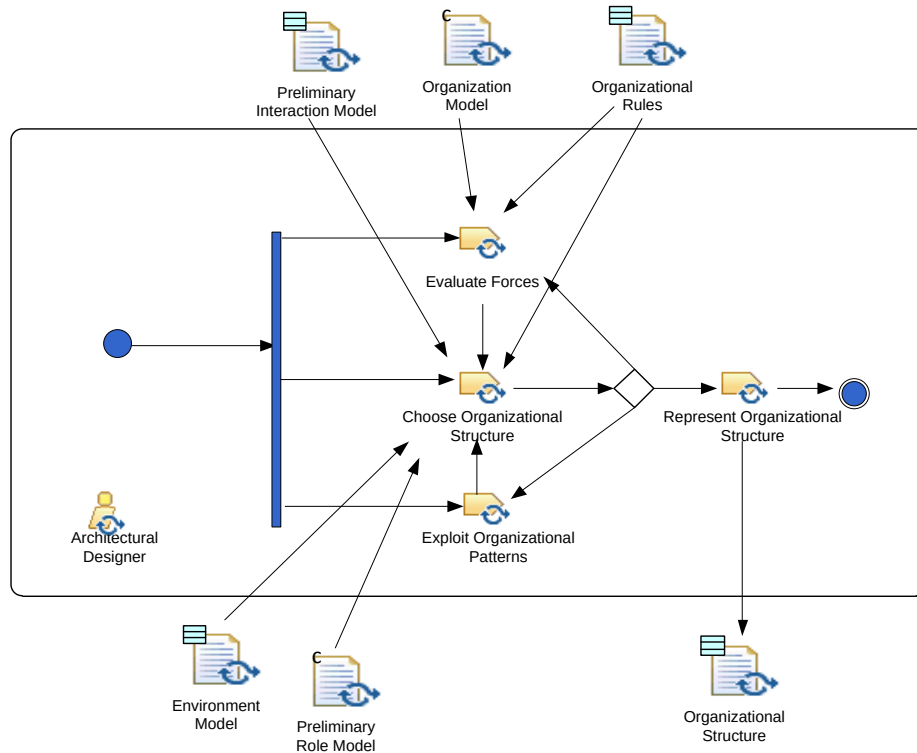


Figure 2.11: The flow of task in the Organizational Structure activity

Complete Role Model activity

The flow of tasks inside this activity is reported in Figure 2.12 and the tasks are detailed in the following table.

Activity	Task		Task Description	Role Involved
Complete Role Model	Identify Roles	New	identification of new roles deriving directly from the adoption of a given organizational structure (i.e. organizational roles)	Architectural Designer (perform)
Complete Role Model	Refine Model	Role	the definitive identification of activities and services of the roles, including the new organizational roles	Architectural Designer (perform)

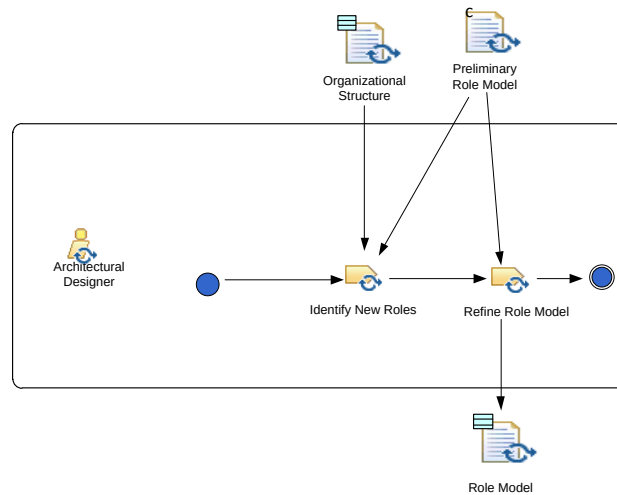


Figure 2.12: The flow of task in the Complete Role Model activity

Complete Interaction Model activity

The flow of tasks inside this activity is reported in Figure 2.13 and the tasks are detailed in the following table.

Activity	Task	Task Description	Role Involved
Complete Interaction Model	Identify New Protocols	identification of new protocols deriving from the adopted organizational structure (i.e. organizational protocols)	Architectural Designer (perform)
Complete Interaction Model	Refine Interaction Model	completing the definition of the protocols required by the MAS, by specifying which roles the protocol will involve	Architectural Designer (perform)

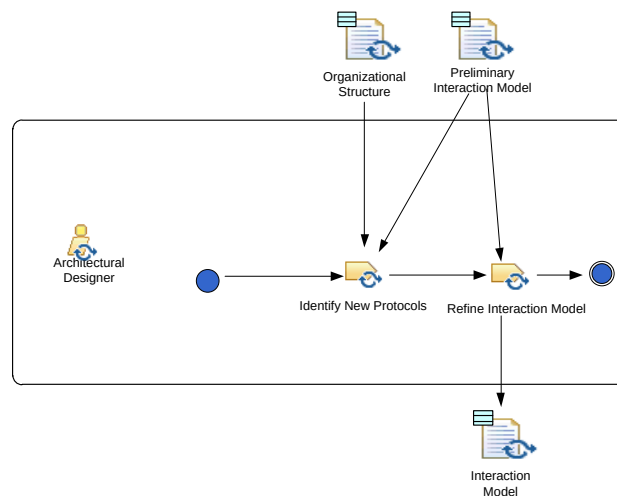


Figure 2.13: The flow of task in the Complete Interaction Model activity

Categorize Roles activity

The flow of tasks inside this activity is reported in Figure 2.14 and the tasks are detailed in the following table.

Activity	Task	Task Description	Role Involved
Categorize Role	Identify Structure Dependent Role	specify which roles are deriving from the adopted organizational structure (i.e. organizational roles)	Architectural Designer (perform)
Categorize Role	Identify Structure Independent Role	specify which roles represent basic skills of the MAS that are independent from the adopted organizational structure	Architectural Designer (perform)

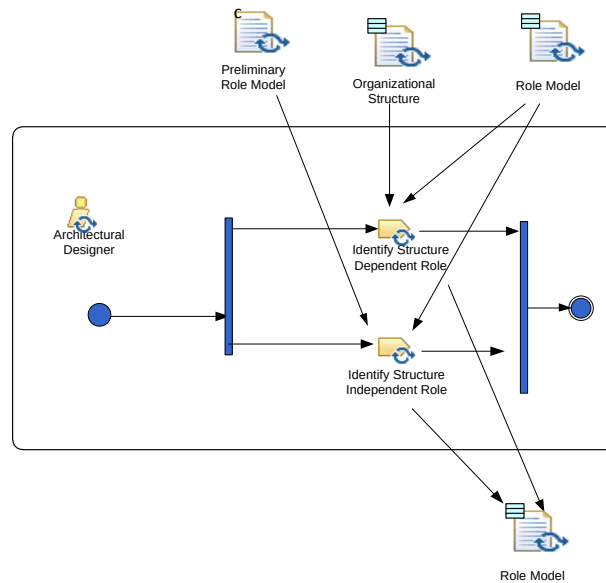


Figure 2.14: The flow of task in the Categorize Role activity

Categorize Protocol activity

The flow of tasks inside this activity is reported in Figure 2.15 and the tasks are detailed in the following table.

Activity	Task	Task Description	Role Involved
Categorize Protocol	Identify Structure Dependent Protocol	specify which protocols are deriving from the adopted organizational structure (i.e. organizational protocols)	Architectural Designer (perform)
Categorize Protocol	Identify Structure Independent Protocol	specify which protocols are independent from the adopted organizational structure	Architectural Designer (perform)

2.2.3 Work products

Figure 2.16 reports the relationships among the work products of this step and the MMMElements of the Architectural Design step.

Work Products Kinds

Table 2.5 describes all the work products of the Architectural Design.

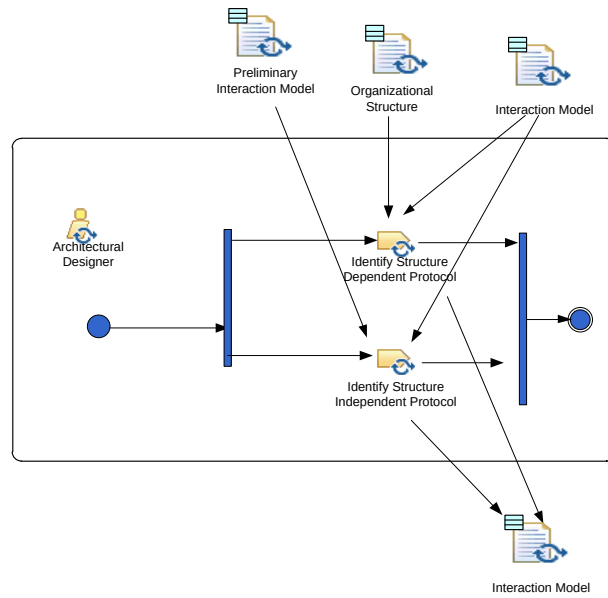


Figure 2.15: The flow of task in the Categorize Protocol activity

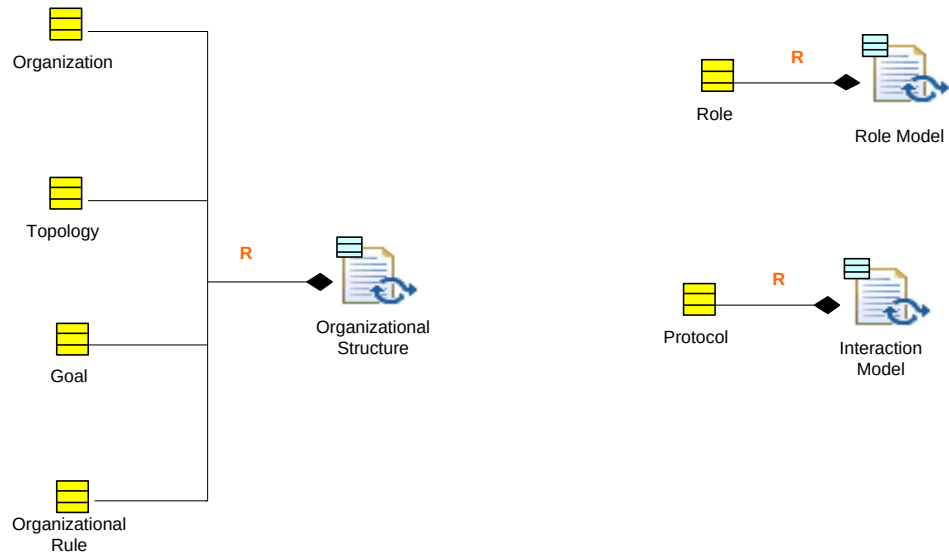


Figure 2.16: The Architectural Design work products

Table 2.5: Architectural Design work products kinds

Name	Description	Work Kind	Product
Organizational Structure	Diagram representing the topology of the MAS organization	Structural	
Role Model	The definitive specification of activities and services of the basic roles and the organizational roles (depending on the adopted organizational structure)	Structural	
Interaction Model	The definitive specification of the protocols (and the corresponding roles involved) required by the MAS	Structural	

Work Products Example

Figure 2.17 presents an example of the Organizational Structure for the conference management.

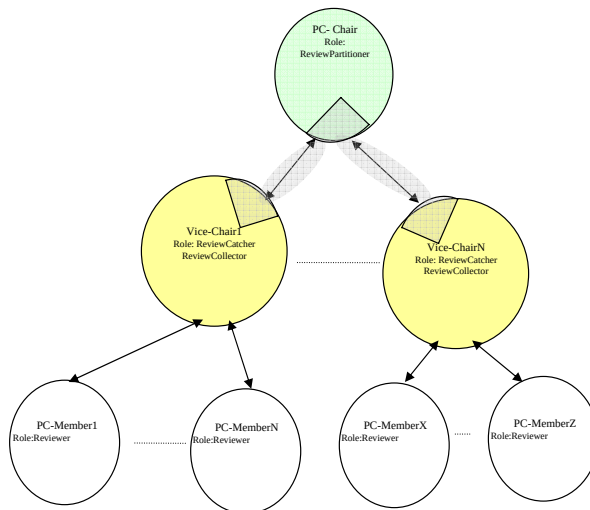


Figure 2.17: Organizational Structure in the conference management case study

Table 2.6 presents an example of the Complete Role Model for the conference management case study.

Table 2.7 presents an example of the Complete Interaction Model for the conference management case study.

Role Name: ReviewPartitioner	
Description: This role is in charge of distributing papers among Vice-Chairs according the area of competence.	
Category: Organizational role	
Protocol and Activities: <u>GetPaper</u> , <u>CheckPaperTopic</u> , <u>CheckViceChairArea</u> , <u>AssignPaperViceChair</u> , <u>UpdateDBSubmission</u> .	
Permissions: Reads <i>paper_submitted</i> in order to check the topic and authors <i>Vice-Chair-data</i> in order to check the area Changes <i>DB Submission</i> assigning the paper to a Vice-Chair area	
Responsibilities: <i>Liveness:</i> ReviewPartitioner = (<u>GetPaper</u>.<u>CheckPaperTopic</u>.<u>CheckViceChairArea</u>.<u>AssignPaperViceChair</u>.<u>UpdateDBSubmission</u>)^w <i>Safety:</i> $\forall$ <i>paper assigned to a ViceChairArea</i>	

Table 2.6: The ReviewPartitioner organizational role schema.

Protocol Name: ReceivePaperAssignment		
Category: organizational protocol		
Initiator: ReviewCatcher	Partner: ReviewPartitioner	Input: <i>paper_submitted</i>
Description: The ReviewPartitioner, having checked the area of the paper, assigns the paper to the corresponding ReviewCatcher (the Vice-Chair in charge of that area).		Output: <i>The paper is assigned to a specific area and the DB Submission is updated</i>

Table 2.7: The *ReceivePaperAssignment* interaction protocol.

2.3 The Detailed Design

Past the architectural design phase, the Detailed Design involves identifying: (i) an agent model, i.e., the set of agent classes in the MAS, implementing the identified roles, and the specific instances of these classes; and (ii) a services model, expressing services (a service is a single, coherent block of activity in which an agent will engage) and interaction protocols to be provided within agent classes. An agent class could package a number of closely related roles for the purposes of convenience or efficiency. It is noting to remember that the detailed design models avoid specification oriented to be implemented in a specific technology leaving this issue to programmer creativity.

Figure 2.18 presents the Detailed Design process, while Figure 2.19 presents the flow of activities, the involved roles and the work products

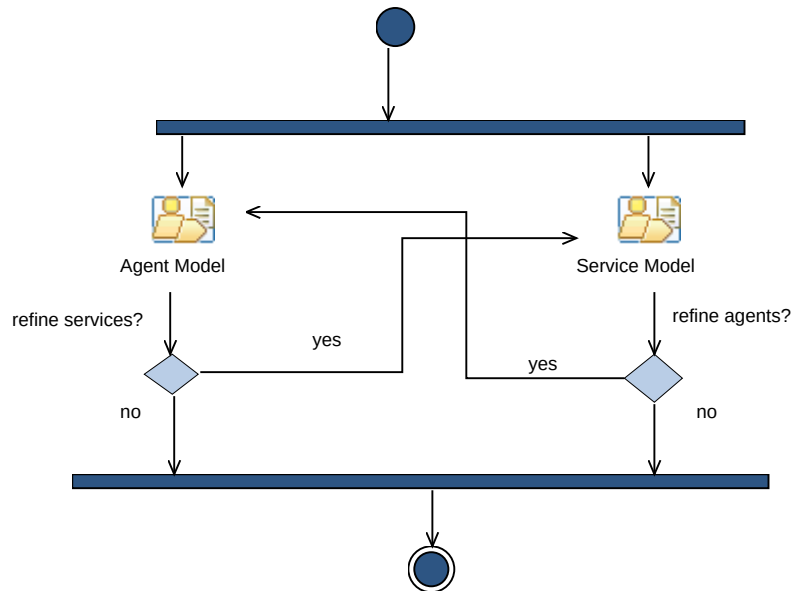


Figure 2.18: The Detailed Design process

2.3.1 Process roles

One role is involved in the Layering pattern: the Detailed Designer.

Detailed Designer

Detailed Designer is responsible for:

- identify agent-types and their cardinalities
- identify services

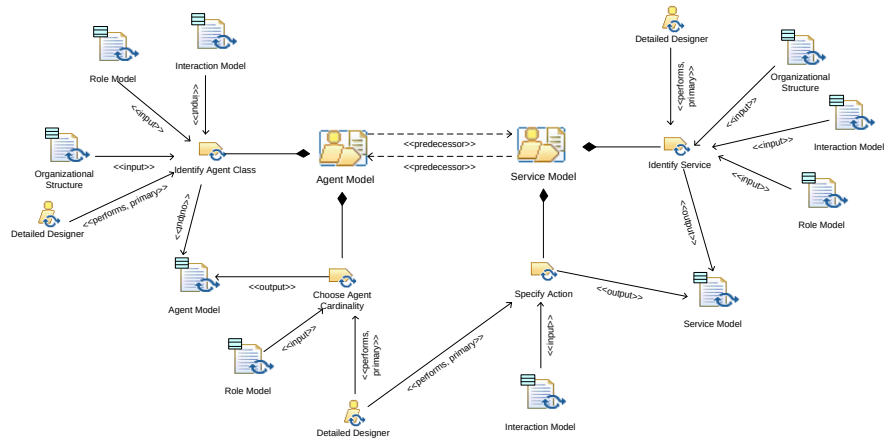


Figure 2.19: The Detailed Design flow of activities, roles and work products

- specify actions

2.3.2 Activity Details

Agent Model activity

The flow of tasks inside this activity is reported in Figure 2.20 and the tasks are detailed in the following table.

Activity	Task	Task Description	Role Involved
Agent Model	Identify Agent Class	identification of a set of pro-active, autonomous entities (i.e. agents), each of them playing one or more roles, and the instances of each class that will appear in the MAS	Detailed Designer (perform)
Agent Model	Choose Agent Cardinality	specify how many instances of each class will appear in the MAS using the cardinality standard	Detailed Designer (perform)

Service Model activity

The flow of tasks inside this activity is reported in Figure 2.21 and the tasks are detailed in the following table.

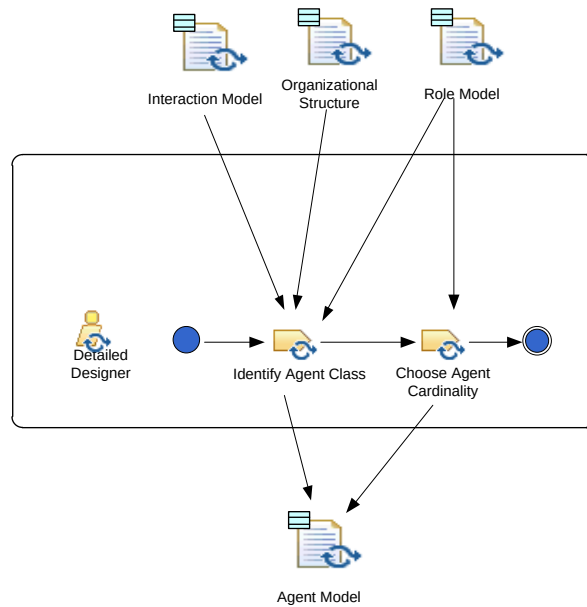


Figure 2.20: The flow of task in the Agent Model activity

Activity	Task	Task Description	Role Involved
Service Model	Identify Service	identify the services associated with each agent class with their inputs, outputs, pre-conditions and post-conditions	Detailed Designer (perform)
Service Model	Specify Action	each service specify a single, coherent block of activity in which an agent will be engaged	Detailed Designer (perform)

2.3.3 Work products

Figure 2.22 reports the relationships among the work products of this step and the MMMElements of the Detailed Design step.

Work Products Kinds

Table 2.8 describes all the work products of the Detailed Design.

Table 2.8: Detailed Design work products kinds

Name	Description	Work Product Kind
Agent Model	Diagram of the agents and the roles they are playing	Structural

Service Model	Specification of the actions associated with each agent class	Structural
---------------	---	------------

Work Products Example

Figure 2.23 and Table 2.9 present an example respectively of the Agent Model and the Service Model for the conference management case study.

Service	Input	Output	Pre-conditions	Post-conditions
Accept/Decline papers Assignment	Request for reviewing papers	Papers to be reviewed	Papers are available for reading (see Environmental Model)	List of papers accepted for reviewing
Accepted Review	Assigned paper	Review form fulfilled	Paper is available for reading Agent has not already reviewed the maximum allowed number of paper	The Review Form has been correctly completed

Table 2.9: An example of the Service Model

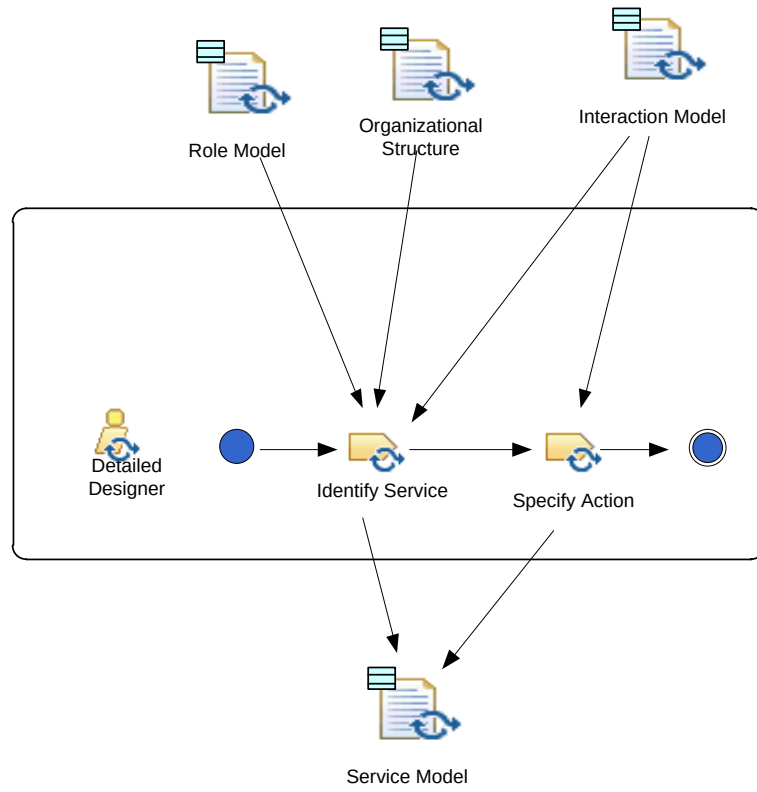


Figure 2.21: The flow of task in the Service Model activity

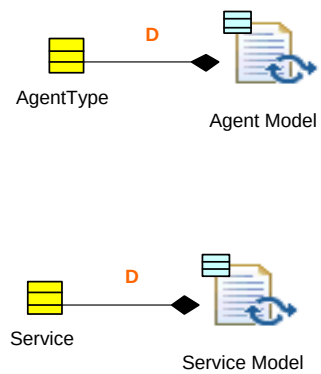


Figure 2.22: The Detailed Design work products

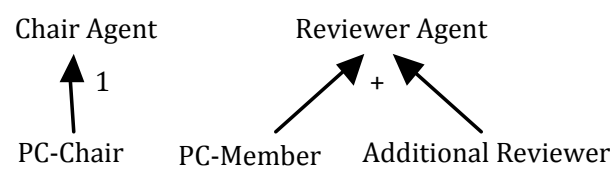


Figure 2.23: An example of the Agent Model

3

Work products dependencies

The diagram in Figure 3.1 describes the dependencies among the different work products.

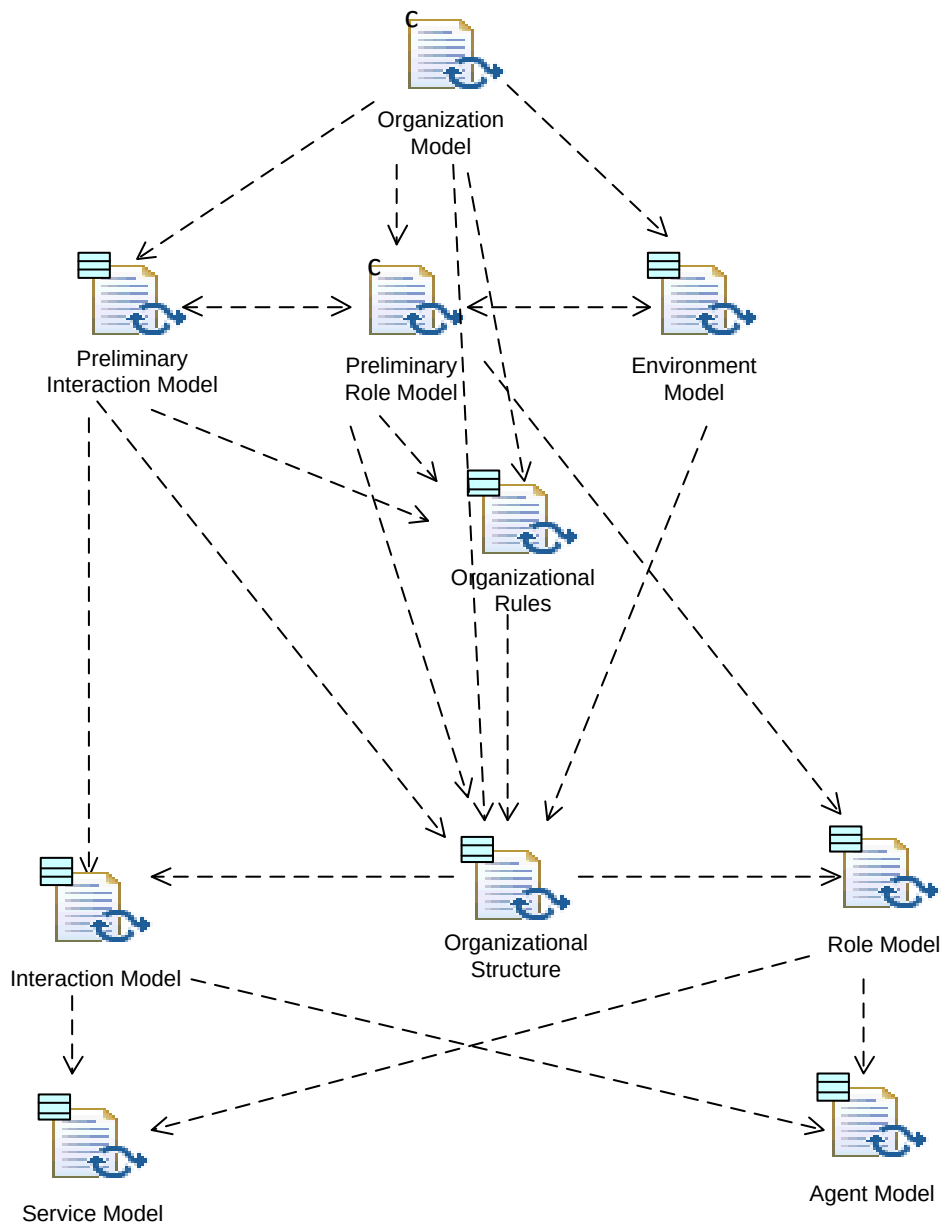


Figure 3.1: The work products dependencies

Bibliography

- [1] Luca Cernuzzi, Thomas Juan, Leon Sterling, and Franco Zambonelli. The gaia methodology. In Federico Bergenti, Marie-Pierre Gleizes, and Franco Zambonelli, editors, *Methodologies and Software Engineering for Agent Systems. The Agent-oriented Software Engineering Handbook*, chapter 4, pages 69–88. Springer-Verlag, New York Inc, 2004.
- [2] Luca Cernuzzi, Ambra Molesini, Andrea Omicini, and Franco Zambonelli. Adaptable multi-agent systems: The case of the Gaia methodology. *International Journal of Software Engineering and Knowledge Engineering*, 21(4):491–521, June 2011.
- [3] Luca Cernuzzi and Franco Zambonelli. Experiencing auml in the gaia methodology. In *ICEIS 2004*, volume III, pages 283–288, 2004. Proceedings of the 6th International Conference on Enterprise Information Systems, Porto, Portugal, April 14-17, 2004.
- [4] Luca Cernuzzi and Franco Zambonelli. Adaptive organizational changes in agent-oriented methodologies. *Knowledge Eng. Review*, 26(2):175–190, 2011.
- [5] Michael Wooldridge, Nicholas R. Jennings, and David Kinny. A methodology for agent-oriented analysis and design. In Oren Etzioni, Jörg P. Müller, and Jeffrey M. Bradshaw, editors, *3rd Annual Conference on Autonomous Agents (AGENTS’99)*, pages 69–76, New York, NY, USA, 1999. ACM.
- [6] Michael Wooldridge, Nicholas R. Jennings, and David Kinny. The gaia methodology for agent-oriented analysis and design. *Autonomous Agents and Multi-Agent Systems*, 3(3):285–312, 2000.
- [7] Franco Zambonelli, Nicholas Jennings, and Michael Wooldridge. Multiagent systems as computational organizations: the Gaia methodology. In Brian Henderson-Sellers and Paolo Giorgini, editors, *Agent Oriented Methodologies*, chapter VI, pages 136–171. Idea Group Publishing, Hershey, PA, USA, June 2005.
- [8] Franco Zambonelli, Nicholas R. Jennings, and Michael Wooldridge. Developing multiagent systems: The gaia methodology. *ACM Transaction on Software Engineering Methodologies*, 12(3):317–370, 2003.