



Adelfe fragmentation

Noelie Bonjean, Tom Jorquera, Carole Bernon, Marie-Pierre Gleizes, Christine Maurel, Frédéric Migeon

► To cite this version:

Noelie Bonjean, Tom Jorquera, Carole Bernon, Marie-Pierre Gleizes, Christine Maurel, et al.. Adelfe fragmentation. [Research Report] IRIT/RR-2009-26-FR, IRIT : Institut de Recherche Informatique de Toulouse. 2009. hal-03798559

HAL Id: hal-03798559

<https://hal.science/hal-03798559>

Submitted on 5 Oct 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

ADELFE FRAGMENTATION

Tom Jorquera
Christine Maurel
Frédéric Migeon
Marie-Pierre Gleizes
Carole Bernon
Noélie Bonjean

IRIT
Université Paul Sabatier
118, route de Narbonne
31 062 Toulouse Cedex 9

Rapport IRIT/RR--2009-26—FR
Septembre 2009

The creation of the new design process for a specific situation using the method engineering approach is based on the composition of a set of reusable method fragments. Thus, we have fragmented ADELFE, an agent-oriented methodology for designing Adaptive Multi-Agent System (AMAS). For describe identified fragments, we use size of documentation which proposed by Cossentino and al.

1. Rules for fragmentation

ADELFE is divided in Work Definitions, Activities and Steps. So, one of this three levels will be opted for unit of fragment decomposition. Work Definition contains a lot of activities of development process. Conversely, Step is an atomic element which enrolls in activities. Step isn't enough "self-sufficient" to establish the fragmentation on steps. On the other hand, an Activity represent "working unit", thus it corresponds at the level of detail looked for a fragment.

That's why activity was held like unit to the granularity level of fragmentation. However, an activity doesn't necessarily correspond to fragment. In fact, some activities are tightly linked by produced documents or by meta-model elements, to be among the same fragment. These activities do not join together in ADELFE because they bring into play different actors. Conversely, ADELFE activity can decompose in several fragments as we are going to see it.

Moreover a part of fragmentation work was to determine echoed them of the fragmentation with regard to the meta-model elements of ADELFE. The result of this work is summarized in section 3.b.

2. Item of fragmentation

a. Fragmentation of the WD1

The first phase of process is composed by two fragments: *Requirements Description* which includes the activities A1, A2 and A3, and *Finalize Requirements* which includes the activities A3 and A4.

In this work definition, we have chosen to bring the three first activities. They aim at identify and clarify the user requirements. They work on a *Requirement Set* although they are separated in ADELFE since different actors step in.

However the activities A4 and A5 are in second fragment because their objective is to formalize requirement expression. Moreover A4 and A5 are tightly linked and will be often realized together because requirements set [final] is going to use keywords set [final] defined in glossary.

This fragmentation is depicted in Figure 1.

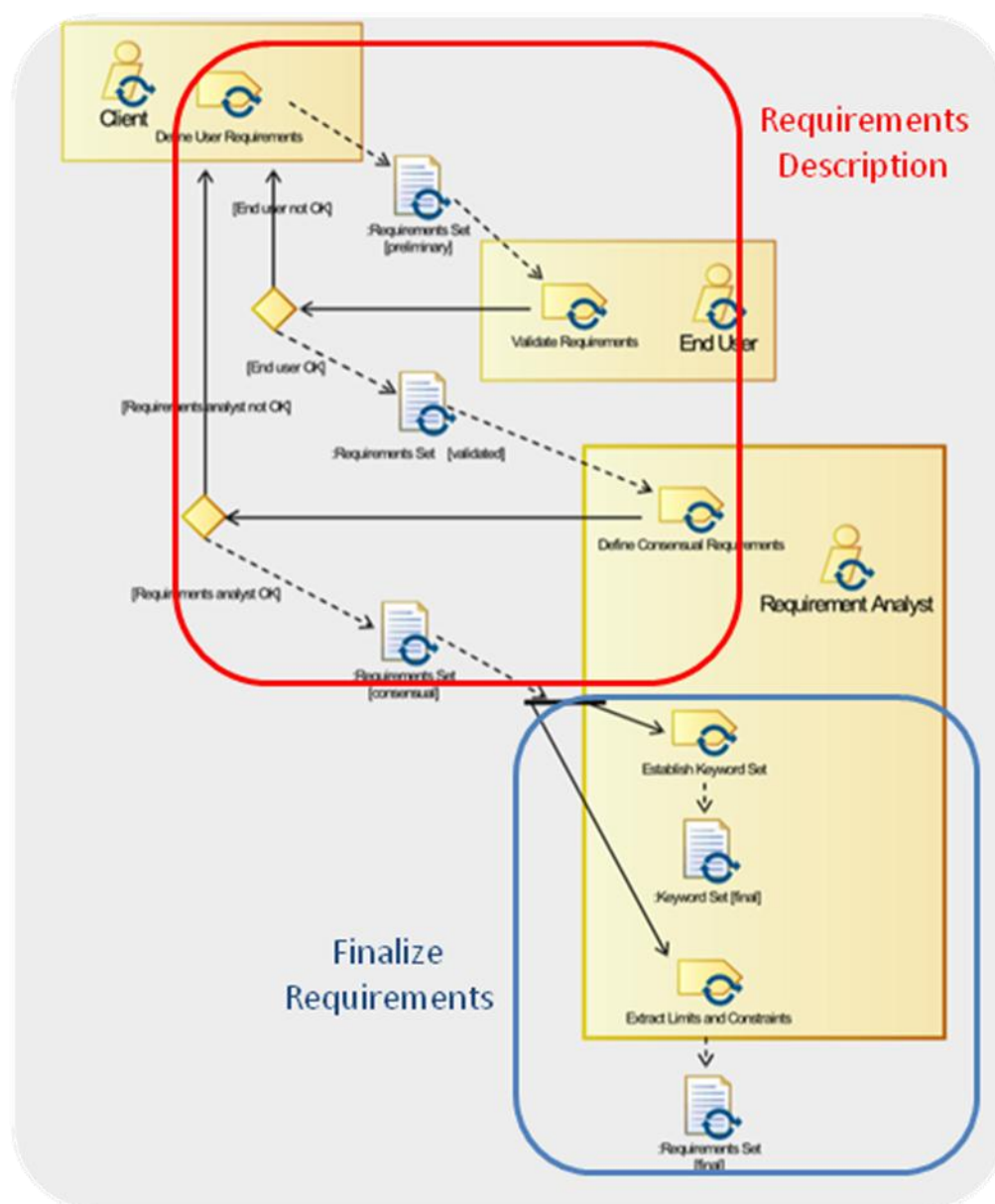


FIGURE 1: FRAGMENTATION OF THE WD1

The aim of fragment *Requirement Description* is to produce requirements set for the system to be designed.

The aim of fragment *Requirement Description* is to finalize requirements set and also to produce keywords set.

b. Fragmentation of the WD2

The second phase of process is composed by three fragments: *Environment Description* which matches to activity A6, *Use Case Definition* which matches to activity A7 and *UI Prototyping* which includes the activities A8 and A9.

A8 and A9 are together because these two activities are tightly bound to the work of prototyping.

This fragmentation is depicted in Figure 2.

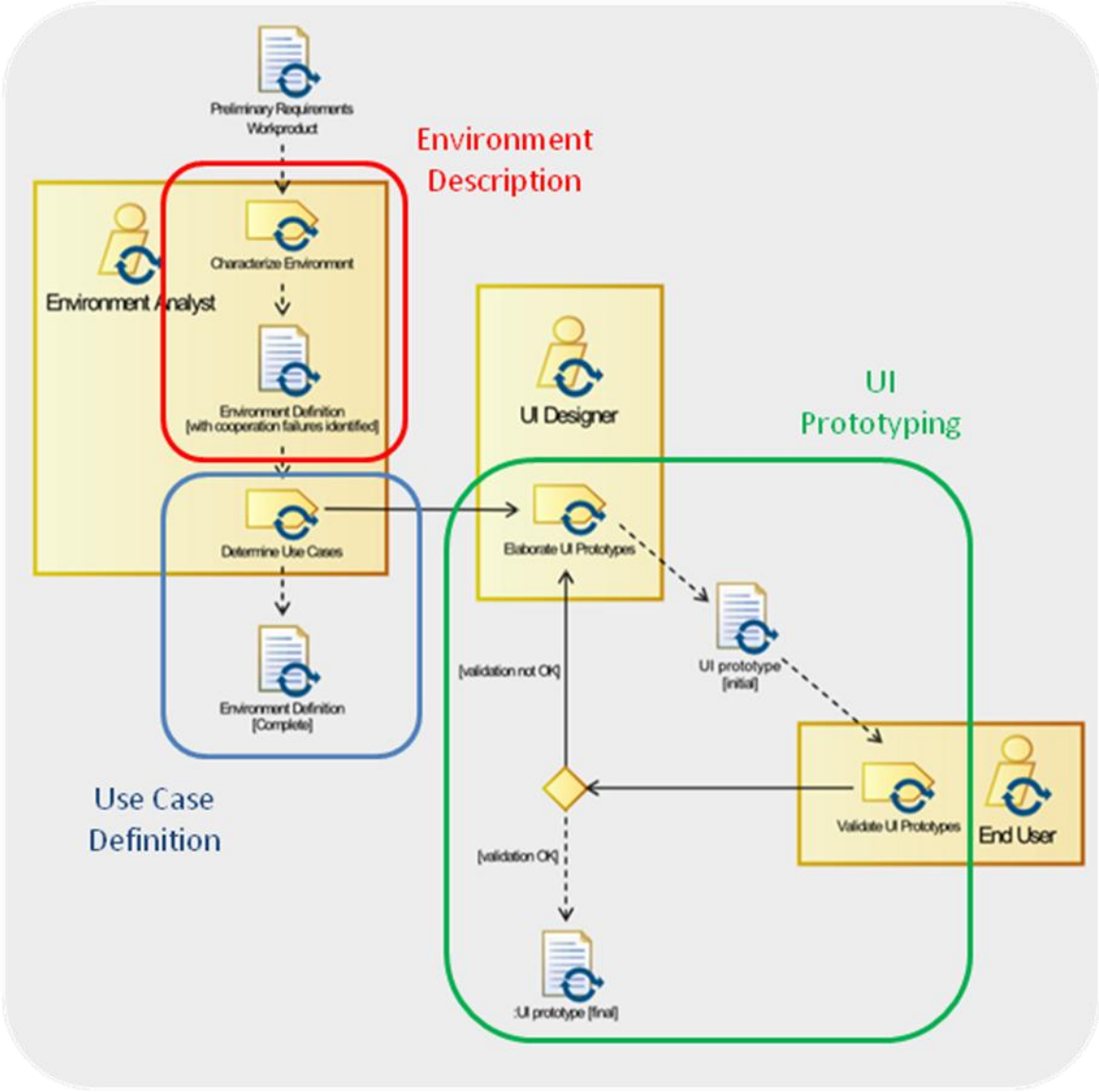


FIGURE 2: FRAGMENTATION OF THE WD2

The role of these fragments is equal to these of corresponding activities.

c. Fragmentation of the WD3

The third phase of process is composed by four fragments: *Domain Analysis* which matches to activity A10, *Verify AMAS Adequacy* which matches to activity A11, *Agent Identification* which matches to activity A12 and *Entities Interaction Study* which matches to activity A13

This fragmentation is depicted in Figure 3.

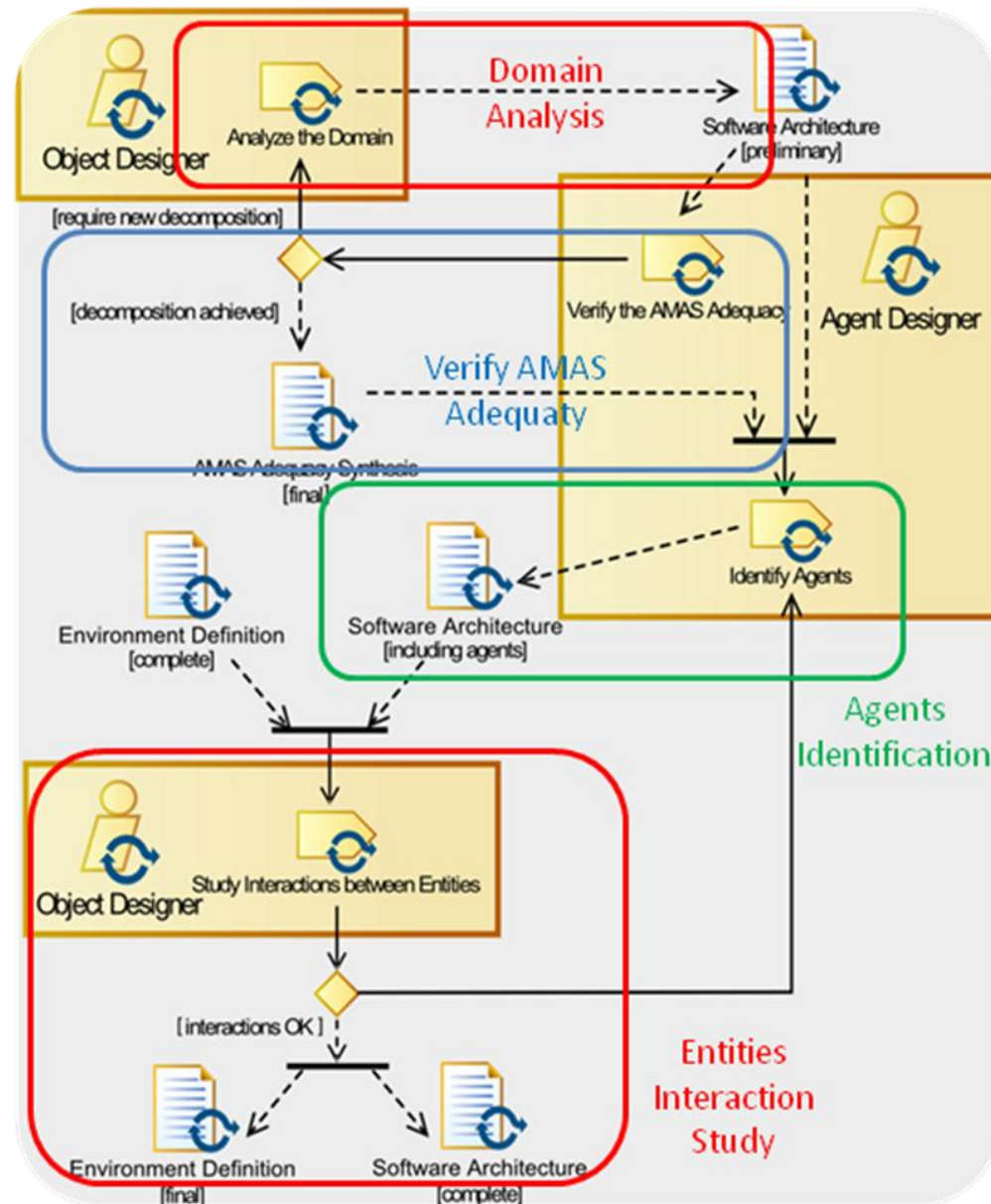


FIGURE 3: FRAGMENTATION OF THE WD3

The role of these fragments is equal to these of corresponding activities.

d. Fragmentation of the WD4

The fourth phase of process is composed by six fragments: *Architecture Definition* which matches to activity A14, *Interaction Language Definition* which matches to activity A15, *Agent Definition* which matches with one part of activity A16, *Non Cooperative Situation Definition* which matches with the other part of activity A16, *Fast Prototyping* which matches to activity A17 and *Architecture Refinement* which matches to activity A18.

This fragmentation is depicted in Figure 4.

This Work Definition is the one which was divided in most large number of fragments. In fact the parting of interaction language definition and agent definition allow to combine ADELFE with others agent architectures. We went farther by dividing A16 into two fragments (the one divide into several fragments). This fragmentation joins in a more global work of the team SMAC which advances a separation between the nominal behavior and the cooperative behavior.

Finally, it is not so surprising to notice that this WD is the one which corresponds to most large number of fragments, given the detail the attention of which it paid in the process ADELFE.

Fragments summary table

| | Title | WD-Activities | Description | Required Documents | Product Documents | Actors | Input | Define | Relate | Quote |
|---|--------------------------|----------------|---|---|--|---------------------------------------|---|---|---|-------|
| 1 | Requirements Description | WD1-A1, A2, A3 | The goal of this fragment is to produce a list of (text) requirements for the system to be designed. | none | Requirement set [consensual] | Client, End User, Requirement Analyst | - | Actor Functional Requirement Non Functional Requirement | Actor-Functional requirement Functional Requirement – non Functional Requirement Actor-Functional requirement | - |
| 2 | Finalize Requirements | WD1-A4, A5 | The goal of this fragment is to produce a finalized list of (text) requirements for the system to be designed and a list of keywords to define the concepts used to describe the application. | Requirement set [consensual] | Requirement Set [final], keywords set | Requirement Analyst | Actor Functional Requirement Non Functional Requirement | Keyword Constraint Limit | Non Funct. Req. – (Limit, Constraint) Keyword –(Funct. Req., Non Funct. Req.) | - |
| 3 | Environment Description | WD2-A6 | The goal of this fragment is to describe the environment of the system. | Requirement set | Enviroment Definition | Enviroment Analyst | Actor Funct. Req Non Funct. Req. Limit Constraint | Active Entities Passive Entities Environment Environment attribute | Passive Entity-System Active Entity-System Environment- Environment attribute | - |
| 4 | Use Case Definition | WD2-A7 | The purpose of this fragment is to clarify the different functionalities that the system must provide. | Environment Definition [with cooperation failures identified] | Environment Definition [complete], Use Case Diagrams | Enviroment Analyst | Active Entity Passive Entity Environment Environment attribute | Use Case Association UC_Relationship | Use Case – UseCase Text Description | Actor |
| 5 | UI Prototyping | WD2-A8,A9 | The goal of this fragment is to define and to test the Graphic User Interface (GUI) that allows the user to interact with the system. | Environment Definition | Prototypes UI | UI Designer, End User | Active Entity Passive Entity Environment Use Case | UI Txt Description UI Mockup | UI Txt Description-UI Mockup | - |
| 6 | Domain Analysis | WD3-A10 | The goal of this fragment is to give a static view and an abstraction of the real work established form the description of the system to be designed. | Requirement set, keyword set, environment definition | Software architecture [preliminary] | Object Analyst | Requirement Active Entity Passive Entity | Analysis Active Entity Analysis Passive Entity | Analysis Active Entity- Analysis Passive Entity | - |

7

| Title | WD-Activities | Description | Required Documents | Produced Documents | Actors | Input | Define | Relate | Quote |
|---------------------------------|---------------|---|---|--|-----------------|--|--|---|---|
| Verify AMAS adequacy | WD3-A11 | In this fragment, we verify that one (maybe more) Adaptive Multi-Agent System (AMAS) is needed to realize the system to be. We must study the adequacy at a global and a local level. | Software architecture [preliminary] | AMAS adequacy synthesis | Agent Analyst | Use Case Analysis Active Entity Environment Environment Attribute | Answer | Question-Answer | Question |
| Agents Identification | WD3-A12 | This fragment aims at finding what we will consider as agents in the system to be. | Software architecture [preliminary], AMAS Adequacy synthesis | Software Architecture [including agents] | Agent Analyst | Analysis Active Entity | Agent | - | Analysis Active Entity |
| Entities Interaction Study | WD3-A13 | This fragment aims at making clear interactions between the identified entities. | Environment definition [complete], software architecture [including agents] | Environment definition [final], software architecture [complete] | Domain Analyst | Agent Analysis Active Entity Analysis Passive Entity | Potential Cooperation Failure | Agent-(Agent, Analysis Active Entity, Analysis Passive Entity) | Agent Analysis Active Entity Analysis Passive Entity |
| Architecture Definition | WD4-A14 | The main objective of this fragment is to define the detailed architecture of the system in term of packages, classes, objects and agents. | Software architecture [complete] | Detailed architecture [initial] | Object Designer | Entities Agent | AdaptativeMultiAgentsSystem | - | - |
| Interaction language Definition | WD4-A15 | In this fragment we define the way the agents will interact | Detailed architecture [initial] | Interaction languages[initial] | Agent Designer | Agent | CooperativeInteractionProtocol Message | CooperativeInteractionProtocol-Message Message-Message Message-ActiveEntity | - |

8

9

10

11

12

| Title | WD-Activities | Description | Required Documents | Produced Documents | Actors | Input | Define | Relate | Quote |
|---|---------------|---|--|--|----------------|--|--|--|--|
| Agent Definition | WD4-A16 | This fragment, for every agent previously identified, aims to define its behavior: the skill, the aptitudes, an interaction language, a world representation. | Detailed architecture [initial], Interaction languages [initial] | Detailed architecture [intermediate], Interaction languages [intermediate] | Agent Designer | Agent CooperativeInteractionProtocol Message | Action Actuator Aptitude Characteristic CommunicationAction CommunicationPerception Perception Sensor Representation Skill AgentState Condition DecisionResult StandardRule | Action-Actuator Perception-Sensor Skill-Action Skill-Aptitude Rule- DecisionResult Rule-AgentStete | NonCooperativesSituationType |
| Non Cooperative Situations Identification | WD4-A16 | The objective of this fragment is to define the different Non Cooperative Situations (NCS) and the ways they are handled by the agents | Detailed architecture [intermediate], Interaction languages [intermediate] | Detailed architecture [draft], Interaction languages [draft] | Agent Designer | Action Actuator Aptitude Characteristic CommunicationAction CommunicationPerception Perception Sensor Representation Skill AgentState Condition DecisionResult StandardRule | Action Actuator Aptitude Characteristic CommunicationAction CommunicationPerception Perception Sensor Representation Skill AgentState Condition DecisionResult StandardRule Condition DecisionResult CooperativeRule | - | NonCooperativesSituationType Action Actuator Aptitude Characteristic CommunicationAction CommunicationPerception Perception Sensor Representation Skill AgentState Condition DecisionResult StandardRule |

13

14

| Title | WD-Activities | Description | Required Documents | Produced Documents | Actors | Input | Define | Relate | Quote |
|-------------------------|---------------|---|---|---|-----------------|--|----------------------------------|-----------------------------------|-------|
| Fast Prototyping | WD4-A17 | In this fragment, the identified behaviors of an agent are tested | Detailed architecture [draft], Interaction languages [draft] | Detailed architecture [including agent model], Interaction languages [final] | Agent Designer | Agent Action Actuator Aptitude Characteristic CommunicationAction CommunicationPerception Perception Sensor Representation Rule Skill AgentState Condition CooperativeRule DecisionResult StandardRule | - | - | - |
| Architecture Refinement | WD4-A18 | This fragment aims to complete the system architecture and the design activities. | Detailed architecture [including agent model] | Detailed architecture [final] | Object Designer | Action Actuator Aptitude Characteristic CommunicationAction CommunicationPerception Perception Sensor Representation Skill AgentState Condition DecisionResult StandardRule | StateExpression StateVariable | StateExpression- StateVariable | - |

15

16

| Title | WD-Activities | Description | Required Documents | Produced Documents | Actors | Input | Define | Relate | Quote |
|-----------------------------|-------------------|---|--|---|----------------------------|--|--|-----------------------|-------|
| Architecture Extraction | WD5-A19, A20 | In this fragment, a muADL model is generated from the AMAS-ML model | Detailed architecture [final], Interaction languages [final] | muADL Model | Agent Architecture Analyst | Action Actuator Aptitude Characteristic CommunicationAction CommunicationPerception Perception Sensor Representation Skill AgentState Condition DecisionResult StandardRule Condition DecisionResult CooperativeRule StateExpression StateVariable | MuArchitecture MuComponent Level LevelComponent ContainerLevel ApplicationLevel MethodReference MethodVisibility Element DataType Feature Method Model | MuComponent-Interface | - |
| Architecture Implementation | WD5 A21, A22, A23 | In this fragment, an API for the modelised agent type is generated. | muADL Model | Specific Cooperative Agent API (java library) | Java Developer | MuArchitecture MuComponent Level LevelComponent ContainerLevel ApplicationLevel MethodReference MethodVisibility Element DataType Feature Method Model | AMAS-ML model jar | - | - |
| Behaviour Implementation | WD5-A24, A25 | The goal of this fragment is to generate and implement the behavioural rules of the agents. | Specific Cooperative Agent API (java library) | AMAS Code | AMAS Developer | AMAS-ML model jar | Java class | - | - |

17

18

1 - Requirements Description (atomic) Fragment

Introduction

In this fragment, the user requirements for the to-be-designed system are determined consensually by the client, the end user and the requirement analyst.

The purpose of the Requirement Set is for one part to assure that the client and the designer of the system have a common vision of what will the system be, and for another part to ensure that the functionalities of the system will be in adequacy of the user's real needs.

This fragment is, for example, a part of the ADELFE process, where it is positioned in the beginning of the first Phase: Preliminary requirements (activities A1 to A3). Its purpose is to describe the to-be-designed system and its environment.

Fragment Description

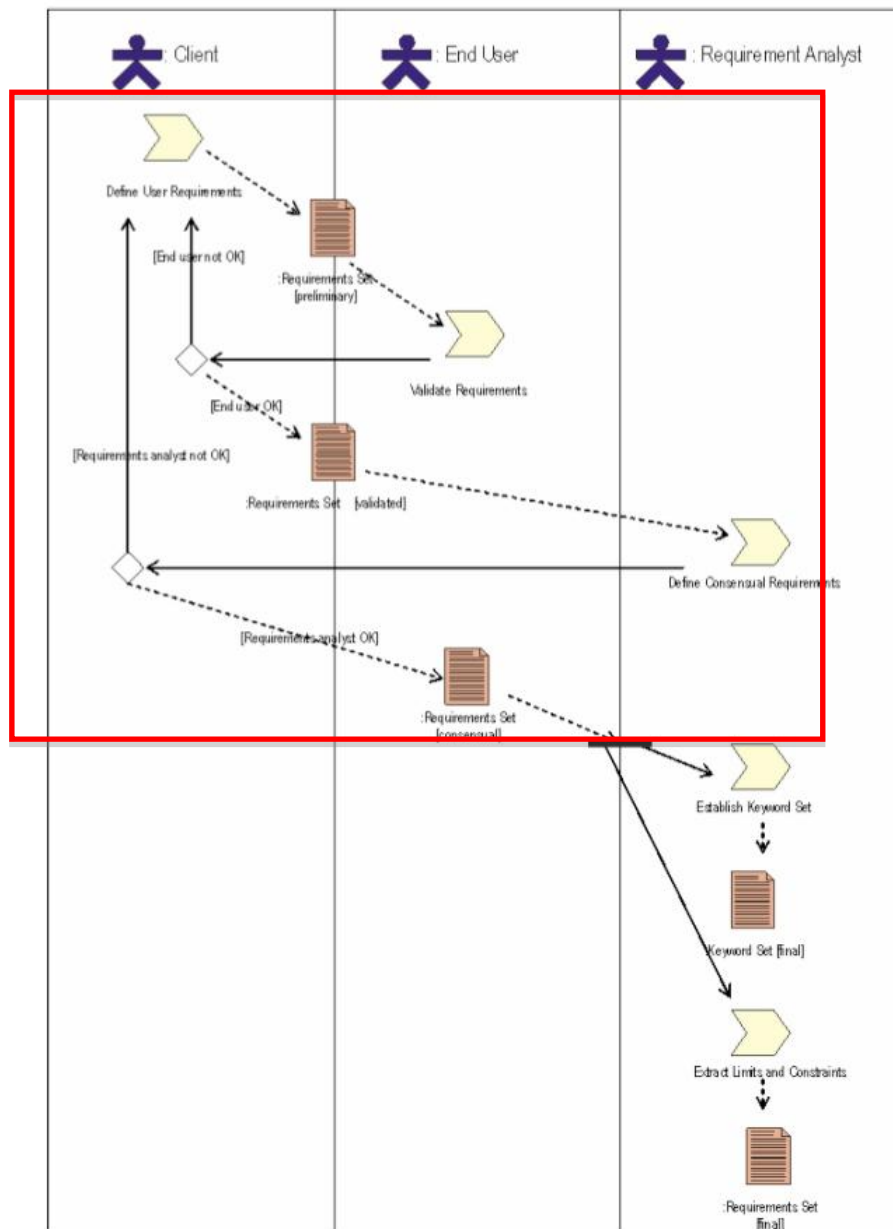
The goal of this fragment is to produce a list of (text) requirements for the system to be designed.

Input: none

Output: Requirement set [consensual]

Portion of Process workflow

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



Activities description:

| Activity | Activity Description | Roles involved |
|--------------------------|--|----------------|
| Define User Requirements | Describe the system and its environment. Define what is to be build or what will be the most adapted system for the end user | Client |
| Validate Requirements | Check and approve the | End User |

| | | |
|--------------------------------|---|---------------------|
| | content of the Requirement Set document. If the document is not approved, then it is necessary to go back to the “Define User Requirements” activity | |
| Define Consensual Requirements | Update and complete the Requirement Set document by including the consensual requirements. As for the precedent activity, if the document is not approved then it is necessary to go back to the “Define User Requirements” activity | Requirement Analyst |

Roles involved in this fragment are:

Client, End User, Requirement Analyst

They are described in the following subsections:

Client

Responsibilities of *Client* are:

- To list the potentials requirements
- To give the context in which the system will be deployed
- To establish the functional and non functional requirements

End User

Responsibilities of *End User* are:

- To check and approve the requirements listed by the client

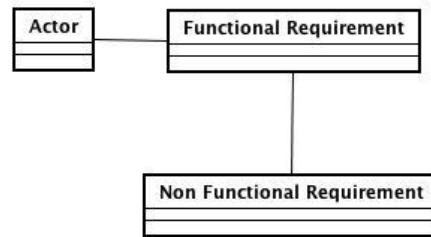
Requirement Analyst

Responsibilities of *Requirement Analyst* are:

- To update the requirement set with the consensual requirements.

Relationship with the MAS metamodel

The portion of metamodel affected by this fragment is:



This fragment refers to the MAS meta-model adopted in *ADELFE* and contributes to define and describe:

- Functional Requirements of the system to be designed
- Non Functional Requirements of the system to be designed
- Actor

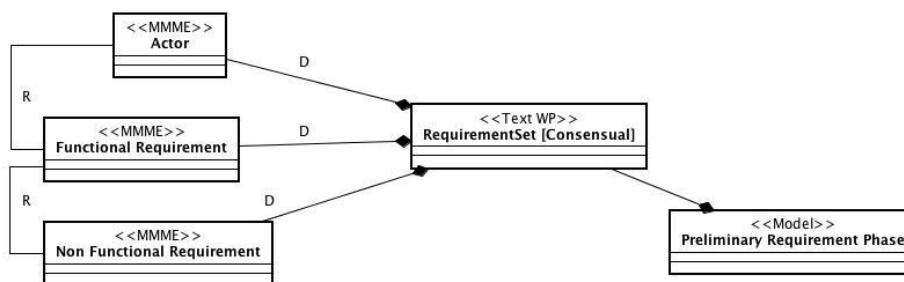
Deliverables

Requirement Set [Consensual]

The Requirement Set is a textual description of the to-be-designed system's basic specifications. Its purpose is to define the goals of the system and to give a formal description of the user requirements.

Deliverable relationships with the MMM

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



Preconditions and concepts to be defined

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

As regards documents:

| Input | Output |
|-------|--|
| | Requirements Description [consensual] |

As regards MAS metamodel elements:

| Input | To Be Designed | To be related | To be quoted |
|--|---|---|--------------|
| Actor Functional Requirement (<i>Text</i>) Non Functional Requirement (<i>Text</i>) [?][?][?] | Actor Functional Requirement Non Functional Requirement | Actor-Functional requirement Functional Requirement – non Functional Requirement | |

Guideline

End-users, clients, analysts and designers have to list the potential requirements. The context in which the system will be deployed must be understood. The functional and non-functional requirements must be established.

The **Requirements Set** document must then be checked, approved and updated with consensual requirements.

Composition Guideline

The requirement set is a classic method used to define in a formal way the system to be designed. The requirement set purpose is to ensure that the client's and the designer's views of the system to be designed are similar enough. So in most approaches, this fragment is intended to be the first of the design process but it can also be preceded by a requirements elicitation fragment

Aspects of fragment

None

Dependency Relationships with other fragments

.None

Glossary

This Fragment refers these terms:

Consensual requirement: a condition or functionality the system must conform to, and upon what the client, end user and the requirement analyst agree.

Functional requirement: a requirement that specify a system function, as calculations, technical details, data manipulation and processing.

Non functional requirement: a requirement that specify the system properties, as the environmental or implementation constraints or constraints of performance, of dependency to a platform, of maintenance, of extensibility and of security.

2 - Finalize Requirements (atomic) Fragment

Introduction

In this fragment, keywords from requirements set should be extracted and limits and constraints of the system should be defined.

In this fragment, the requirement set is completed to take in account the limits and constraints of the system to be designed. It allows a more precise definition of these restrictions of the system, on which the client and designer can agree. The definition of a keyword list disambiguate the terms used in the Requirement Set.

This fragment is for instance present in the ADELFE process in the first Phase: Preliminary requirements (activities A4 and A5).

Fragment Description

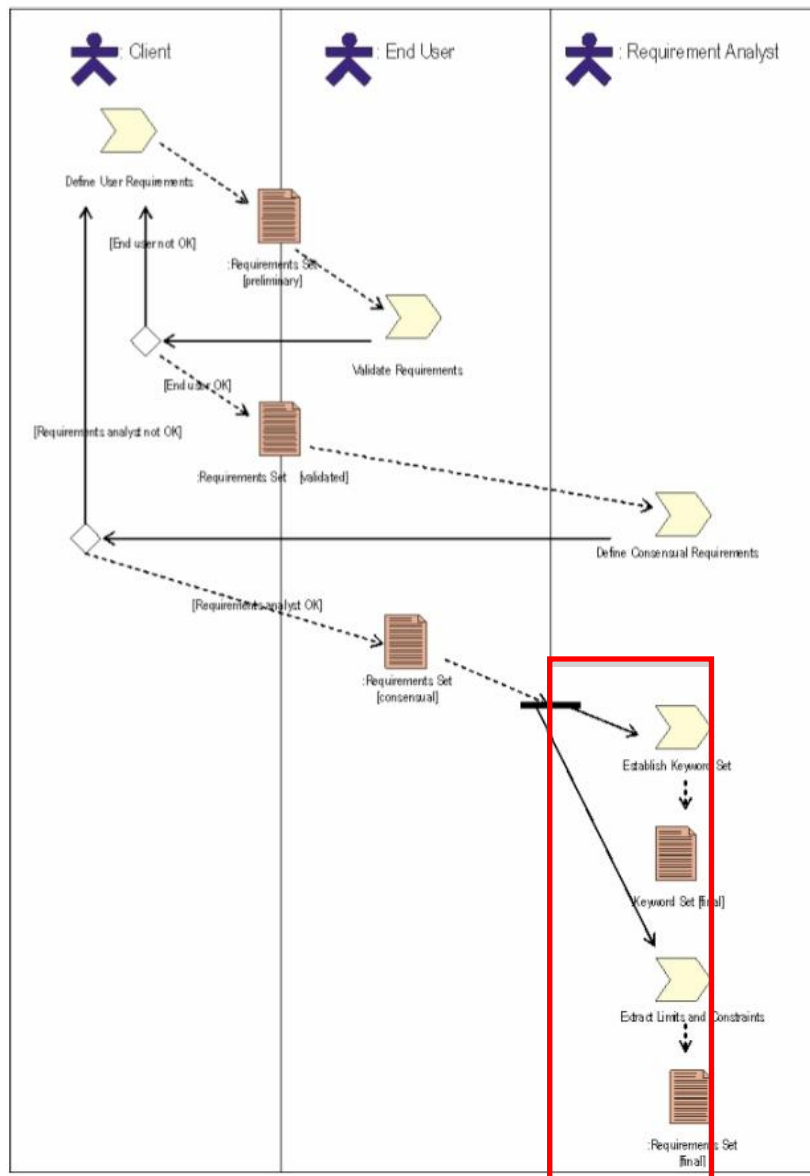
The goal of this fragment is to produce a finalized list of (text) requirements for the system to be designed and a list of keywords to define the concepts used to describe the application.

Input: Requirement Set [consensual]

Output: Requirement Set [final], keywords set

Portion of Process workflow

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



Activities description:

| Activity | Activity Description | Roles involved |
|--------------------------------|---|---------------------|
| Establish a glossary | List the primary concepts used to describe the application and its domain (system and environment). | Requirement Analyst |
| Extract limits and constraints | Define limits and constraints of the system to build. They can be | Requirement Analyst |

| | | |
|--|--|--|
| | deduced of the non functional requirements expression and context definition | |
|--|--|--|

Roles involved in this fragment are:

Requirement Analyst

They are described in the following subsections:

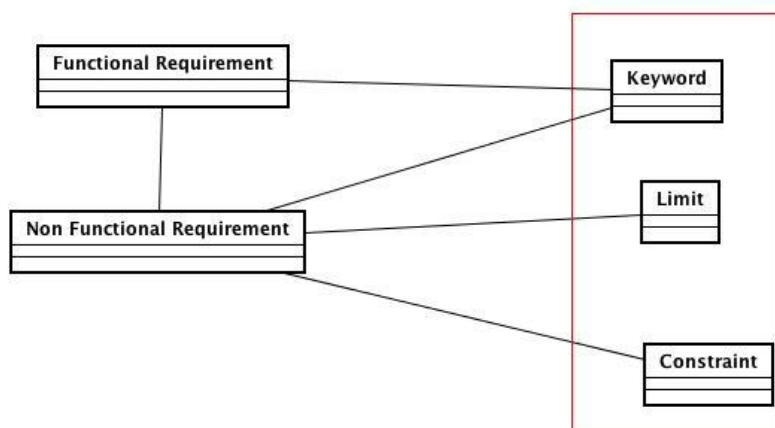
Requirement Analyst

Responsibilities of *Requirement Analyst* are:

- To establish a keyword set
- To extract the limits and constraints of the system to be designed

Relationship with the MAS metamodel

The portion of metamodel affected by this fragment is:



This fragment refers to the MAS meta-model adopted in *ADELFE* and contributes to define and describe:

- Keywords used in the Requirement Set
- Limits of the system to be designed
- Constraints of the system to be designed

Deliverables

Requirement Set [FINAL]

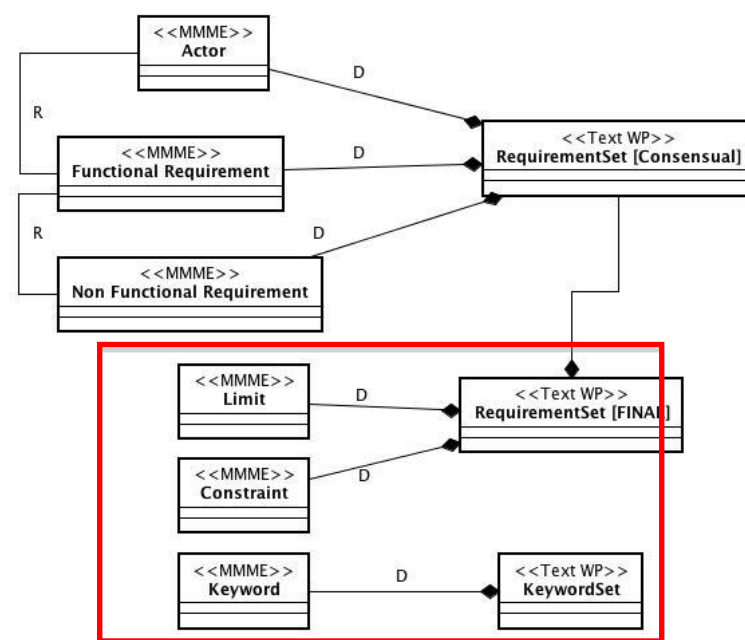
The Requirement Set is a textual description of the to-be-designed system's basic specifications. Its purpose is to define the goals of the system and to give a formal description of the user requirements.

Keywords Set

This document contains a definition of each primary concept used to describe the application and its domain

Deliverable relationships with the MMM

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



Preconditions and concepts to be defined

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

As regards documents:

| Input | Output |
|---------------------------------|-------------------------|
| Requirement Set [Consensual] | Keyword Set |
| | Requirement Set [Final] |

As regards MAS metamodel elements:

| Input | To Be Designed | To be related | To be quoted |
|-----------------|----------------|---|--------------|
| Actor | Keyword | Non Funct. Req. – Limit, Constraint | |
| Funct. Req | Constraint | Keyword – Funct. Req., Non Funct. Req. | |
| Non Funct. Req. | Limit | | |

Guideline

Limits and constraints can be found in the expression of non functional requirements and in the definition of the context in which the system will be deployed.

You have to list the main concepts used to describe the application and its domain (the system and its environment).

Composition Guideline

None

Aspects of fragment

None

Dependency Relationships with other fragments

As this fragment needs a consensual requirement set, it should be preceded by a fragment which can product such a document, for example the Requirements Description fragment.

Glossary

This Fragment refers these terms:

Functional requirement: a requirement that specify a system function, as calculations, technical details, data manipulation and processing.

Non functional requirement: a requirement that specify a system property, as the environmental or implementation constraints, constraints of performance, of dependency to a platform, of maintenance, of extensibility and of security.

Environment description (atomic) fragment

Introduction

The principal objective of this fragment is to characterise the environment of the system to be designed. This definition will allow an easier definition of the use cases and constraints. These constraints and rules can make appear some environmental problems like non determination or discontinuity. It is important to start looking for this kind of characteristics as soon as possible in the process. This fragment is present in the ADELFE process on the second Phase: Final requirements (Activity A6)

Fragment Description

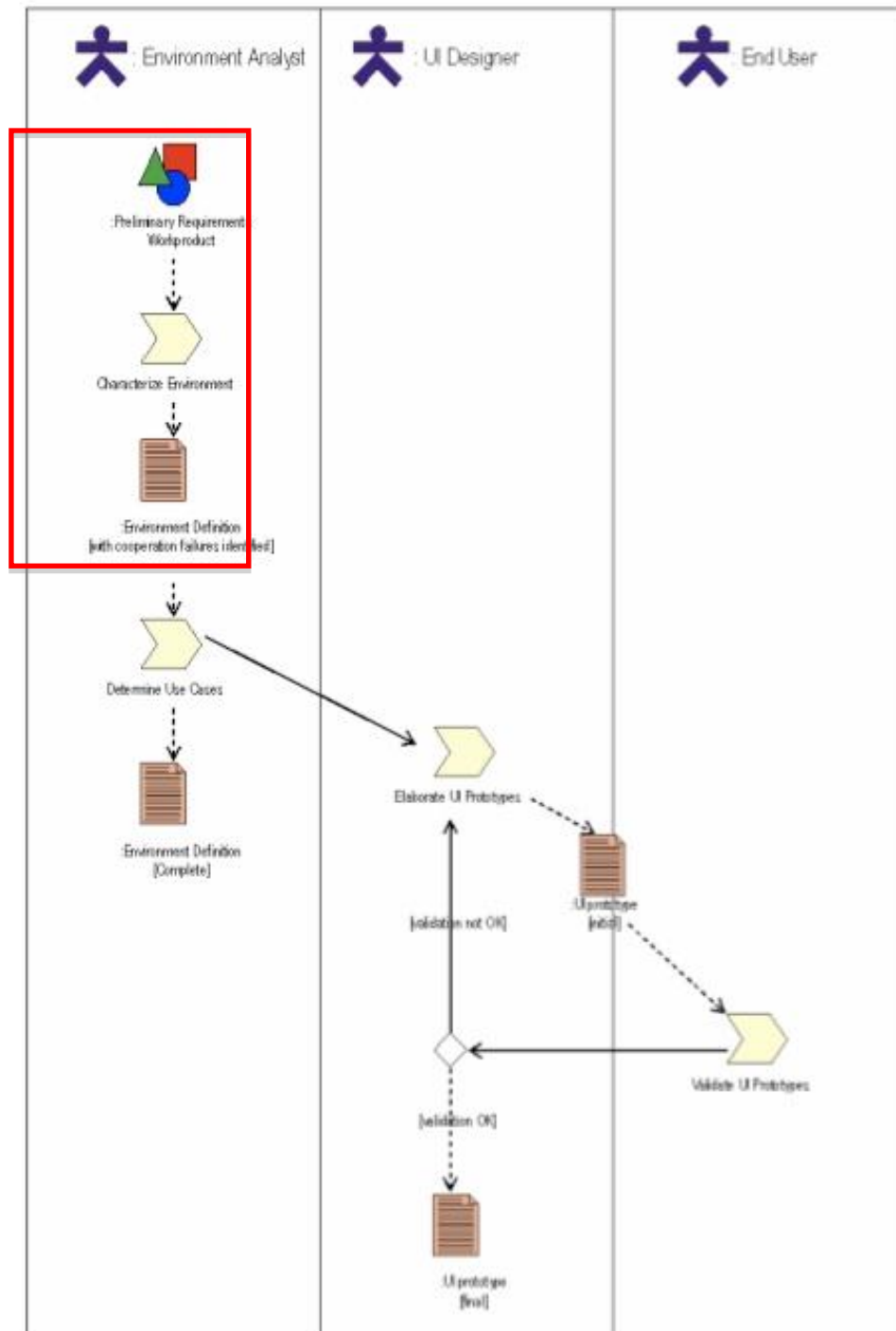
The goal of this fragment is to describe the environment of the system.

Input: Preliminary requirement workproduct.

Output: Environment definition [with cooperation failures identified]

Portion of Process workflow

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



Activities description:

| Activity | Activity Description | Roles involved |
|-------------------|--|---------------------|
| Identify entities | Identify actives and passives entities in interaction with the | Environment Analyst |

| | | |
|--------------------------|---|---------------------|
| | system, as well as constraints on these interactions. | |
| Define context | Characterise the data flux and interactions between identified entities and the system. | Environment Analyst |
| Characterize Environment | Characterize the environment using the terms proposed by Russel and Norvig. [Be more precise?] [?] | Environment Analyst |

Roles involved in this fragment are:

Environment Analyst

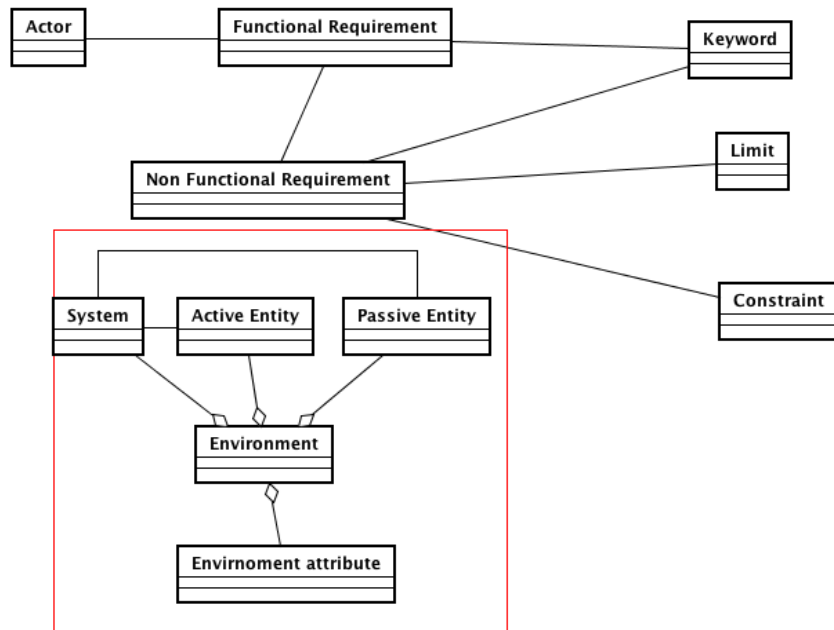
They are described in the following subsections:

Environment Analyst

Responsibilities of *Environment Analyst* are:

- To determine the entities in interaction with the system
- To define the context of these interactions
- To characterise the environment using terms defined by Russel and Norvig [?]

The portion of metamodel affected by this fragment is:



Report the metamodel and show the elements concerned with this fragment

This fragment refers to the MAS meta-model adopted in *ADELFE* and contributes to define and describe:

- the entities (passive or active) in interaction with the system
- the context of these interactions
- the environment of the system

Environment definition [with cooperation failures identified]

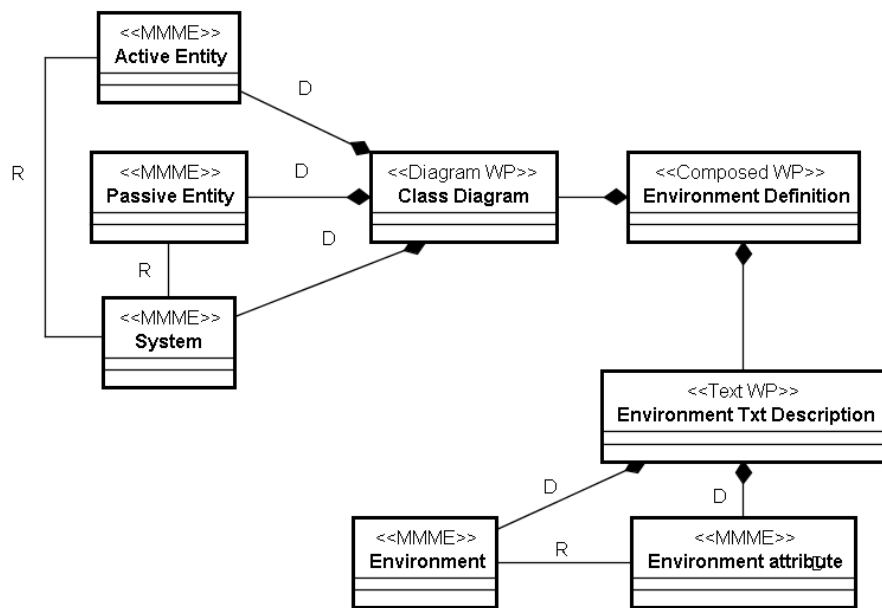
Report an example in a picture

Describe what type of diagram it is (class diagram, UML, Tropos ...)

[?] Sequence/Collaboration Diagrams to describe the context [?]

Deliverable relationships with the MMM

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



Preconditions and concepts to be defined

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

As regards documents:

| Input | Output |
|-----------------|------------------------|
| Requirement Set | Environment Definition |

As regards MAS metamodel elements:

| Input | To Be Designed | To be related | To be quoted |
|-----------------|-----------------------|-----------------------|--------------|
| Actor | Active Entities | Passive Entity-System | |
| Funct. Req | Passive Entities | Active Entity-System | |
| Non Funct. Req. | Environment | Environment - | |
| Limit | Environment attribute | Environment attribute | |

| | | | |
|------------|--|--|--|
| Constraint | | | |
|------------|--|--|--|

Guideline

This fragment is characterized by three phases: to determine entities, to define context and to characterize environment.

The environment can be characterized by these terms:

- Accessible or not
- Deterministic or not
- Static or dynamic
- Discrete or continuous

Composition Guideline

None

Aspects of fragment

None

Dependency Relationships with other fragments

One goal of this fragment is to make easier the definition of the use cases, so it is well placed when just before a use cases definition fragment.

To use this fragment, you need to have a clear vision of the system to be designed, so this fragment should be placed after a fragment or a composition of fragments that give a formal description of the system.

Glossary

This Fragment refers these terms:

Active Entity: An *entity* that can behave autonomously; for instance, by bringing modifications to their functioning constraints. They are able to act in a dynamical way with the system. Agents composing the system will be found among them

Entity: An actor in the UML sense; It is a set of coherent roles which the users of use cases play when they interact with use cases. In this fragment we make the distinction between *active entities* and *passive entities*.

Environment:

Passive Entity: An *entity* that can be considered as a resource by the system. Interactions with the system are restricted to data exchanges in order to realise the task the system must achieve. *Passive entities* may be used or modified by *active entities* but they do not change by themselves in an autonomous manner.

Dynamic: Teachers, Students groups, Rooms manager and Courses manager are unpredictable. They can add or modify constraints at any time.

Accessible: All the rooms are described

4 - Use Case Definition Fragment

Introduction

The main objective of this fragment is to clarify the different functionalities that the studied system must provide. Lately, these functionalities will be clustered in one or several use case diagrams between the entities and the system.

This fragment is used in the ADELFE process as the Activity 6 of the second Phase: Final Requirements.

Fragment Description

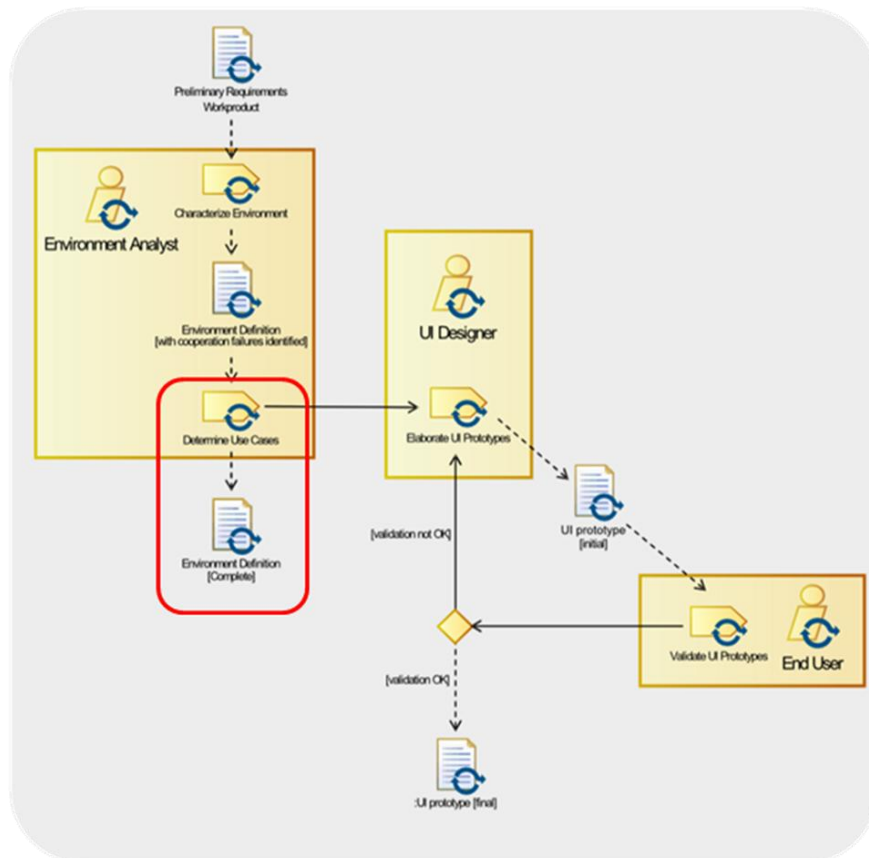
The purpose of this fragment is to clarify the different functionalities that the system must provide.

Input: Environment Definition [with cooperation failures identified]

Output: Environment Definition [complete]

Portion of Process workflow

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



Activities description:

| Activity | Activity Description | Roles involved |
|----------|----------------------|----------------|
|----------|----------------------|----------------|

| | | |
|-------------------------------|--|---------------------|
| List use cases | Identify the different use cases that can exist for the system. Build the corresponding use case diagrams. | Environment Analyst |
| Identify cooperation failures | Determine events that can lead to situations not totally controlled by the system developer and potentially harmful. | Environment Analyst |
| Establish Sequence Diagrams | For each use case, make a corresponding sequence diagram. | Environment Analyst |

Roles involved in this fragment are:

Environment Analyst

They are described in the following subsections:

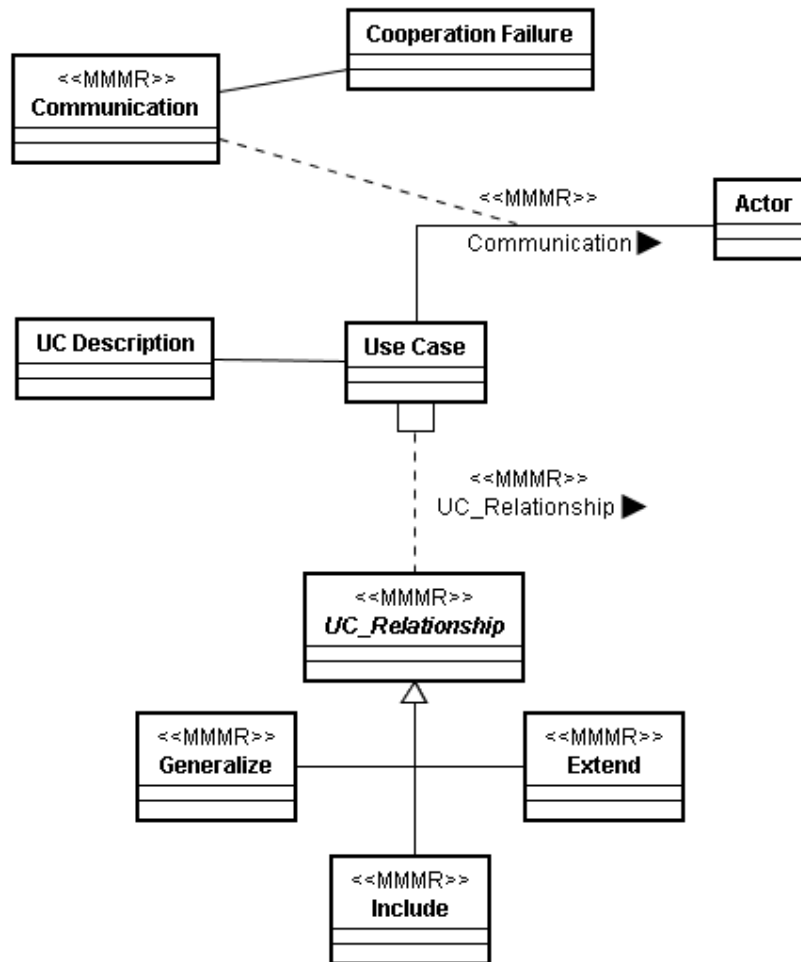
Environment Analyst

Responsibilities of *Environment Analyst* are:

- List the use cases
- For each use case, make a use case diagram
- Identify cooperation failures
- For each use case, make a corresponding sequence diagram

Relationship with the MAS metamodel

The portion of metamodel affected by this fragment is:



This fragment refers to the MAS meta-model adopted in *ADELFE* and contributes to define and describe:

- The different use cases ;
- The cooperation failures.

Deliverables

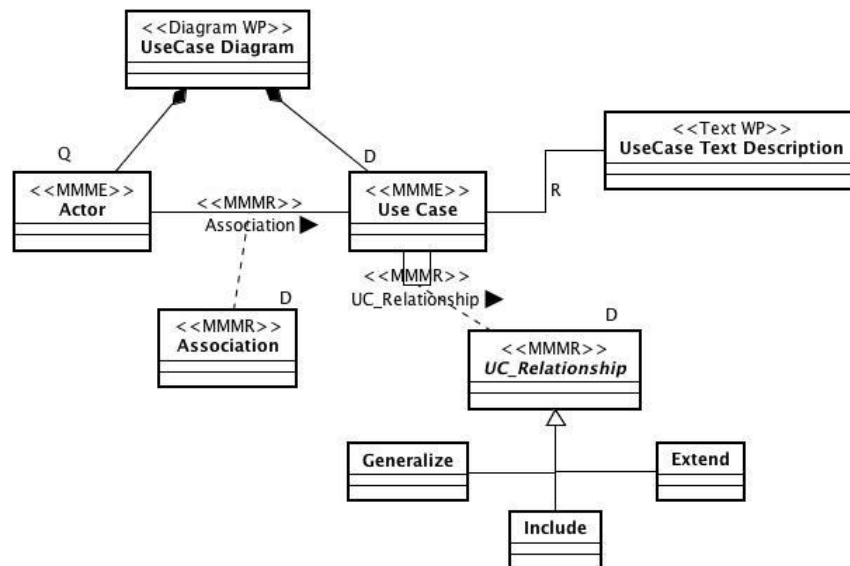
Environment Description [Complete]

The functionalities of the system are represented through use case diagrams in the Functional Description Model. This information completes the Environment Definition document.

A use case is detailed using a textual description and specific sequence diagrams. To manage a possible exception can be inserted a special box in the use case.

Deliverable relationships with the MMM

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



Preconditions and concepts to be defined

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

As regards documents:

| Input | Output |
|------------------------|--|
| Environment Definition | Use Case diagram Environment Definition [complete] |

As regards MAS metamodel elements:

| Input | To Be Designed | To be related | To be quoted |
|--|--|--|--------------|
| Active Entities Passive Entities Environment Environment attribute | Use Case Association UC_Relationship | Use Case – UseCase Text Description | Actor |

Guideline

The functionalities of the system are represented through use case diagrams in the Functional Description Model. This information completes the Environment Definition document.

A use case is detailed using a textual description and specific sequence diagrams. To manage a possible exception can be inserted a special box in the use case.

Composition Guideline

None

Aspects of fragment

Use Cases are expressed using UML diagrams.

Dependency Relationships with other fragments

This fragment depends of an Environment Description fragment since the use cases represent the system functionalities defined by studying the environment of the system.

5 - UI Prototyping

Introduction

Insert here a text description of the design process and the entire SPEM diagram that can help in positioning the fragment in the process workflow

Fragment Description

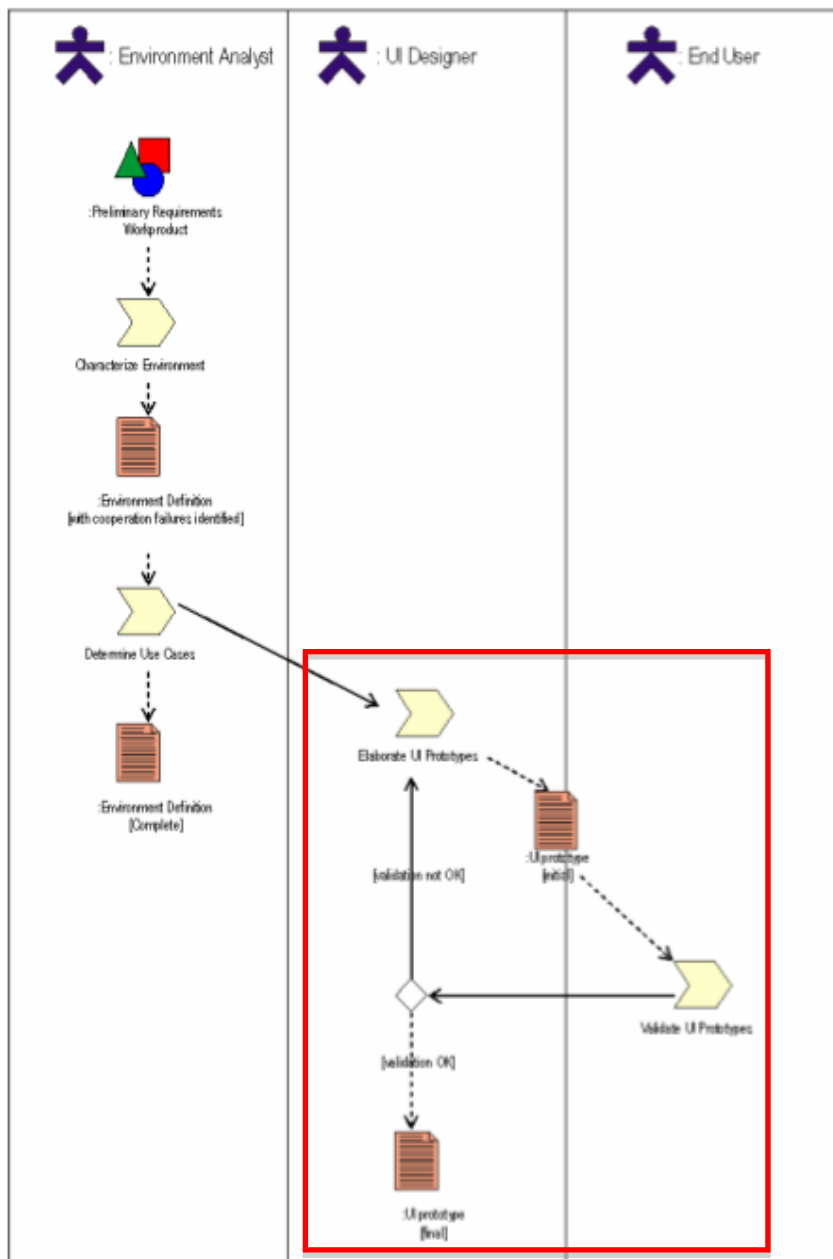
The goal of this fragment is to define and to test the Graphic User Interface (GUI) that allows the user to interact with the system.

Input: [?]

Output: UI Prototypes

Portion of Process workflow

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



Activities description:

| Activity | Activity Description | Roles involved |
|-------------------------|---|----------------|
| Elaborate UI Prototypes | Specify the Graphic User Interfaces (GUI) that will allow the user to interact with the system and define the relations between these interfaces. | UI Designer |
| Validate UI Prototypes | Study and judge the GUI | End User |

| | | |
|--|--|--|
| | defined in the last activity from a functional and a non functional point of view. If the validation fail, it is necessary to redo the Elaborate UI Prototypes activity. | |
|--|--|--|

Roles involved in this fragment are:

- UI Designer
- End User

They are described in the following subsections:

UI Designer

Responsibilities of *UI Designer* are:

- Elaborate UI Prototypes

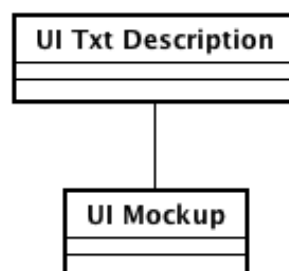
End User

Responsibilities of *End User* are:

- Validate UI Prototypes

Relationship with the MAS metamodel

The portion of metamodel of this fragment is:



This fragment refers to the MAS meta-model adopted in *ADELFE* and contributes to define and describe ...<what?List of elements> [?]

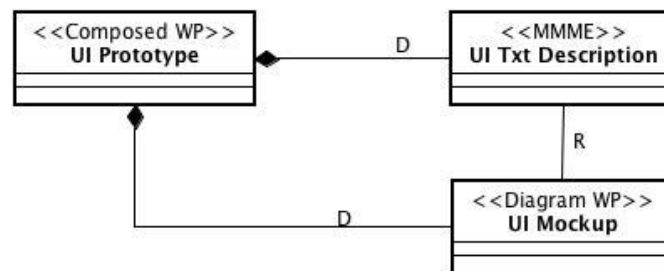
Deliverables

Prototypes UI

The GUIs must be described in the UI Prototype (final) document and on the Interface Models represented through UML diagrams.

Deliverable relationships with the MMM

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



Preconditions and concepts to be defined

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

As regards documents:

| Input | Output |
|-------|---------------|
| [?] | UI Prototypes |

As regards MAS metamodel elements:

| Input | To Be Designed | To be related | To be quoted |
|-------|---------------------------------|---------------------------------|--------------|
| [?] | UI Txt Description UI Mockup | UI Txt Description-UI Mockup | |

Guideline

Describe guidelines for performing the activities described in the fragment and correctly achieve the goal

Composition Guideline

None

Aspects of fragment

A possible means to describe the GUIs is to use the basic tool provided by OpenTool

Dependency Relationships with other fragments

To apply this fragment, you have to know the different use cases of the system, since the UI must be defined knowing the system functionalities. So this fragment must be preceded by a use cases fragment.

Glossary

This Fragment refers these terms:

6 - Domain Analysis

Introduction

Insert here a text description of the design process and the entire SPEM diagram that can help in positioning the fragment in the process workflow

Fragment Description

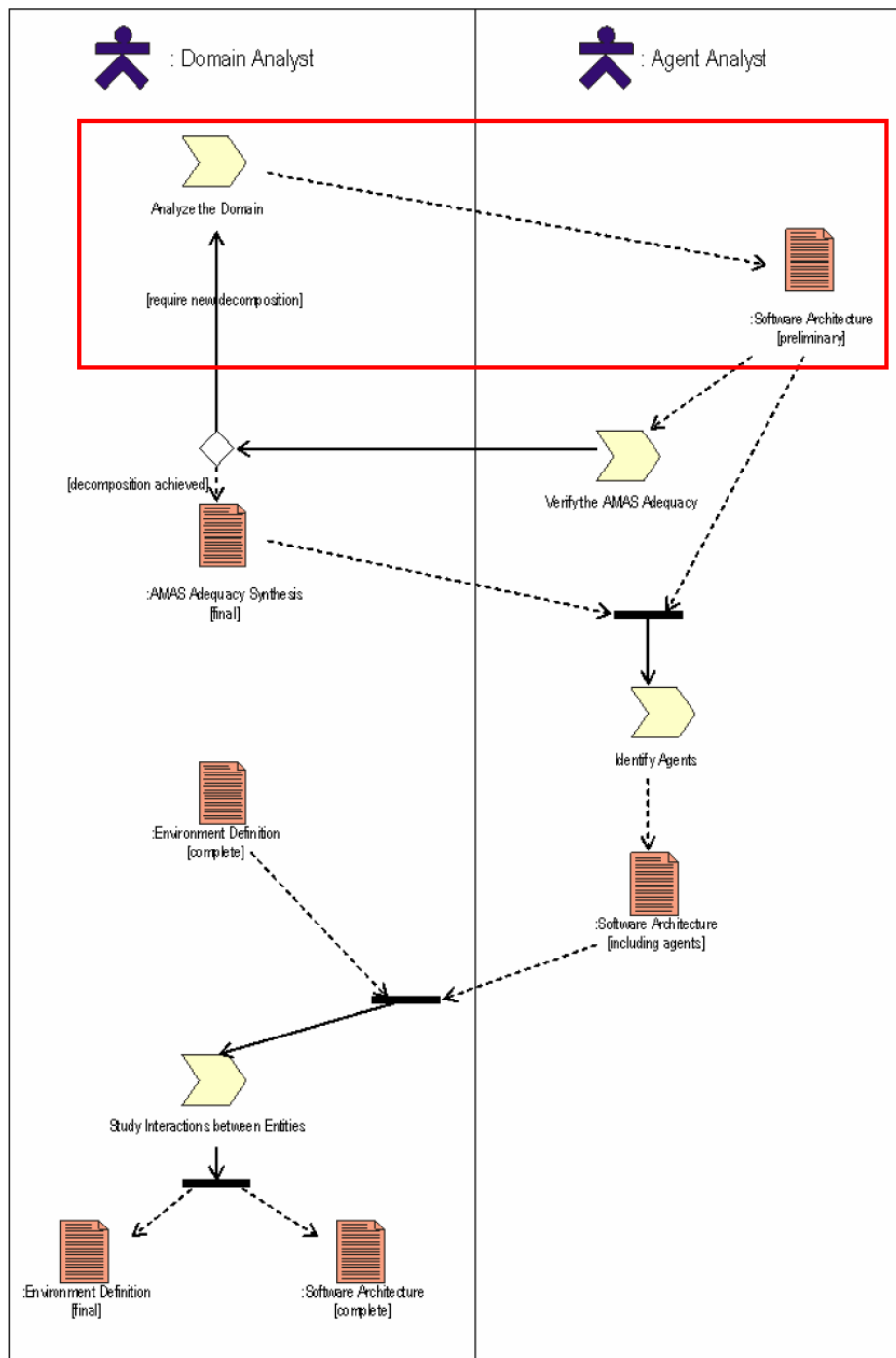
The goal of this fragment is to give a static view and an abstraction of the real work established from the description of the system to be designed.

Input: Requirement set, keyword set, environment definition

Output: Software architecture [preliminary]

Portion of Process workflow

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



Activities description:

| Activity | Activity Description | Roles involved |
|------------------|--|----------------|
| Identify classes | The set of use cases, the corresponding sequence diagrams and the keywords set must be | Object Analyst |

| | | |
|-----------------------------------|--|----------------|
| | analysed to identify needed classes. When identifying these classes you may have to update the already given entities list | |
| Study relationships among classes | Study the interactions between the different classes by studying the use cases and sequence diagrams. | Object Analyst |
| Build preliminary class diagram | Once the different classes and their interactions are identified, construct the preliminary class diagram. | Object Analyst |

Roles involved in this fragment are:

Object Analyst

They are described in the following subsections:

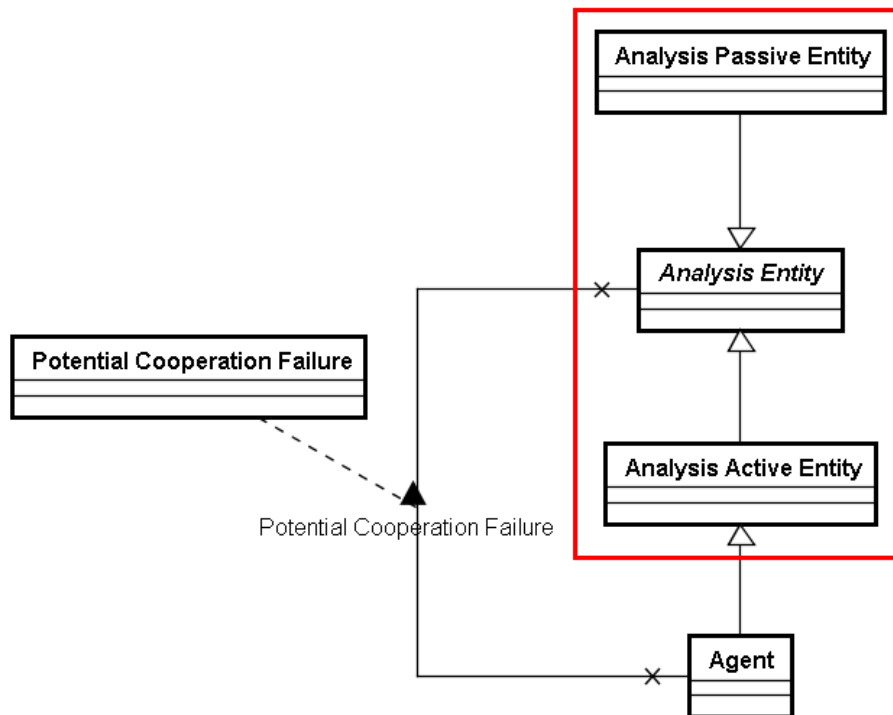
Object Analyst

Responsibilities of *Object Analyst* are:

- Identify classes
- Study relationship among classes
- Build preliminary class diagram

Relationship with the MAS metamodel

The portion of metamodel of this fragment is:



This fragment refers to the MAS meta-model adopted in *ADELFE* and contributes to define and describe:

- Analysis Passive Entity
- Analysis Active Entity

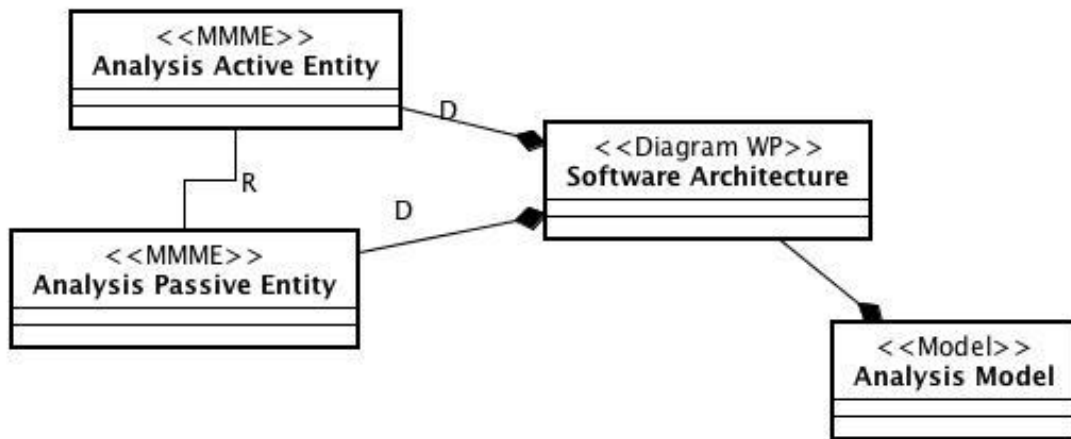
Deliverables

Software Architecture [preliminary]

The output will be a set of entities that will compose a preliminary class diagram (Domain Model) through UML notation and a Software Architecture (preliminary) document.

Deliverable relationships with the MMM

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



Preconditions and concepts to be defined

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

As regards documents:

| Input | Output |
|------------------------|---|
| Requirement Set | Software Architecture class diagram [preliminary] |
| Keyword Set | |
| Environment Definition | |

As regards MAS metamodel elements:

| Input | To Be Designed | To be related | To be quoted |
|--|---|--|--------------|
| Requirement Active Entity Passive Entity | Analysis Active Entity Analysis Passive Entity | Analysis Active Entity- Analysis Passive Entity | |

Guideline

Describe guidelines for performing the activities described in the fragment and correctly achieve the goal

Composition Guideline

None

Aspects of fragment

None

Dependency Relationships with other fragments

This fragment depends of two fragments, Requirement Description and Keywords

Identification since the domain is described using the Requirements Set and the Keywords Set documents.

Glossary

This Fragment refers these terms:

7 - Verify AMAS Adequacy

Introduction

Insert here a text description of the design process and the entire SPEM diagram that can help in positioning the fragment in the process workflow

Fragment Description

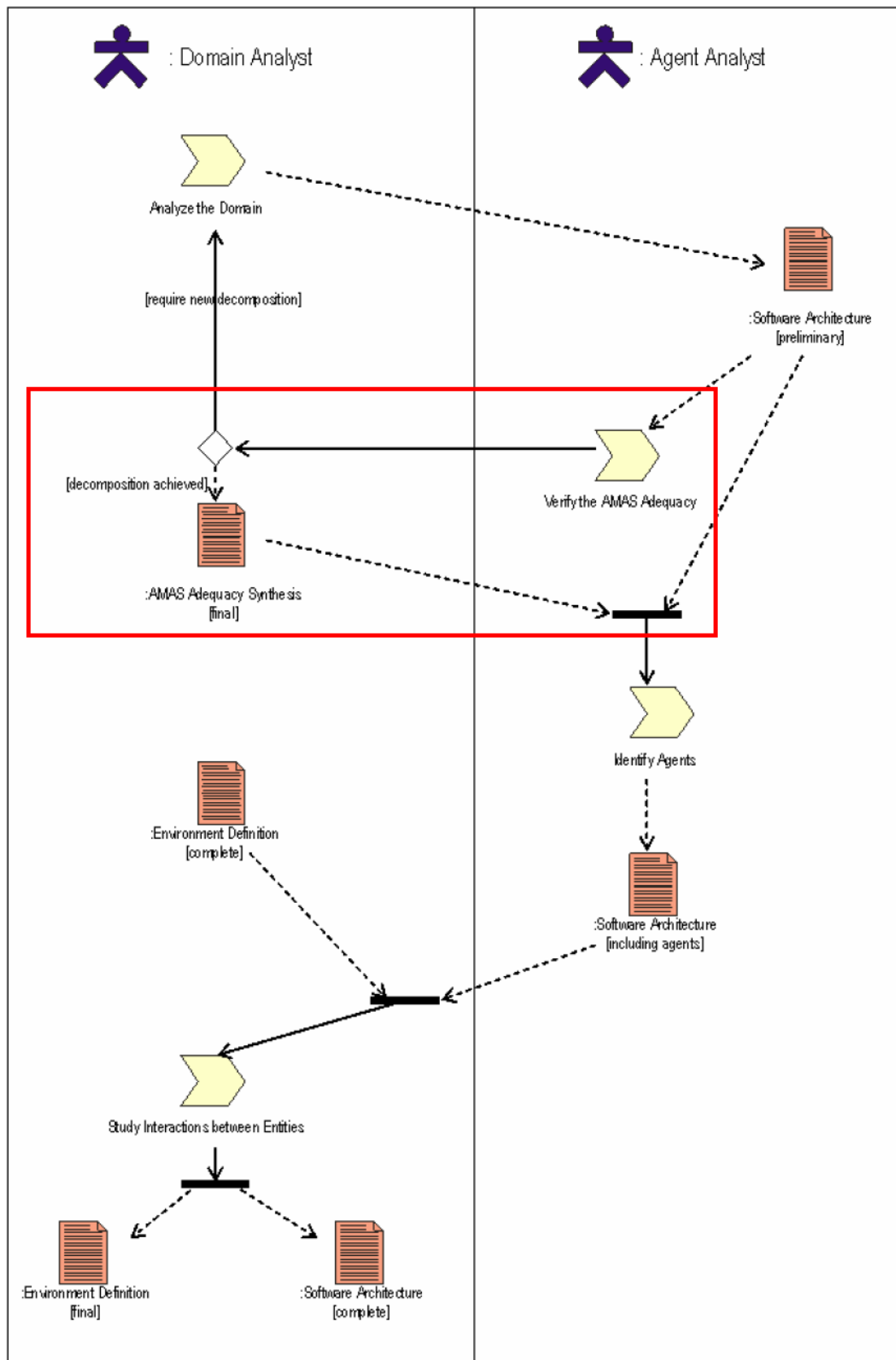
In this fragment, we verify that one (maybe more) Adaptive Multi-Agent System (AMAS) is needed to realize the system to be. We must study the adequacy at a global and a local level.

Input: Software architecture [preliminary]

Output: AMAS adequacy synthesis

Portion of Process workflow

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



Activities description:

| Activity | Activity Description | Roles involved |
|------------------------------|--|----------------|
| Compute Global AMAS Adequacy | At the global level answer to the question "is an AMAS required to implement the system?" | Agent Analyst |
| Compute Local AMAS Adequacy | At the local level try to determine if some agents are needed to be implemented like AMAS i.e. if a certain kind of decomposition or recursion is required during the building of your system. | Agent Analyst |

Roles involved in this fragment are:

Agent Analyst

They are described in the following subsections:

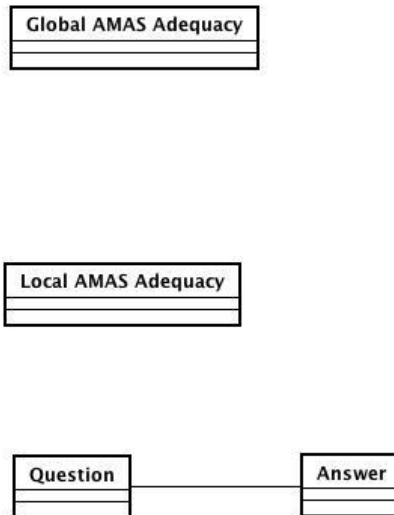
Agent Analyst

Responsibilities of *Agent Analyst* are:

- Compute Global AMAS adequacy
- Compute Local AMAS adequacy

Relationship with the MAS metamodel

The portion of metamodel of this fragment is:



This fragment refers to the MAS meta-model adopted in *ADELFE* and contributes to define and describe:

- Global AMAS Adequacy
- Local AMAS Adequacy
- Answer

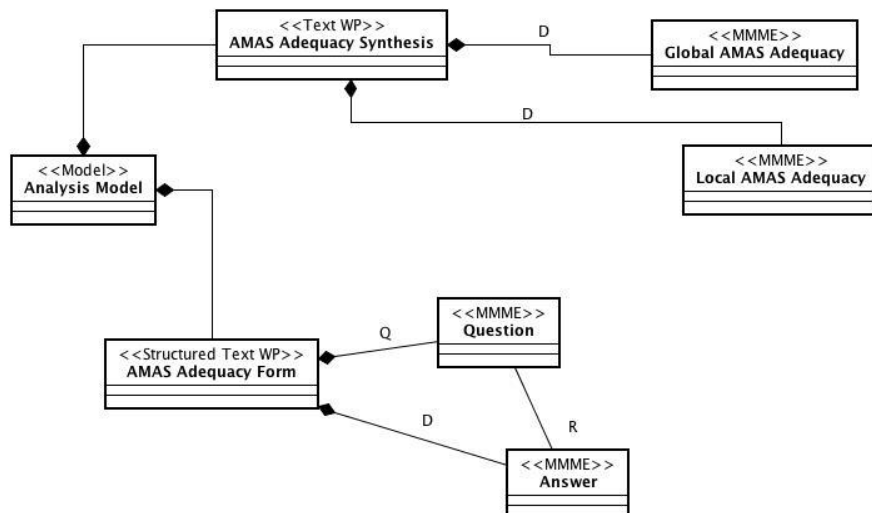
Deliverables

AMAS Adequacy Synthesis

The document in which the conclusions will be written on this verification

Deliverable relationships with the MMM

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



Preconditions and concepts to be defined

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

As regards documents:

| Input | Output |
|--|-------------------------|
| Software Architecture [Preliminary] Environment Definition [?] | AMAS Adequacy Synthesis |

As regards MAS metamodel elements:

| Input | To Be Designed | To be related | To be quoted |
|-------|----------------|---------------|--------------|
|-------|----------------|---------------|--------------|

| | | | |
|------------------------|----------------------|-----------------|----------|
| Use Case | | | |
| Analysis Active Entity | Answer | | |
| Environment | Global AMAS Adequacy | Question-Answer | Question |
| Environment Attribute | Local AMAS Adequacy | | |

Guideline

It is possible to remake the description of the domain if decomposition has been detected, in order to precise the domain analysis and obtain a sufficient enough decomposition.

Composition Guideline

None

Aspects of fragment

The AMAS adequacy graphical tool of the ADELFE method can be used for the adequacy verification since it helps to answer to the questions on the global and local levels.

Dependency Relationships with other fragments

This fragment depends of the Domain Description fragment since the AMAS Adequacy is verified using the Software Architecture (preliminary) document.

8 - Agents Identification

Introduction

Insert here a text description of the design process and all the SPEM diagram that can help in positioning the fragment in the process workflow

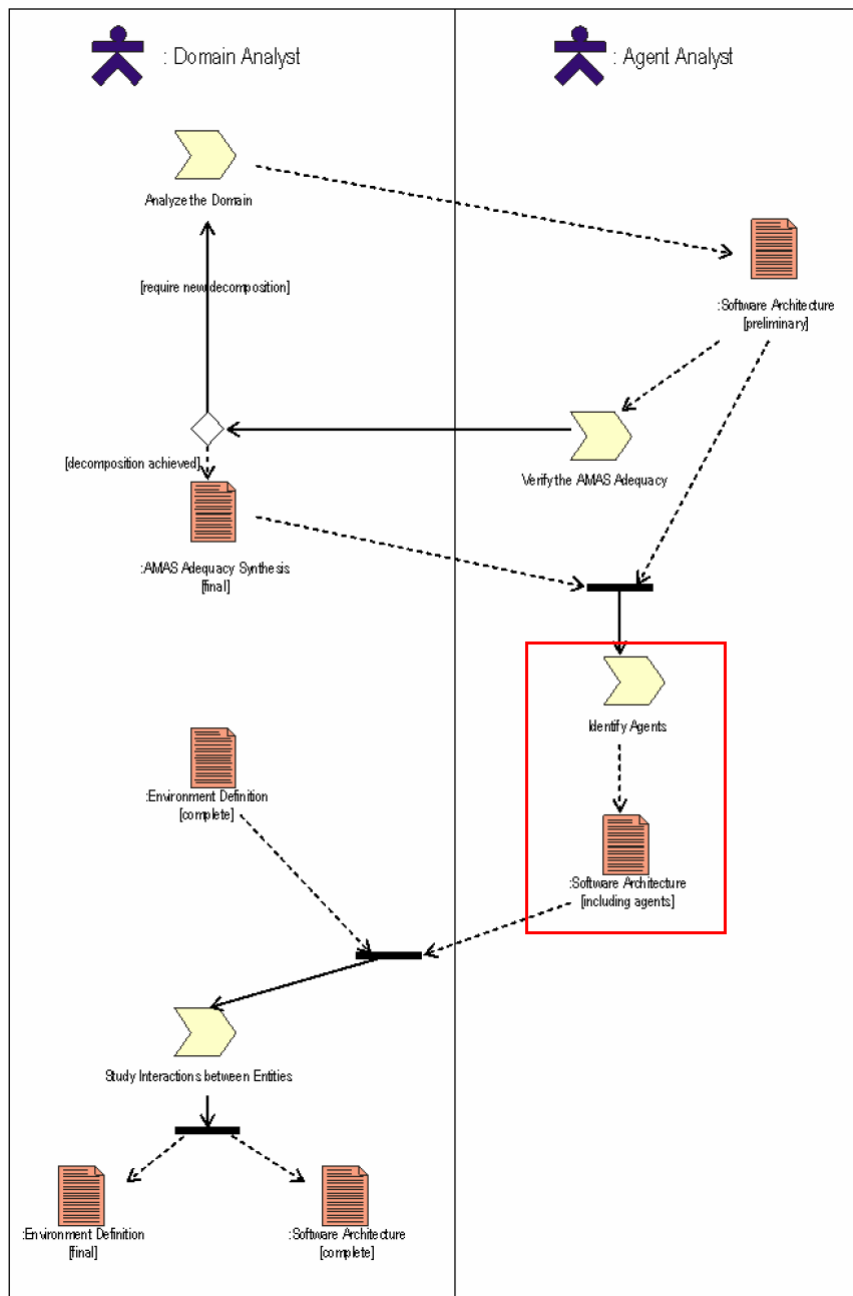
Fragment Description

This fragment aims at finding what we will consider as agents in the system to be.

In this fragment, we are only interested in agents which enable a designer to build AMAS. These agents are looked for among the identified entities in interaction with the system and previously defined classes.

Portion of Process workflow

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



Activities description:

| Activity | Activity Description | Roles involved |
|--------------------------------|---|----------------|
| Study existing active entities | For each defined entity, you have to decide if it : -is autonomous -has a local goal to pursue | Agent Analyst |

| | | |
|---|--|---------------|
| | <p>-has to interact with some other entities</p> <p>And if it:</p> <p>-has a partial view of the environment</p> <p>-has some abilities of negotiation</p> <p>Entities verifying all the three criteria may be viewed as agents, additional characteristics will be studied in the next activity</p> | |
| Identify potentially cooperative entities | <p>For each entity coming from the previous activity, you have to determine if it:</p> <p>-has to move in a dynamic environment</p> <p>- has to face up to cooperation failures</p> <p>-has to treat Non Cooperative Situations</p> <p>Entities that are verifying at least the last criterion must be marked during the next activity</p> | Agent Analyst |
| Define Agents | <p>The entities coming from the previous activity can be now considered as agent. So, their class has now to be refereed as a</p> | Agent Analyst |

| | | |
|--|-------------------|--|
| | Cooperative Agent | |
|--|-------------------|--|

Roles involved in this fragment are:

Agent Analyst

They are described in the following subsections:

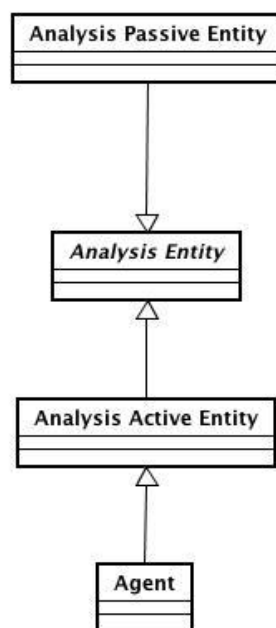
Agent Analyst

Responsibilities of *Agent Analyst* are:

- Study existing active entities
- Identify potentially cooperative entities
- Define Agents

Relationship with the MAS metamodel

The portion of metamodel of this fragment is:



This fragment refers to the MAS meta-model adopted in *ADELFE* and contributes to define and describe:

-Agents

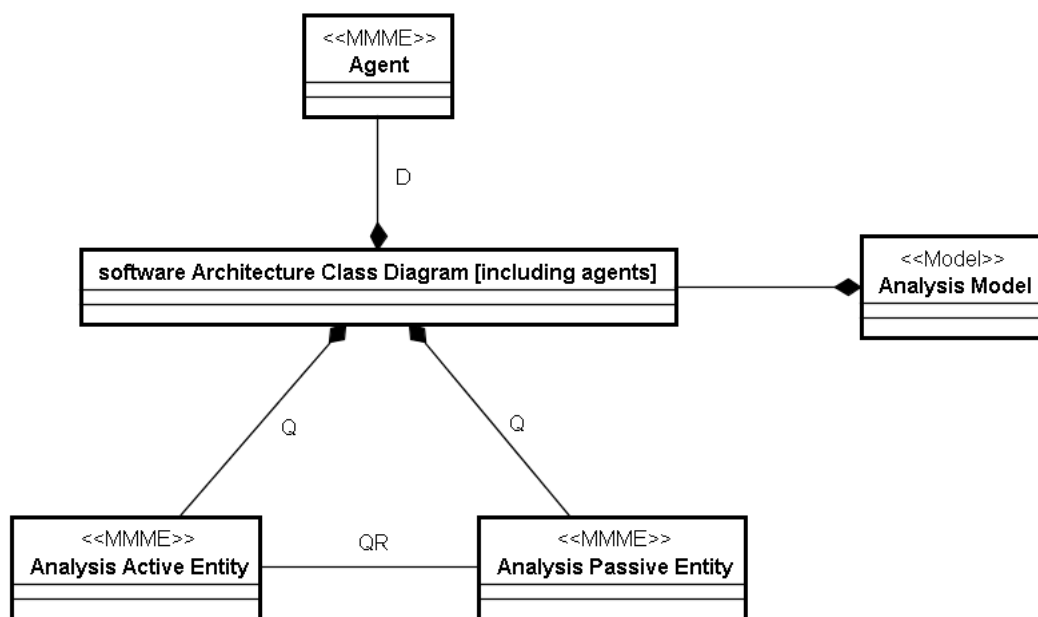
Deliverables

Software Architecture [including agents]

The Software Architecture document must be updated to indicate the classes that have been identified as agents.

Deliverable relationships with the MMM

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



Preconditions and concepts to be defined

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

As regards documents:

| Input | Output |
|--|--|
| Software Architecture AMAS Adequacy Synthesis | Software Architecture Class Diagram [including agents] |

As regards MAS metamodel elements:

| Input | To Be Designed | To be related | To be quoted |
|-------|----------------|---------------|--------------------------|
| | Agent | | Analysis Active Entities |

Guideline

To guaranty the good practice rules, you can stereotype the classes of entities identified as agent. It is for example used in the ADELFE method with the stereotype “cooperative agent”.

Composition Guideline

This fragment is based on the Adaptive Multi Agent System (AMAS) theory, so only cooperative agents are identified. It is recommended to compose it with fragments that take the AMAS theory in account.

Aspects of fragment

This fragment is based on the Adaptive Multi Agent System (AMAS) theory, so only cooperative entities are identified as agents. In other approaches, more entities could be agentified, but their non cooperative behaviour would be against the AMAS principle.

Dependency Relationships with other fragments

This fragment require the entities in interaction with the system to be identified and the classes of the system to be defined, so one or more fragments that do these tasks are needed before this fragment.

Also, before identifying agents, you should use a fragment that check if multi agent systems are adequate to the problem.

This fragment identifies agents, so it should go before fragments that work on agent concept.

9 - Entity Interactions Study

Introduction

Insert here a text description of the design process and the entire SPEM diagram that can help in positioning the fragment in the process workflow

Fragment Description

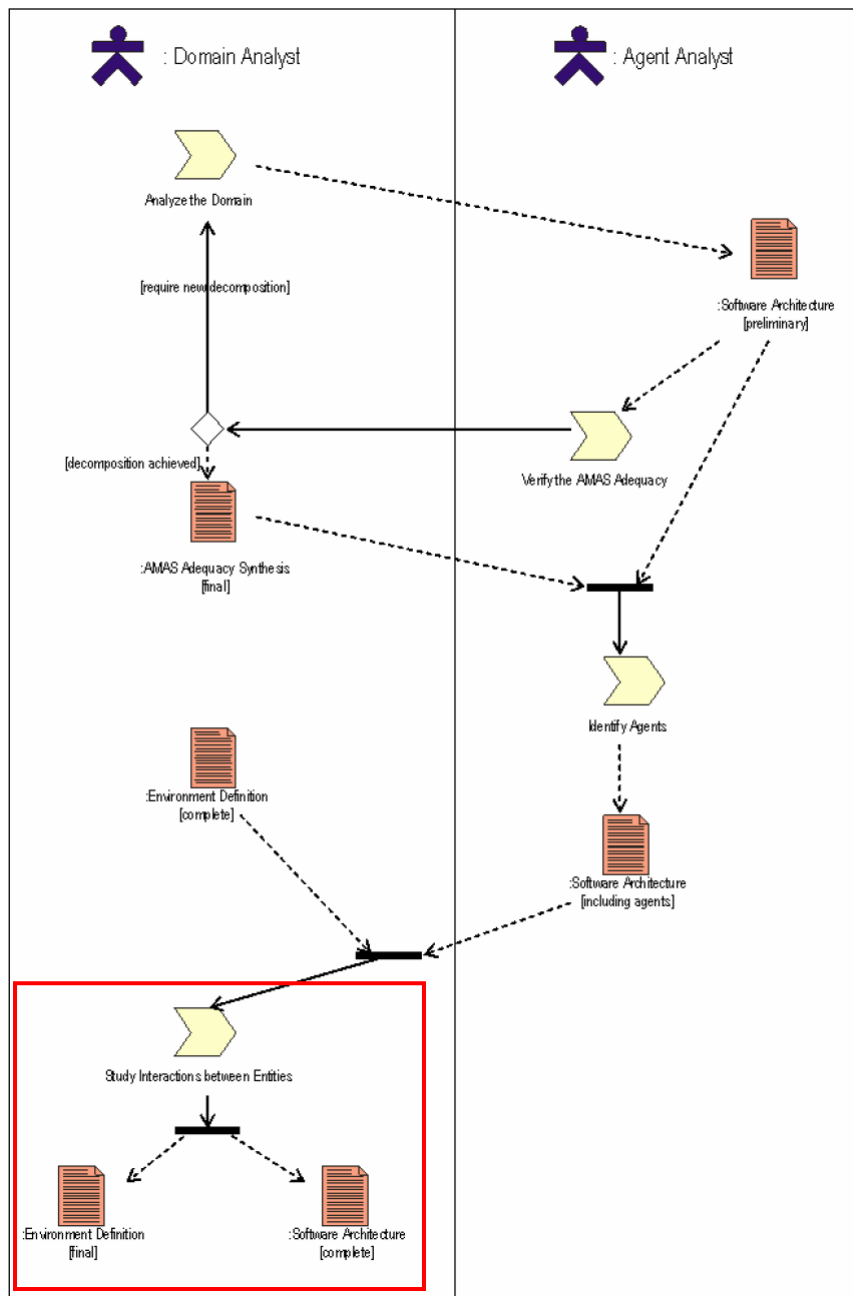
This fragment aims at making clear interactions between the identified entities.

Input: Environment definition [complete], software architecture [including agents]

Output: Environment definition [final], software architecture [complete]

Portion of Process workflow

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



Activities description:

| Activity | Activity Description | Roles involved |
|--|---|----------------|
| Active-Passive Entity Relationship Study | Study the active-passive entities relationships and model them with sequence diagrams or collaboration diagrams | Object Analyst |
| Active Entity | Study the active entities relationships and model | Object Analyst |

| | | |
|---------------------------|---|----------------|
| Relationships Study | them with sequence diagrams | |
| Agent Relationships Study | Study the agents relationships and model them with AUML protocol diagrams | Object Analyst |

Roles involved in this fragment are:

Object Analyst

They are described in the following subsections:

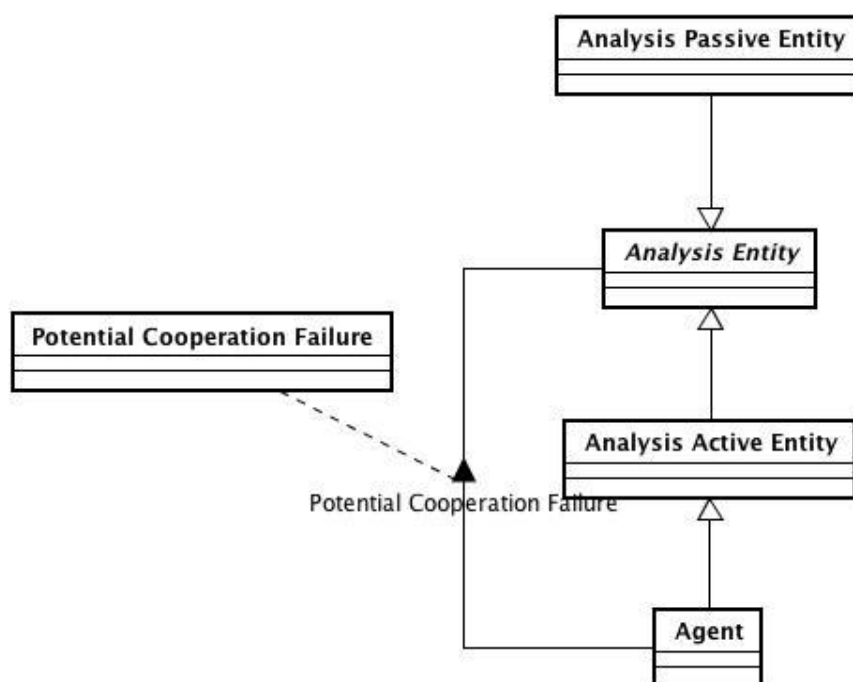
Object Analyst

Responsibilities of *Object Analyst* are:

- Active-Passive Entity Relationship Study
- Active Entity Relationships Study
- Agent Relationships Study

Relationship with the MAS metamodel

The portion of metamodel of this fragment is:



This fragment refers to the MAS meta-model adopted in *ADELFE* and contributes to define and describe:

-Potential Cooperation Failures

Deliverables

Environment Definition [final]

Report an example in a picture

Describe what type of diagram it is (class diagram, UML, Tropos, ...)

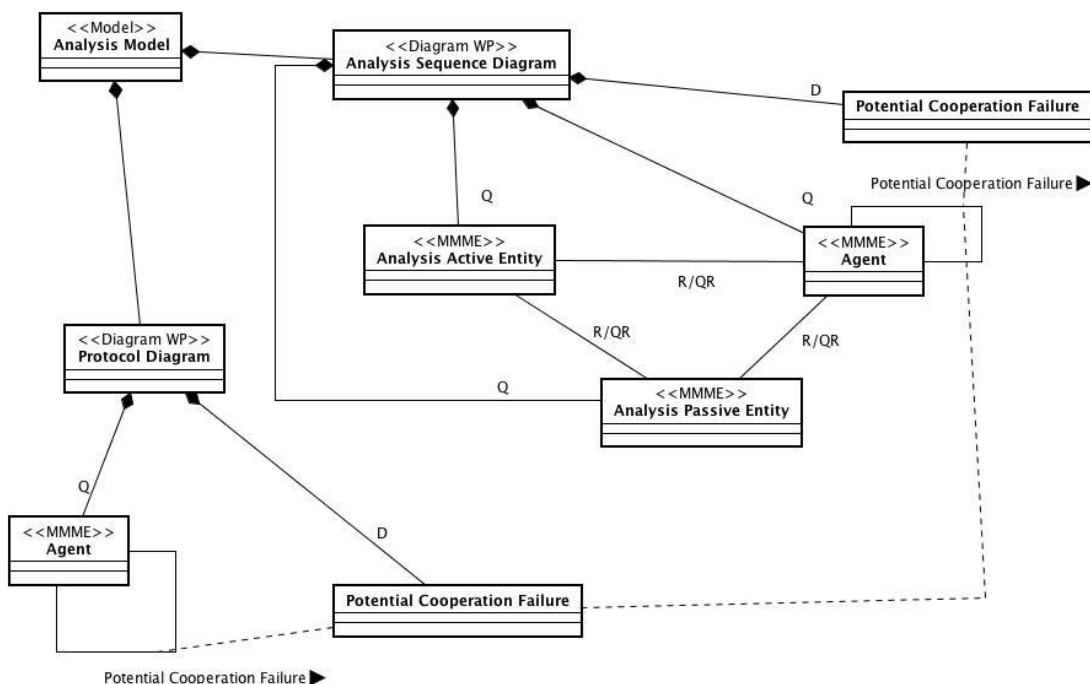
Software Architecture [complete]

Report an example in a picture

Describe what type of diagram it is (class diagram, UML, Tropos, ...)

Deliverable relationships with the MMM

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



Preconditions and concepts to be defined

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

As regards documents:

| Input | Output |
|---|-------------------------------------|
| Environment Definition [Complete] | Environment Definition [Final] |
| Software Architecture [including agents] | Software Architecture [Complete] |

As regards MAS metamodel elements:

| Input | To Be Designed | To be related | To be quoted |
|--|-------------------------------|--|--|
| Agent Analysis Active Entity Analysis Passive Entity | Potential Cooperation Failure | Agent-(Agent, Analysis Active Entity, Analysis Passive Entity) | Agent Analysis Active Entity Analysis Passive Entity |

Guideline

None

Composition Guideline

None

Aspects of fragment

The interactions can be expressed using UML and AUML diagrams.

Dependency Relationships with other fragments

To use this fragment, you need one or more fragments that identify the different agents and other entities of the system.

10 - ARCHITECTURE DEFINITION

Introduction

Fragment Description

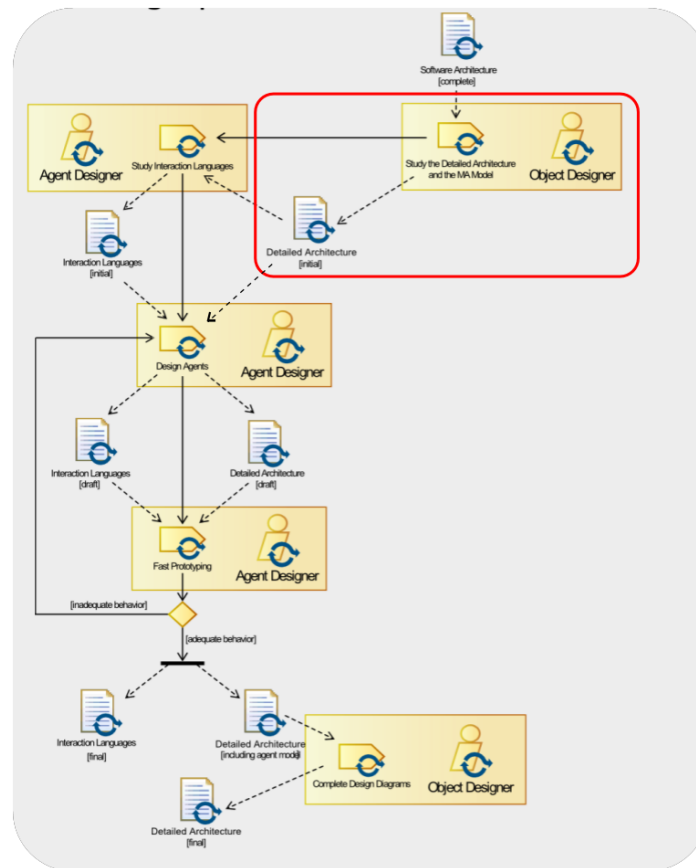
The main objective of this fragment is to define the detailed architecture of the system in term of packages, classes, objects and agents.

Input: Software architecture [complete]

Output: Detailed architecture [initial]

Portion of Process workflow

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



Activities description:

| Activity | Activity Description | Roles involved |
|----------|----------------------|----------------|
|----------|----------------------|----------------|

| | | |
|--------------------|--|-----------------|
| Define Packages | Identify package in order to group classes by field : PackageAgent, PackageEnvironment (entity active and/or passive), PackageInterface, PackageContoller | Object Designer |
| Share out concepts | Share out concepts of Activity 10 in a good package | Object Designer |

Roles involved in this fragment are:

Object Designer

They are described in the following subsections:

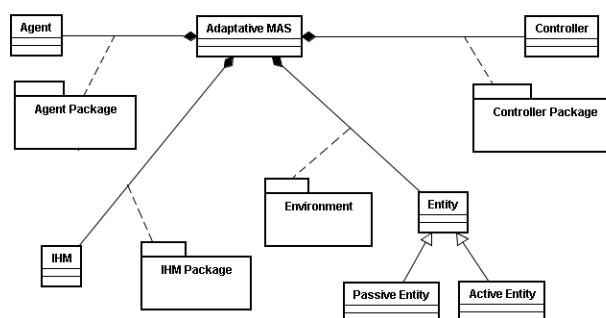
Object Designer

Responsibilities of *Object Designer* are:

- Define packs
- Share out classes into packs
- For each pack, make a class diagram

Relationship with the MAS metamodel

The portion of metamodel affected by this fragment is:



This fragment refers to the MAS meta-model adopted in *ADELFE* and contributes to define and describe:

- Package definitions

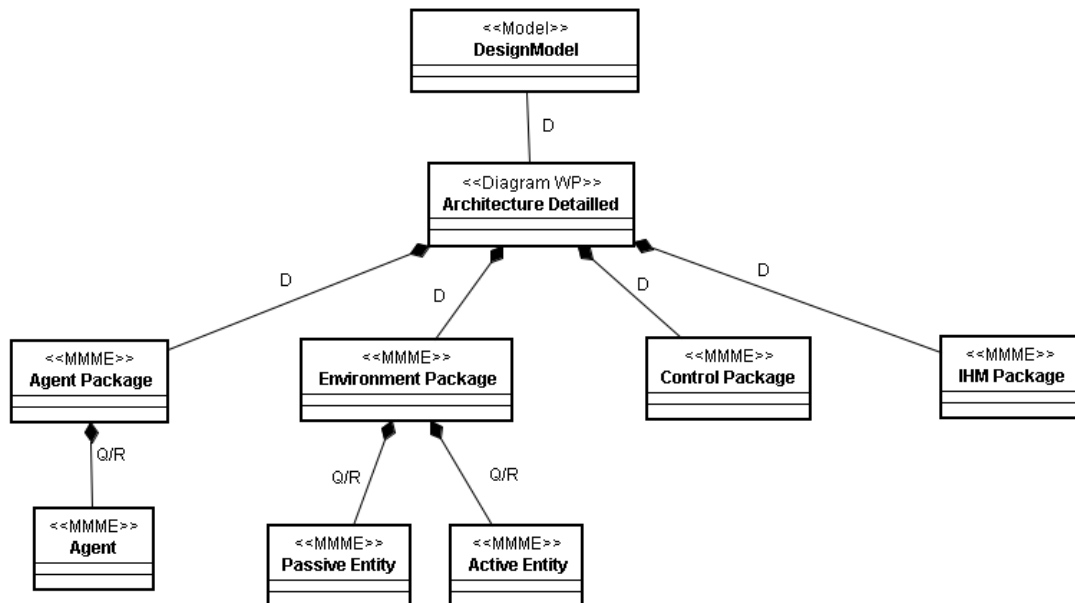
Deliverables

Detailed Architecture [Initial]

This fragment is composed by packages which contain concepts defined in A10.

Deliverable relationships with the MMM

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



Preconditions and concepts to be defined

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

As regards documents:

| Input | Output |
|-------------------------------------|------------------------------------|
| Software architecture [complete] | Detailed Architecture [Initial] |

As regards MAS metamodel elements:

| Input | To Be Designed | To be related | To be quoted |
|-------------------|----------------------------|---------------|--------------|
| Entities Agent | AdaptativeMultiAgentSystem | | |

Guideline

We define four packages:

- Agent package contains the agents
- Graphical interface package contains the graphical interface
- Control package contains the controller
- Environment package contains both the environment and the communication

Composition Guideline

None

Aspects of fragment

Class diagram are expressed using UML diagrams.

Dependency Relationships with other fragments

This fragment depends on the Agents Identification fragment and the Environment description fragment. These fragments must be realized before that one.

Glossary

None

11 - INTERACTION LANGUAGE DEFINITION

Introduction

We note that an interaction language is useless if the communication between agents is indirect, for instance, if the agents communicate via the environment.

If agents interact to communicate, for each scenario, they have to describe the exchanges of information between agents. These protocols will be specified via protocols diagram using AUMI.

Fragment Description

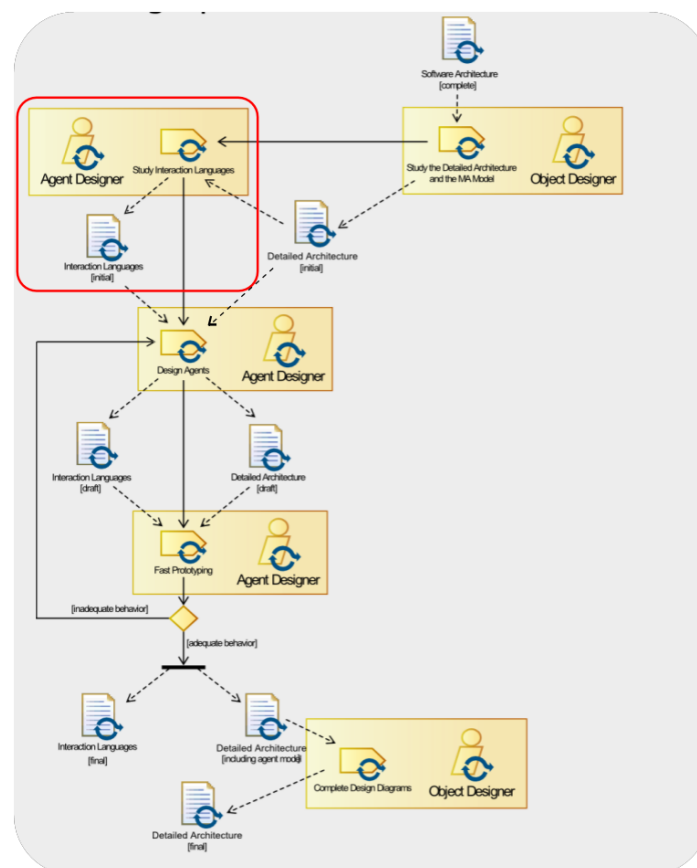
In this fragment we define the way the agents will interact.

Input: Detailed architecture [initial]

Output: Interaction languages [initial]

Portion of Process workflow

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



Activities description:

| Activity | Activity Description | Roles involved |
|--------------------|---|----------------|
| Define interaction | Define the way of communication between | Agent Designer |

| | | |
|--|---------------|--|
| | agents | |
|--|---------------|--|

Roles involved in this fragment are:

Agent Designer

They are described in the following subsections:

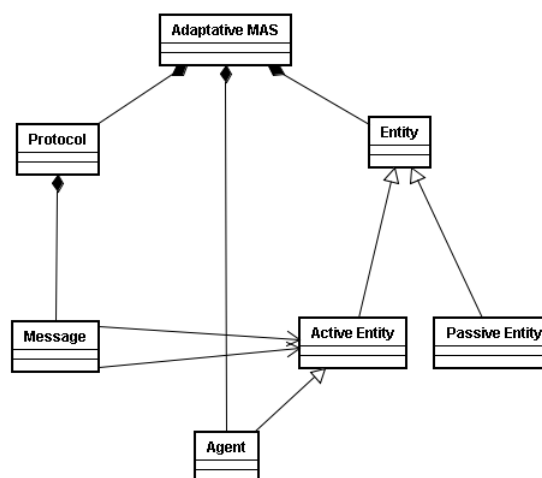
Agent Designer

Responsibilities of *Agent Designer* are:

- Define direct agent interaction

Relationship with the MAS meta-model

The portion of meta-model affected by this fragment is:



This fragment refers to the MAS meta-model adopted in *ADELFE* and contributes to define and describe:

- Protocol
- Message

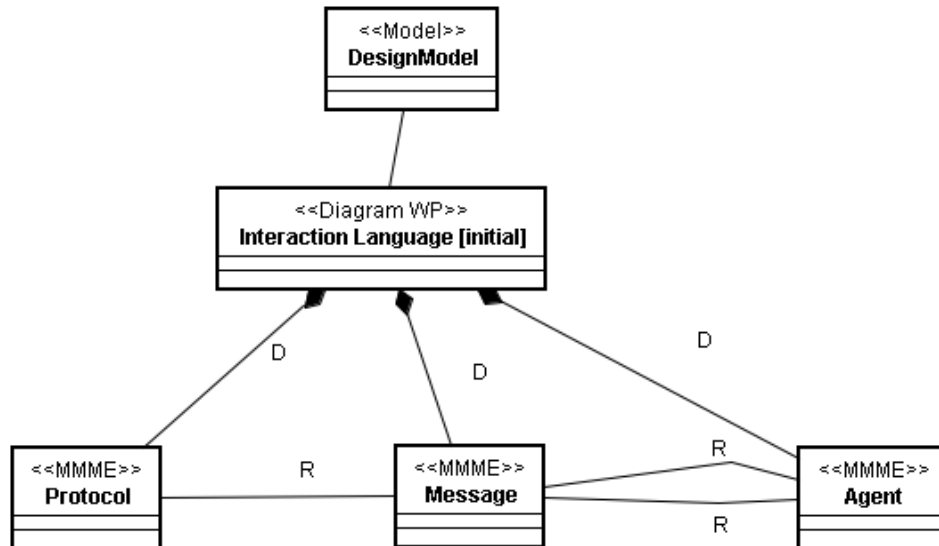
Deliverables

Interact Language [Initial]

Interact Language represent the language interaction diagrams.

Deliverable relationships with the MMM

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



Preconditions and concepts to be defined

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

As regards documents:

| Input | Output |
|------------------------------------|-----------------------------------|
| Detailed Architecture [Initial] | Interaction language [Initial] |

As regards MAS metamodel elements:

| Input | To Be Designed | To be related | To be quoted |
|-------|---|--|--------------|
| Agent | CooperativeInteractionProtocol Message | CooperativeInteractionProtocol – Message Message-Message Message-ActiveEntity | |

Guideline

None

Composition Guideline

None

Aspects of fragment

Class diagram are expressed using UML diagrams.

Dependency Relationships with other fragments

This fragment depends on Agents identification fragment.

Glossary

None

12 - AGENT DEFINITION

Introduction

This fragment and the following fragment come from the same action but recent research shows that separate the nominal behavior of a cooperative agent from its cooperative one simplifies the visualization. This fragment defines the nominal behavior.

As future work, this fragment will be replaced by another one which comes from another methodology eventually more specific.

Fragment Description

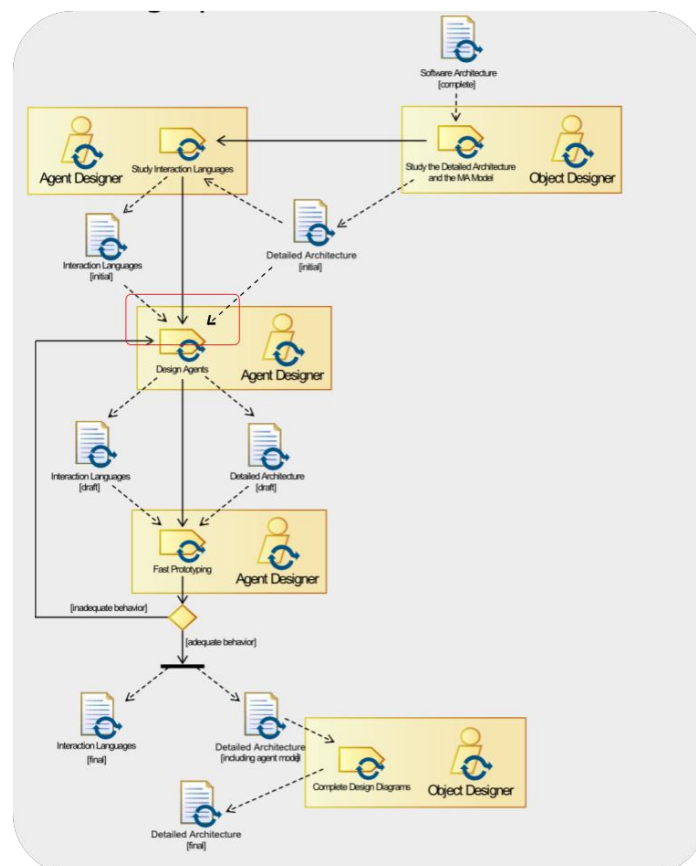
This fragment, for every agent previously identified, aims to define its behavior: the skill, the aptitudes, an interaction language, a world representation.

Input: Detailed architecture [initial], Interaction Language [initial]

Output: Detailed architecture [intermediate], Interaction Language [intermediate]

Portion of Process workflow

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



Activities description:

| Activity | Activity Description | Roles involved |
|----------|----------------------|----------------|
|----------|----------------------|----------------|

| | | |
|-------------------------|--|----------------|
| Define nominal behavior | Define agent diagram and agent behavior rules. These diagrams are complementary in their function. Behavior diagram allows identifying characteristics and representation. Representation specifies needs in terms of perception and communication. Reciprocally, the agent structure description specifies the composition of its representation module. Skills come from recursive definition in terms of another skill or elementary action. Aptitudes are independent capacities of field. | Agent Designer |
|-------------------------|--|----------------|

Roles involved in this fragment are:

Agent Designer

They are described in the following subsections:

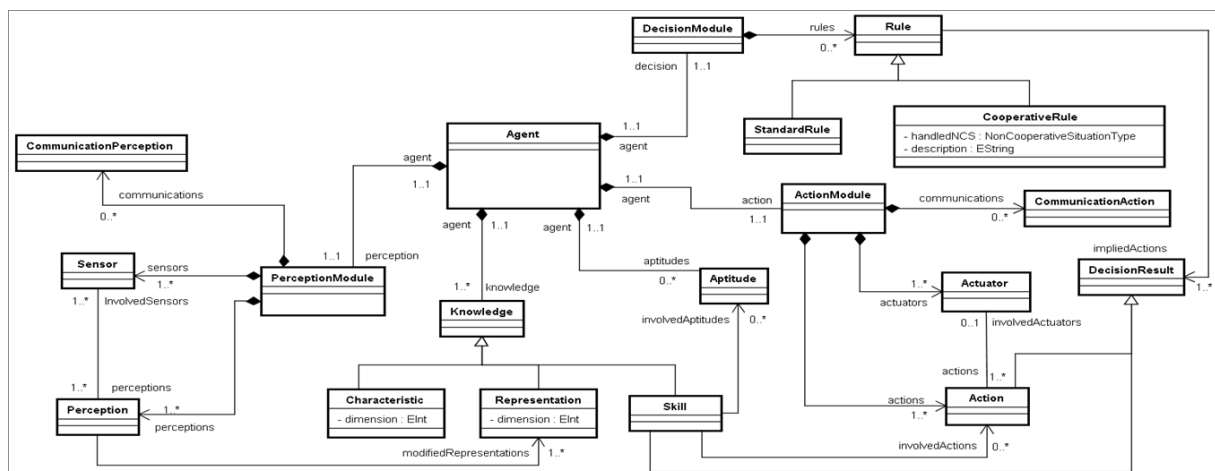
Agent Designer

Responsibilities of *Agent Designer* are:

- Define nominal behavior (structure and interaction language)

Relationship with the MAS metamodel

The portion of metamodel affected by this fragment is:



This fragment refers to the MAS meta-model adopted in *ADELFE* and contributes to define and describe:

- Action
- Actuator
- Aptitude
- Characteristic
- CommunicationAction
- CommunicationPerception
- Perception
- Sensor
- Representation
- Skill
- AgentState
- Condition
- DecisionResult
- StandardRule

Deliverables

Detailed Architecture [intermediate]

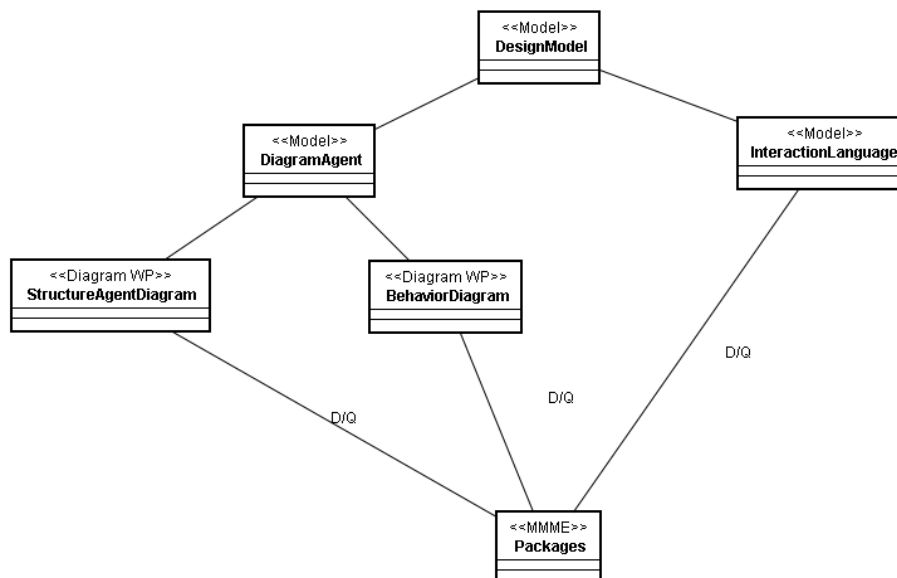
Detailed Architecture was enhanced with the nominal behavior

Interaction Language [intermediate]

Interact Language [intermediate] represent the language interaction diagrams composed by Interaction Language [initial] and interaction language of agent nominal behavior.

Deliverable relationships with the MMM

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



For readability, all packages previously quoted bring together in <<MMME Packages>>.

Preconditions and concepts to be defined

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

As regards documents:

| Input | Output |
|--|--|
| Detailed Architecture [initial] Interaction Languages [initial] | Detailed Architecture [intermediate] Interaction Languages [intermediate] |

As regards MAS metamodel elements:

| Input | To Be Designed | To be related | To be quoted |
|--|--|---|--------------|
| Agent CooperativeInteractionProtocol Message | Action Actuator Aptitude Characteristic CommunicationAction CommunicationPerception Perception Sensor Representation Skill AgentState Condition DecisionResult StandardRule | Action-Actuator Perception- Sensor Skill-Aptitude Skill-Action Rule- DecisionResult RuleAgentState | |

Guideline

The nominal behavior represents the basic behavior of an agent, what it does for achieving its local function.

cf. N. Bonjean, C. Bernon, P. Glize, Engineering Development of Agents using the Cooperative Behaviour of their Components, <http://ceur-ws.org/Vol-494/masspaper5.pdf>

Composition Guideline

None

Aspects of fragment

None

Dependency Relationships with other fragments

This fragment depends on Architecture Definition fragment and Interaction Language Definition fragment.

Glossary

None

13 - NON COOPERATIVE SITUATION IDENTIFICATION

Introduction

This fragment and the following fragment come from the same action but recent research shows that separate the nominal behavior of a cooperative agent from its cooperative one simplifies the visualization. This fragment defines the cooperative behavior.

Fragment Description

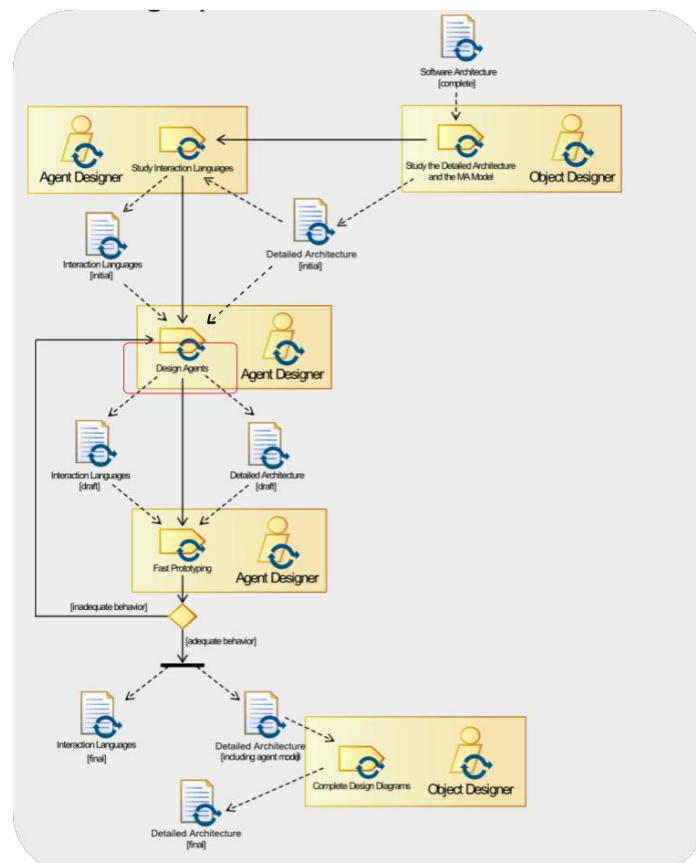
The main objective of this fragment is to define the Non Cooperative Situation (NCS) and the ways they are handled by the agents.

Input: Detailed Architecture [intermediate], Interaction Language [intermediate]

Output: Detailed Architecture [draft], Interaction Language [draft]

Portion of Process workflow

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



Activities description:

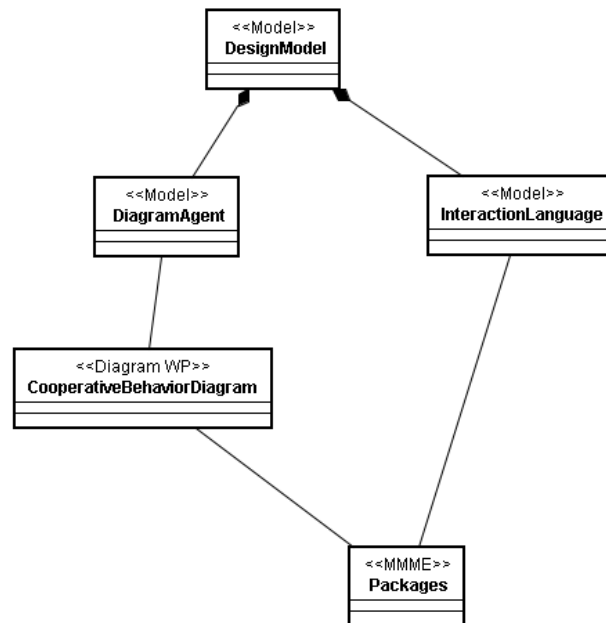
| Activity | Activity Description | Roles involved |
|--------------------|---|----------------|
| Define cooperative | Define tuning rules, reorganization and | Agent Designer |

Interaction Language [draft]

Interaction Language [draft] represents the interaction language contained in this system.

Deliverable relationships with the MMM

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



For readability, all packages previously quoted bring together in <<MMME Packages>>.

Preconditions and concepts to be defined

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

As regards documents:

| Input | Output |
|---|---|
| Detailed Architecture [intermediate] Interaction Language [intermediate] | Detailed Architecture [draft] Interaction Language [draft] |

As regards MAS metamodel elements:

| Input | To Be Designed | To be related | To be quoted |
|-------|----------------|---------------|--------------|
|-------|----------------|---------------|--------------|

| | | | |
|--|--|--|--|
| Action Actuator Aptitude Characteristic CommunicationAction CommunicationPerception Perception Sensor Representation Skill AgentState Condition DecisionResult StandardRule | Action Actuator Aptitude Characteristic CommunicationAction CommunicationPerception Perception Sensor Representation Skill AgentState Condition DecisionResult StandardRule Condition DecisionResult CooperativeRule | | Non Cooperative Situation Type Action Actuator Aptitude Characteristic CommunicationAction CommunicationPerception Perception Sensor Representation Skill AgentState Condition DecisionResult StandardRule |
|--|--|--|--|

Guideline

The adaptive behavior, added on top of the nominal one, aims at dealing with these cooperative failures in three different ways:

- by trying to adjust the values of the parameters used during the nominal behavior (tuning rules)
- by changing its relationships with others for trying to solve dead-ends (reorganization rules)
- by self-removing or creating other agents if NCS still remain (evolution rules)

cf. N. Bonjean, C. Bernon, P. Glize, Engineering Development of Agents using the Cooperative Behaviour of their Components, <http://ceur-ws.org/Vol-494/masspaper5.pdf>

Composition Guideline

None

Aspects of fragment

None

Dependency Relationships with other fragments

This fragment depends on Agent Definition fragment.

Glossary

None

14 - FAST PROTOTYPING

Introduction

This fragment tests behavior during working. This simulation must be simulated and limited in terms of development and must allow identifying behavior lacks or behavioral anomalies.

Fragment Description

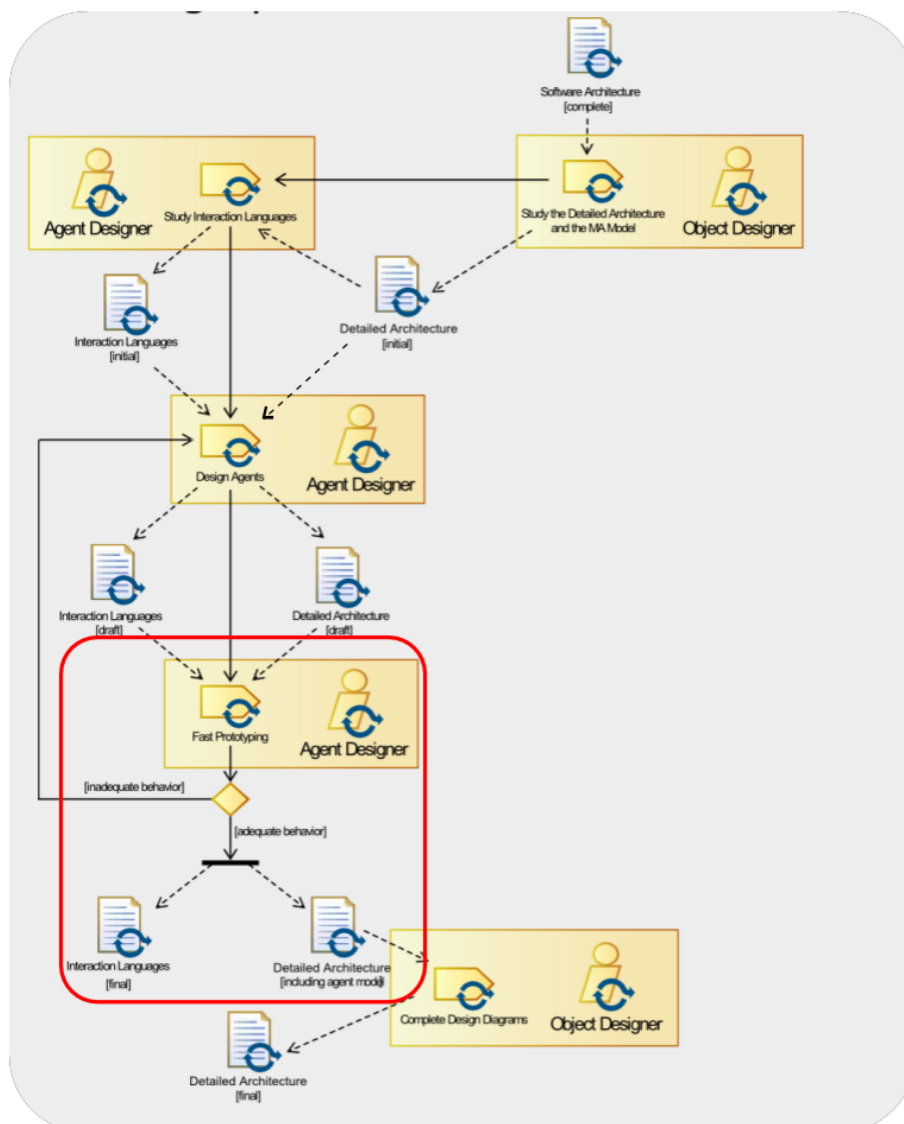
In this fragment, the identified behaviors of an agent are tested.

Input: Detailed Architecture [draft], Interaction Languages [draft]

Output: Detailed Architecture [including agent model], Interaction Languages [final]

Portion of Process workflow

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



Activities description:

| Activity | Activity Description | Roles involved |
|---------------------|---|----------------|
| Test agent behavior | Define simulation environment and implement methods to test. If agent behavior is unsuitable, come back to Agent Definition fragment. | Agent Designer |

Roles involved in this fragment are:

Agent Designer

They are described in the following subsections:

Agent Designer

Responsibilities of *Agent Designer* are:

- Test agent behavior

Relationship with the MAS metamodel

The portion of metamodel affected by this fragment is:

This fragment refers to the MAS meta-model adopted in *ADELFE* and contributes to define and describe:

-

Deliverables

Detailed Architecture [Including agent model]

Detailed Architecture [Including agent model] is composed by packages (environment, ihm, agent and controller) where agent package is defined and verified.

Interaction Language [final]

Interaction Language [final] represents all the interaction languages contained in this system.

Deliverable relationships with the MMM

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

Preconditions and concepts to be defined

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

As regards documents:

| Input | Output |
|---|---|
| DetailedArchitecture [draft] Interaction Languages [draft] | DetailedArchitecture [including agent model] Interaction Languages [final] |

As regards MAS metamodel elements:

| Input | To Be Designed | To be related | To be quoted |
|--|----------------|---------------|--------------|
| Agent Action Actuator Aptitude Characteristic CommunicationAction CommunicationPerception Perception Sensor Representation Rule Skill AgentState Condition CooperativeRule DecisionResult | | | |

| | | | |
|--------------|--|--|--|
| StandardRule | | | |
|--------------|--|--|--|

Guideline

None

Composition Guideline

None

Aspects of fragment

None

Dependency Relationships with other fragments

This fragment depends on Non Cooperative Situations Identification fragment.

Glossary

None

15 - ARCHITECTURE REFINEMENT

Introduction

This fragment is the last fragment of WD4. Before beginning the establishment, the detailed architecture must be over and all the diagrams must be complete.

Fragment Description

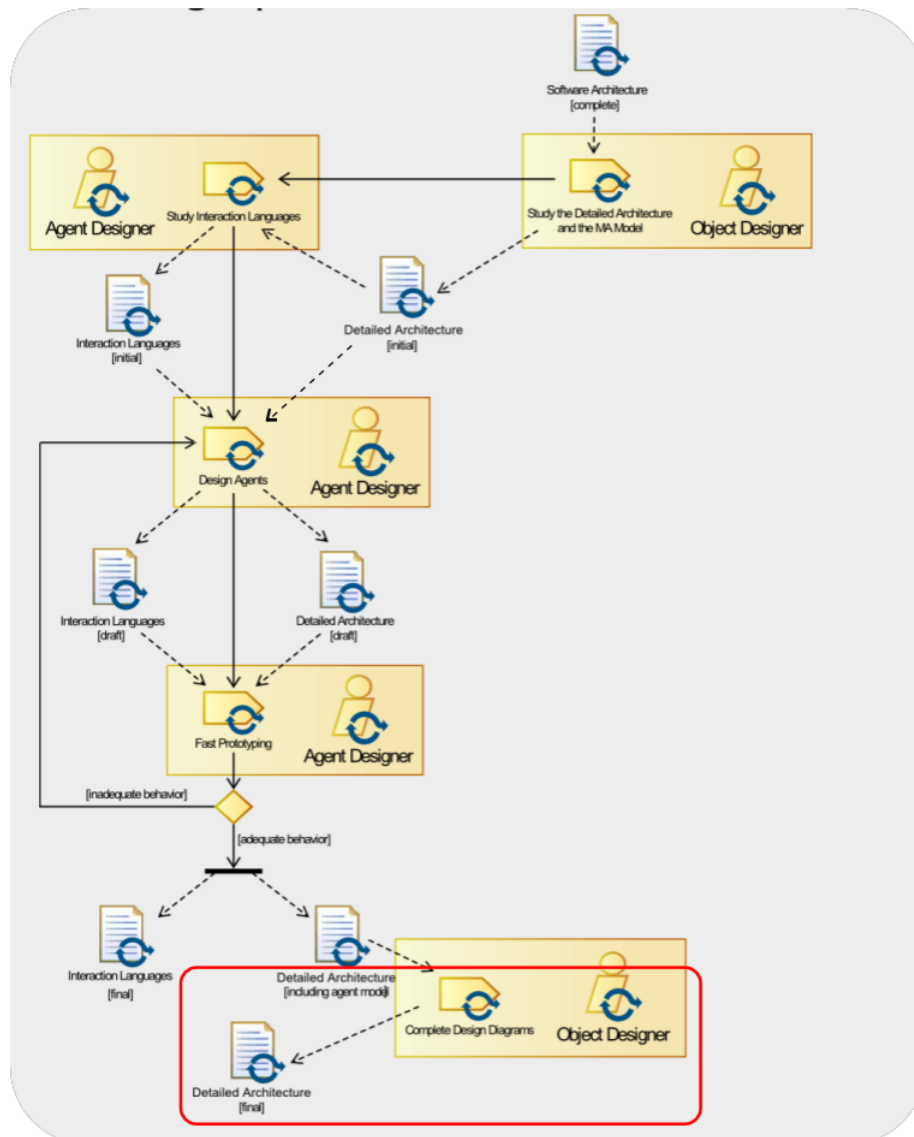
This fragment aims to complete the system architecture and design activities.

Input: Detailed Architecture [including agent model]

Output: Detailed Architecture [final]

Portion of Process workflow

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



Activities description:

| Activity | Activity Description | Roles involved |
|-------------------|--|-----------------|
| Complete diagrams | Complete class diagrams previously established | Object Designer |

Roles involved in this fragment are:

Object Designer

They are described in the following subsections:

Object Designer

Responsibilities of *Object Designer* are:

- Complete diagrams

Relationship with the MAS metamodel

The portion of metamodel affected by this fragment is:

This fragment refers to the MAS meta-model adopted in *ADELFE* and contributes to define and describe:

- State expression
- State Variable

Deliverables

Detailed Architecture [final]

Detailed Architecture [final] is textual documents which describe detailed architecture.

Deliverable relationships with the MMM

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

Preconditions and concepts to be defined

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

As regards documents:

| Input | Output |
|--|-------------------------------|
| Detailed Architecture [including agent model] | Detailed Architecture [final] |

As regards MAS metamodel elements:

| Input | To Be Designed | To be related | To be quoted |
|--|----------------------------------|-----------------------------------|--------------|
| Action Actuator Aptitude Characteristic CommunicationAction CommunicationPerception Perception Sensor Representation Skill AgentState Condition DecisionResult StandardRule | StateExpression StateVariable | StateExpression- StateVariable | |

Guideline

None

Composition Guideline

None

Aspects of fragment

None

Dependency Relationships with other fragments

This fragment depends on Fast Prototyping fragment.

Glossary

None

16 - ARCHITECTURE EXTRACTION

Introduction

Fragment Description

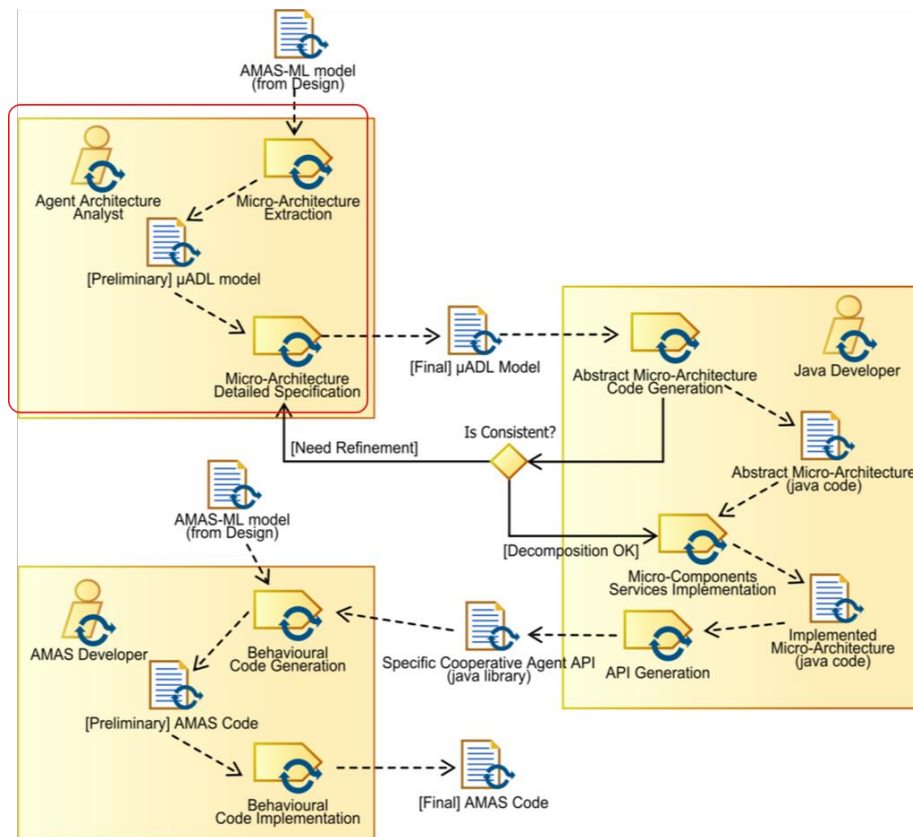
In this fragment, a muADL model is generated from the AMAS-ML model.

Input: Architecture Detailed [final], Interaction Languages [final]

Output: muADL Model

Portion of Process workflow

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



Activities description:

| Activity | Activity Description | Roles involved |
|--|---|----------------------------|
| Micro-Architecture Extraction | Change model into first version of muADL model | Agent Architecture Analyst |
| Micro-Architecture Detailed Implementation | Verify the model transformation and modify muADL model if it need | Agent Architecture Analyst |

Roles involved in this fragment are:

Agent Architecture Analyst

They are described in the following subsections:

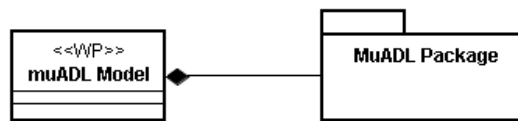
Agent Architecture Analyst

Responsibilities of *Agent Architecture Analyst* are:

- Micro-Architecture Extraction
- Micro-Architecture Detailed Implementation

Relationship with the MAS metamodel

The portion of metamodel affected by this fragment is:



Preconditions and concepts to be defined

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

As regards documents:

| Input | Output |
|---|-------------|
| Detailed Architecture [final], Interaction Languages [final] | muADL Model |

As regards MAS metamodel elements:

| Input | To Be Designed | To be related | To be quoted |
|--|--|---------------------------|--------------|
| Action Actuator Aptitude Characteristic CommunicationAction CommunicationPerception Perception Sensor Representation Skill AgentState Condition DecisionResult StandardRule Condition DecisionResult CooperativeRule StateExpression StateVariable | MuArchitecture MuComponent Level LevelComponent ContainerLevel ApplicationLevel MethodReference MethodVisibility Element DataType Feature Method Model | MuComponent- Interface | |

Guideline

None

Composition Guideline

None

Aspects of fragment

None

Dependency Relationships with other fragments

This fragment depends on Agent Definition fragment, Non Cooperative Situations Identification fragment and Architecture Refinement fragment.

Glossary

None

17 - ARCHITECTURE IMPLEMENTATION

Introduction

Fragment Description

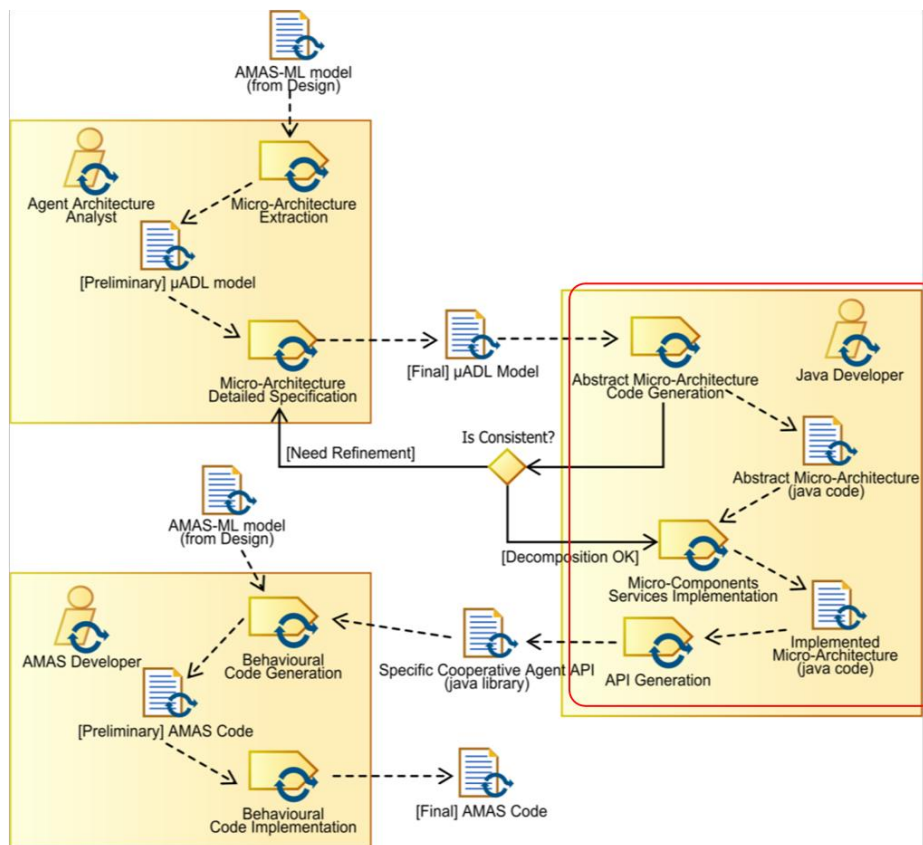
In this fragment, an API for the modelised agent type is generated.

Input: muADL

Output: Specific Cooperative Agent API (java library)

Portion of Process workflow

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



Activities description:

| Activity | Activity Description | Roles involved |
|---|--|----------------|
| Abstract Micro-Architecture Code Generation | Generate automatically code from abstract Micro Architecture | Java Developer |
| Micro-Components Services Implementation | Implement manually micro-components in java code | Java developer |

| | | |
|----------------|-----------------------------------|----------------|
| API generation | Generation of API in terms of jar | Java Developer |
|----------------|-----------------------------------|----------------|

Roles involved in this fragment are:

Java Developer

They are described in the following subsections:

Java Developer

Responsibilities of *Java Developer* are:

- Generate automatically MAY into JavaCode
- Implement manually micro-composant
- Generate semi-automatically API

Relationship with the MAS metamodel

The portion of metamodel affected by this fragment is:



This fragment refers to the MAS meta-model adopted in *ADELFE* and contributes to define and describe:

- Jar File

Deliverables

Specific Cooperative Agent API (java library)

Specific Cooperative Agent is

Deliverable relationships with the MMM

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



Preconditions and concepts to be defined

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

As regards documents:

| Input | Output |
|-------------|--|
| muADL Model | Specific Cooperation Agent API (java library) |

As regards MAS metamodel elements:

| Input | To Be Designed | To be related | To be quoted |
|--|-------------------|---------------|--------------|
| MuArchitecture MuComponent Level LevelComponent ContainerLevel ApplicationLevel MethodReference MethodVisibility Element DataType Feature Method Model | AMAS-ML model jar | | |

Guideline

None

Composition Guideline

None

Aspects of fragment

None

Dependency Relationships with other fragments

Architecture Implementation fragment depends on Architecture Extraction fragment.

Glossary

18 - BEHAVIOUR IMPLEMENTATION

Introduction

Fragment Description

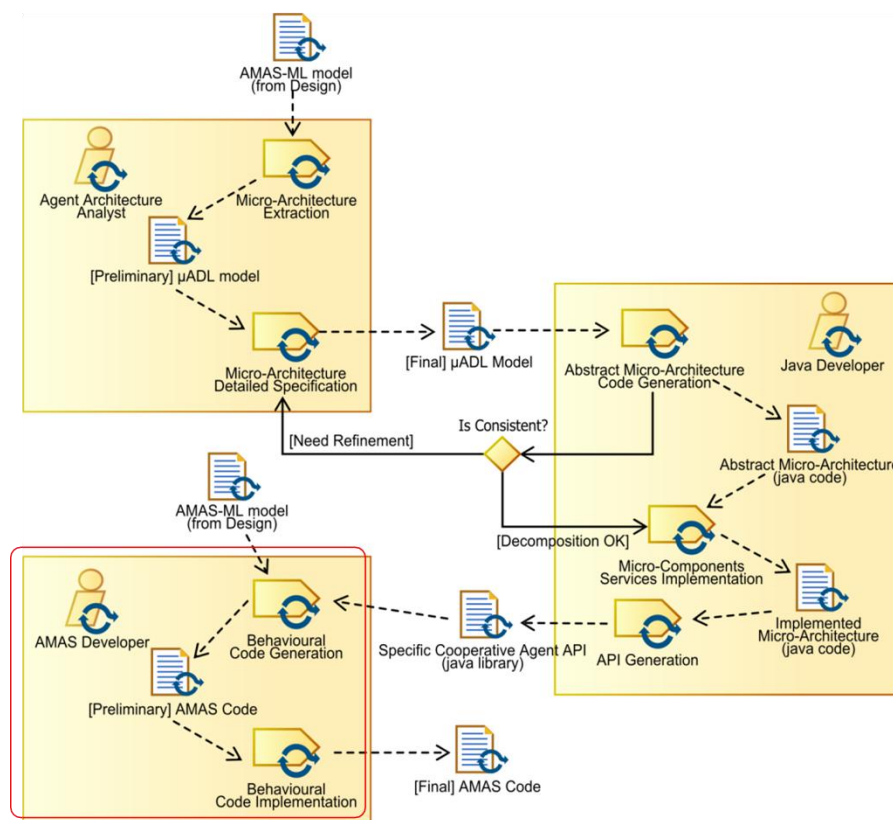
The goal of this fragment is to generate and implement the behavioral rules of the agents.

Input: Specific Cooperative Agent API (java library)

Output: AMAS Code

Portion of Process workflow

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



Activities description:

| Activity | Activity Description | Roles involved |
|------------------------------|---------------------------------|----------------|
| Behavior code Generation | Generate the code automatically | AMAS Developer |
| Behavior Code Implementation | Generate the code manually | AMAS Developer |

Roles involved in this fragment are:

AMAS Developer

They are described in the following subsections:

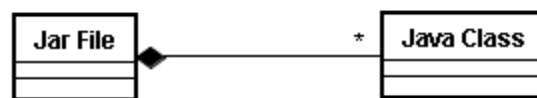
AMAS Developer

Responsibilities of *AMAS Developer* are:

-

Relationship with the MAS metamodel

The portion of metamodel affected by this fragment is:



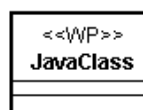
This fragment refers to the MAS meta-model adopted in *ADELFE* and contributes to define and describe:

- Java Class

Deliverables

Deliverable relationships with the MMM

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



Preconditions and concepts to be defined

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

As regards documents:

| Input | Output |
|---|-----------|
| Specific Cooperative Agent API (java library) | AMAS Code |

As regards MAS metamodel elements:

| Input | To Be Designed | To be related | To be quoted |
|-------------------|----------------|---------------|--------------|
| AMAS-ML model jar | Java Class | | |

Guideline

None

Composition Guideline

None

Aspects of fragment

None

Dependency Relationships with other fragments

This fragment depends on architecture implementation fragment.

Glossary