

Requirements analysis and evaluation procedures: *[use case provider]*

Introduction

The requirement engineering phase is a key step of the project. To develop the right system that satisfies the needs and expectations of the users, computer scientists need to have a complete understanding of the requirements. Past experiences indicate that determining the requirements can be a very hard task. Many factors contribute to that:

- Computers scientists and domain experts usually speak “different languages”, due to very different backgrounds.
- Users of the system usually do not have a clear idea about the true requirements at the beginning of the project. This is mainly because they are considering a new system that does not yet exist. As usual, there will be many new, and previously unconsidered, requirements in later phases of the project.
- Computer scientists usually have their own ideas about how the system should work and this biases their understanding of the requirements.
- Different users of the system usually have different requirements. Common misspecifications are due to this problem.
- The existence of different contexts in which the system will be used might also affect the requirements and need to be considered.



D. Pilone, R. Miles “Head First Software Development”. O’Reilly.

[Mustang is either a famous sport car brand or a free-roaming horse]

The requirements engineering process in AMIDST

The overall requirement engineering process will be carried out in an iterative fashion that is expected to involve a high level of cooperation and interaction between the partners. During this process the document for the requirements analysis will be gradually refined and expanded. Below is a description and an illustration of the expected requirements engineering process within AMIDST. The overall process is divided into the following phases:

1. **Preparation I.** This phase started with Work package 1 and the collection of data characteristics.
2. **Elicitation.** The distribution of the present document/template marks the initialization of this phase. Its aim is to get an initial high-level description of the different use cases and their requirements. This information should be specified by the use case providers (in this case Daimler). Please, consider that the use-cases and requirements (*which should be within the scope of AMIDST*) are the ones that will be addressed in the project. Once the use case providers return the present document with the requested information, we expect to provide feedback on the submitted information in order to refine the information in collaboration with the use case provider. At the end of the elicitation phase, we aim to have a first coherent description of the requirements for each use case provider.
3. **Prioritization.** In this phase the use case providers should complete an extended version of the document template used in the previous phase. This template will firstly be used to link each of the requirements to the relevant work packages and tasks in the AMIDST project. Moreover, the template will also allow the use case providers to provide a more fine grained prioritization of the relevant requirements for the AMIDST framework.
4. **Validation.** In the last phase we will collect the requirements of the three use-case providers to get the “big picture”. We will then discuss to what extent the requirements can be accommodated within the project. Revisions and negotiations of the detailed requirements may therefore be necessary.
5. **Evaluation and Testing.** In this phase, focus will be on the elicitation of the evaluation and testing procedures that are going to be used in the AMIDST project. This phase starts with the distribution of a new document template. Its aim is to obtain a high level description of the evaluation and testing methods that will be used to measure the performance of the AMIDST framework in relation to the use cases elicited in the previous phases.

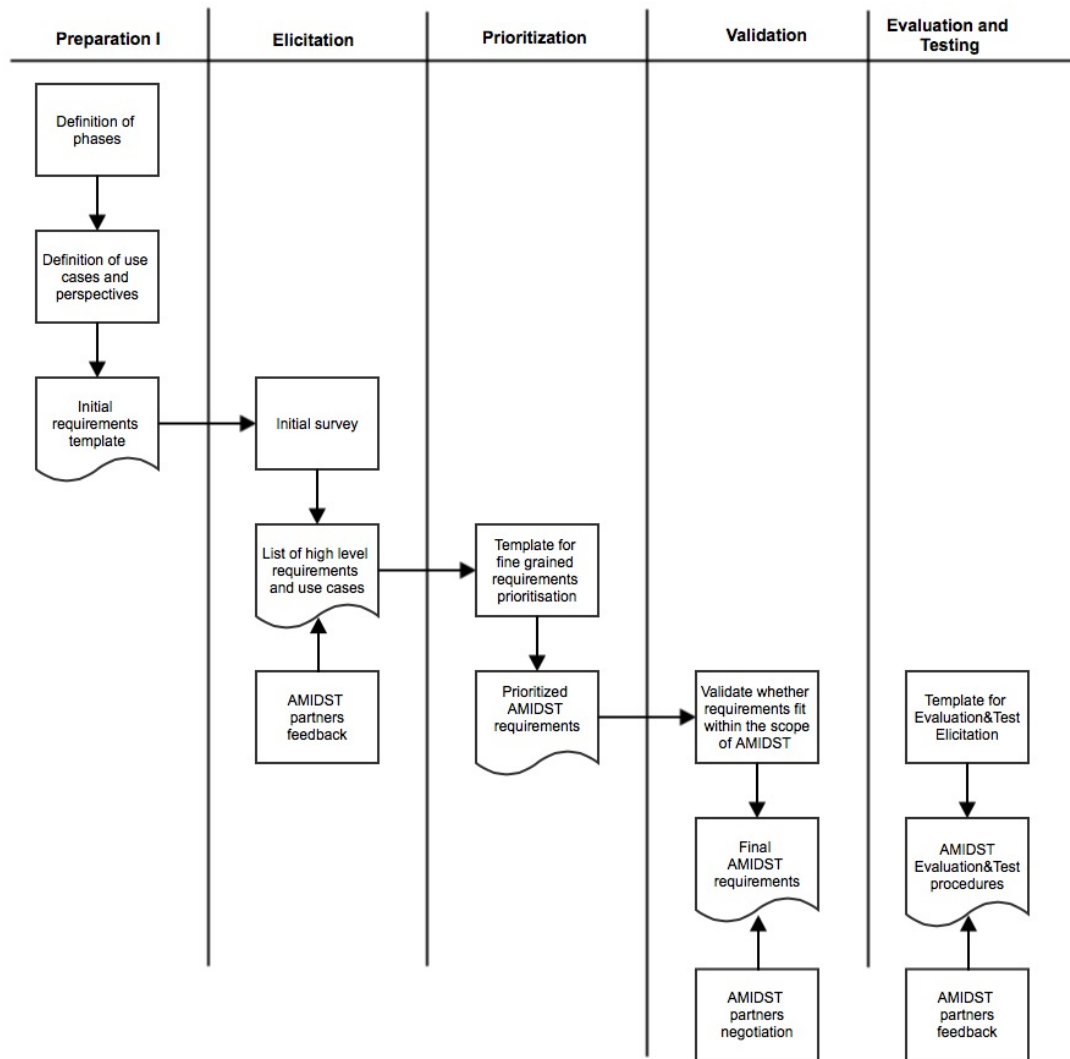


Figure 1: Requirements engineering process within AMIDST.

It is important to emphasize again that the requirement for the use case providers obviously need to be aligned with the AMIDST objectives. Hence, when specifying the requirements it is important to also justify and demonstrate their necessity in relation to AMIDST.

[Use case provider]: System context

Please describe the context for the AMIDST framework for this application domain. In particular, describe relevant aspects of existing systems that can influence the requirements for the AMIDST framework and its future impact.

User groups

At this point we will try to identify the different user groups *within your organizations* that will use the developments of the ADMIST project. The potential user groups in the AMIDST project are very diverse, ranging from active AMIDST partners with intimate knowledge of the AMIDST framework to users in the organizations not directly involved in the AMIDST project.

The issue here is to identify those user groups that may have different requirements and needs for the system. It is therefore important to identify and characterize these different user groups, because each requirement may be targeted to a specific user group.

As inspiration, for *[Use case provider]* we could tentatively envision (at least) three user groups:

- **User Group 1:** *[Insert example of hypothetical user group]*
- **User Group 2:** *[Insert example of hypothetical user group]*
- **User Group 3:** *[Insert example of hypothetical user group]*

Please, include the information for each user group in a table as the one provided below:

User Group 1: *[Name of user group]*

User Group 2: *[Name of user group]*

User Group 3: *[Name of user group]*

AMIDST use cases

The requirements for AMIDST are intrinsically related to the usage of the AMIDST framework. To simplify the requirements engineering process and provide a context for the individual requirements, we need to characterize the general use cases that both the AMIDST framework and the deployed system should address. In order to support this process it is important that the use cases are within the scope of the AMIDST project and does e.g. not focus on possible future unplanned extensions and integrations of the AMIDST framework.

A **use case** is typically a list of steps or actions that define the interactions between an *actor* and an AMIDST framework or system in order to achieve a certain objective. The term actor should be considered in a broad sense and can e.g. refer to an actual user or a software module in DrillEdge. When specifying each use case, try to answer to the following questions:

1. Who/Which are the actors involved in the use case? An actor is an external person or entity that interacts with the AMIDST framework. If the entity is a person, please indicate which user group s/he belongs to.
2. What is the main event that initiates the use case? This could e.g. be an external business event or a system event that causes the use case to begin, or it could be the initial step in a normal work flow.
3. What are the main user actions and system responses that will take place during the normal execution of the use case?. This dialog sequence will ultimately lead to accomplishing the goal implied by the use case name and description.
4. How do you evaluate the success of a use case.

Please, consider that a use case should ideally be indivisible. If a use case can be decomposed into multiple sub-use cases, each with a well-defined sub-objective relevant for AMIDST, then these sub-use cases should be described separately.

In order to indicate the relative importance of the use-cases, please assign points (between 0 and 100) to each use case so that the sum of all assigned points is 100.

Use Case 1: <i>[Name of use case]</i>	N. of Points: XX

Use Case 2: <i>[Name of use case]</i>	N. of Points: XX

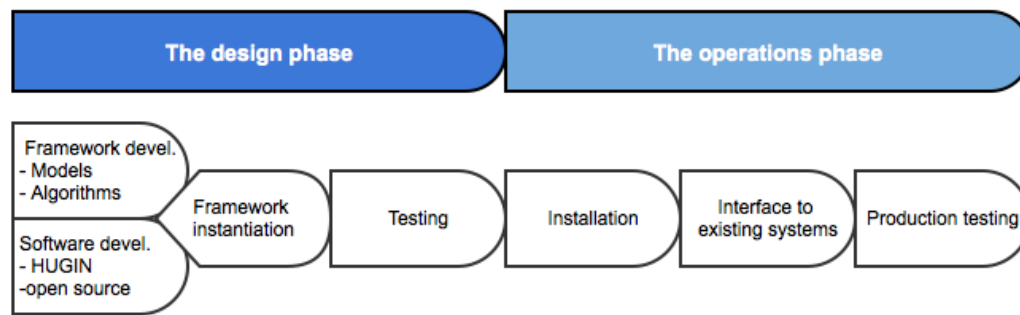
Use Case 3: <i>[Name of use case]</i>	N. of Points: XX

Use-case requirements

Based on the identified use-cases and user groups, we will in this section elicit the general system requirements. The main question that we try to answer here is: what should the software be able to address in relation to the previously detailed use-cases?

In order to structure the requirements we are going to follow Kano's model¹, which is a standard methodology to determine product and service requirements. We will also ask you to specify the requirements for the system at two different stages of the development process: *design/model development* and *operations*. Please again keep in mind that all requirements should be described considering the scope of the AMIDST project.

¹ Look at http://en.wikipedia.org/wiki/Kano_model for further information about this method.



For the *design stage* we look for general functionality requirements for the system, i.e. what the system should do and support. In the above figure, we detail key steps inside this phase. The first step consists of the design of the general framework (models and algorithms) as well as the design and development of the software tools. In a second step, the general framework and software will be instantiated for each specific use case. Finally, initial tests of the use case instantiated framework will be conducted.

At this design phase, possible design requirements could e.g. address

- the scope of the model
- the interpretability of the learned models
- the extent and type of domain knowledge that can be integrated into the models
- documentation
- etc.

The requirements for the *operation phase* concern the functionality of the deployed system. In the above figure, we decompose this phase into three stages: installation, interface to existing systems, and production testing. The requirements for this phase could e.g. address

- hardware constraints
- interfaces to existing software or data base systems
- inference functionality, i.e., what queries the system should be able to answer

When specifying the requirements, try to think in each one this stages and define the requirements in relation to each of them.

Please specify each requirement concisely, using no more than one or two sentences. If you find this difficult, it might mean that the requirement should be

split into smaller entities. Moreover, please indicate which use cases and user groups that are associated to each requirement. Additionally, for each requirement, please classify it as either must, should, or could:

- **Must:** These requirements include properties guaranteed in the AMIDST description of work document.
- **Should:** These requirements could be expected by the use-case provider, but is not explicitly agreed upon.
- **Could:** Optional requirements.

When filling the tables, please consider the following points

- Each requirement should be labeled in the following way: a unique identifier containing the name of the partner (i.e. "DAI", "CAJ" or "VER") followed by an identifier of the corresponding use case (from 1 to n) and, also, by an identifier of the requirement ("D" for design requirements and "O" for operations). For example DAI.U4.O2 refers to the second requirement from Daimler for the use case 4 in the operation phase.
- Each requirement should be linked to the work package and task in which it is involved. I.e., we should identify which part of the AMiDST project (work package and task) will address the requirement. If we do not find a clear place where the requirement can be located, this might indicate that this requirement falls out of the scope of the project.
- With the aim of providing a fine grained prioritization, each requirement should be scored with a number between 0 (lowest priority) and 100 (highest priority) in such a way that all requirements in the same table and within the same category ("Must", "Should" or "Could") have to sum up to 100 points. For example, you should distribute 100 points among the "Should" requirements of the design phase for Use Case 1.

Use Case 1

Requirements Operation Phase

Requirements Operation Phase

[illegible]