MOMOCS

Model driven Modernisation of Complex Systems

D2.1
XIRUP SUPPORTING TOOLS REQUIREMENTS

Dissemination Level:	Public
Work package:	WP2
Lead Participant:	ATOS
Contractual Delivery Date:	January 2007
Document status:	Final
Preparation Date:	31st January 2007
Document Version:	1.7

INDEX

INDEX OF TABLE

INDEX OF FIGURES

1 Introduction

1.1 Objectives and scope of the document

The present document aims at giving an overview of what kind of tools will be specified and developed in the scope of MOMOCS project. The inputs for the document are the case studies requirements (described in D1.2) and the XIRUP methodology requirements (described in D1.1). In order to determine the best tools according to the user requirements already analyzed, a deep study of the state of the art has been necessary.

The document will specify the supporting tools using a functional requirements definition for identified tool. The more detailed specification of the tools and their architecture will be produced in deliverable D4.1 Supporting Tools Specification.

1.2 Overview

The document starts with an introductory section; then an overview of the expected supporting tools; after there is a description of the functional requirements of the tools and some non-functional requirements definition that are also desirable to consider; and finally a conclusions section where the purposes of the document and its content are summarized.

2 Supporting Tools Overview

The Description of Work describes the need to develop three type of Supporting Tools for the XIRUP Methodology in the modernisation of information systems. Although the tools are classified in three types (Data, Process and *Ware), it does not mean that only three tools will be required for modernisation process. Thus, the identified tools will be grouped in the three types above mentioned.

In order to understand how the tools will be deployed, the following picture shows the basic infrastructure that MOMOCS proposes to accomplish the modernisation of systems supporting the XIRUP Methodology.

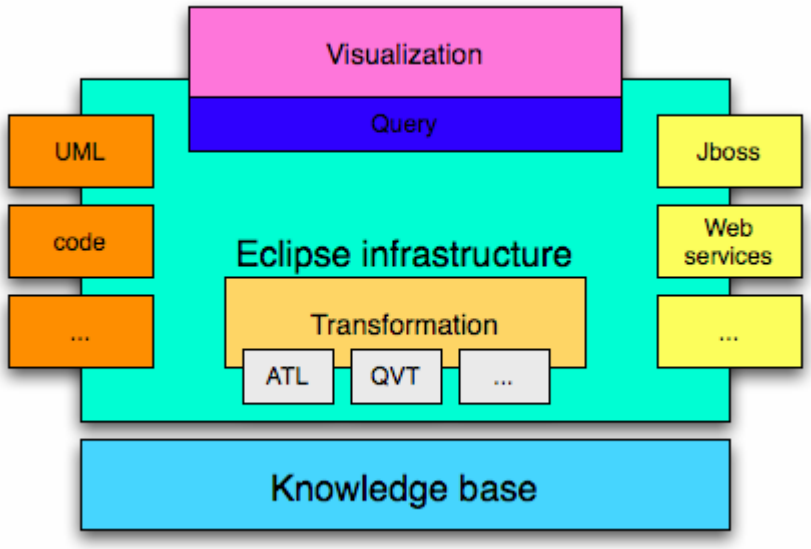

Figure 1Momocs tools conceptual architecture

As it is shown in the figure, the main concept is basically to populate a Knowledge Base, and using some Eclipse plug-ins (filters) query the base and visualize the system to decide which transformations apply to modernize it.

The main steps in the modernisation process will guide us in the definition of the tools we need to support this process. Let's use an example of modernisation process to understand why the identified

tools are needed. Imagine a legacy system in a company that can be modernised. The first step is to understand the system in order to model it as a XIRUP System Model (XSM). XSM is the set of components, with attributes and operations (interfaces) plus a set of links, dependencies or associations among these components. It will be described in an abstract syntax, but the concrete syntax can be, for instance, UML. The next step will be the analysis of the extracted TBMS model and generate a report for the user. Once the TBMS model is presented to the company user (domain expert), he/she will be assisted to edit and modify the model rules to get the appropriate model of the TBMS. Then, the TBMS model is visually presented to the user to help him/her in the modernisation of the system. In order get the MS model from the TBMS model, some transformations will be required. Then another tool will help him/her to create, edit and modify the transformations of the obtained model. There will be also a set of tools for hosting the repository of model type components and its management. Finally, another tool for supporting for the requirement elicitation will be also required.

Based on the above described schema and having in mind the methodology steps and the requirements elicitation as a result of this document, the following tools have been identified:

Tool	Description	Global Fuctions	Input	Output
K-Xtraction Tool	This Tool creates Xirup System model (XSM) of the TBMS from different external document or software sources. This tool is also capable of importing TBMS models in other formats than Xirup System metamodel.	Elaborate the TBMS models/documents compliant with Xirup System metamodel. Map legacy models/documents to XIRUP System Metatmodel.	1-Document unstructured (doc, txt, pop). 2-Document structured (xml, java, xmi, uml) XIRUP Meta-model.	XIRUP system model of the TBMS

XSM-Analysis Tool	Tool to model/edit TBMS/MS models using Xirup metamodel	Load a XSM and a set of constraints. Generate report and save it.	1-XIRUP System Model (XSM) of TBMS/MS. 2-Set of constraints created by human (specific domain) are needed at the beginning to be considered in analysis phase. 3-XIRUP Meta-model	1-Refined XSM of TBMS/MS. 2-Report about the compliance of XIRUP System Model to the set of constraints.
Constraints Editor Tool	Editor to build/edit model constraints and rules.	Load XIRUP Meta-model and set of constraints. Assist the user to edit/modify rules Check rules for well-formed validation	1-XIRUP expert and domain expert interaction 2-XSM of TBMS/MS. 3-Set of constraints/rules. 4-XIRUP Meta-model	Set of constraints/rules.
XSM-Visualization Tool	Graphical Xirup System model visualization tool with graphical capabilities for model edition.	Show the XIRUP System Model to the user using a graphical layout, with zoom facilities, view creation and management support.	XSM of the TBMS/MS	Different graphical views of the current model.
XSM-Transformation Tool	Creation/Edition/ Execution of XSM discreet and continue transformations assisted by wizards.	Creation/Editon/Execution of XSM transformations between TBMS and MS.	TBMS XSM, Xirup metamodel, MS metamodel, QVT compliant transformation	QVT compliant Transformation, MS XSM

XSM-Type Library Tool and frontend	Repository of Model Type Components, XSM models, management of the repository	Component type model and XSM model repository, types and models addition, edition, deletion, searching, browsing, etc	Component Type model, XSM models	Component type model/XSM model repository
Requirement Elicitation Tool	Support for requirement elicitation elaboration and its storage.	Requirement Creation/Edition/Searching/Browsing from the requirement repository.	Requirement information	Requirement specification Requirements repository.

Table 1 Momocs tools

3 Supporting Tools Requirements

This section enumerates all the functional requirements that are desirable for the supporting tools. The deliverable D1.2 has analyzed the two case studies in the project (Siemens and TID) and it has concluded with the list of all the unified requirements shared by two cases. The deliverable D1.1 has produced the desirable requirements for XIRUP Methodology based also on the requirements from the case studies.

Having these mentioned requirements, analysing the needs from case studies and considering the state of the art, the following table summarizes the requirements for supporting tools. The table contains the next fields:

- Number of requirement
- Description of requirement
- Further explanation that may help to better understand the requirement
- Priority of the requirement (essential, important, desirable, nice to have). This requirement has been mainly established by the case studies users.
- Related requirements, if the requirement is related to or depends on others
- Requirement source, if the requirement comes from one of the two case studies or both ones. Other requirement sources can be the deliverables produced on WP1. If the requirement has not been derived directly from those sources this field will remain empty.

It is important to point out that the priority of each requirement has been established mainly by the responsible partners in charge of providing the Momocs case studies, that is, this priority has to be considered as case study user request. Momocs tools development partners will try to implement in the tools those requirement as far as possible according with the following criteria:

1. Requirements with higher priority will be considered first.
2. Requirements considered affordable in terms of their technological feasibility (that is, whether or not the current technology allows their reasonable implementation), with affordable development efforts and available resources within the project lifetime, will be also faced first.

3.1 General requirements

This section collects those requirements that have been considered of general purpose either because they can be applied to any of the Momocs' tools or because they have not being assigned as they don't fit yet to any of the envisaged tools.

Req. ID: GR1

Description: Support to document tag creation and supervision of technical documentation associated to the model.

Explanation: During the creation of technical documentation, "tags" should be defined to identify information that is dependent on the model of the system. There should be a supervising engineering process that updates these tags in the documentation

Priority: Desirable.

Related Req. Ids: ATR4

Source UC: Siemens. Stackable Tray System

Req. ID: GR2

Description: Support the simulation of the behaviour of the system

Explanation: Momocs tools should permit the simulation (i.e. execution) of the processes in order to understand for example consistency, to identify bottleneck

Priority: Desirable.

Related Req. Ids:

Source UC: Siemens. Stackable Tray System

Req. ID: GR3

Description: Support the component exchange impact analysis

Explanation: When an engineer wants to exchange a component, he must be aware of all the processes that need this componen, and of the impact that any change on the component has on these processes

Priority: Important.

Related Req. Ids: TLR4

Source UC: Siemens. Stackable Tray System

Req. ID: GR4

Description: Support the traceability among the modernisation steps

Explanation: All the time, the methodology has to facilitate to each actor of the modernization process (project manager, project developers, customer, etc.) a knowledge of the status of the progress in the modernization process

Priority: Important.

Related Req. Ids:

Source UC: Siemens/TID

Req. ID: GR5

Description: Support the traceability among Data, *ware and Process

Explanation: There should be the possibility to check mappings among entities in the TBMS and the modernized system.

Priority: Important.

Related Req. Ids: GR4

Source UC: Siemens/TID

Req. ID: GR6

Description: Support the definition and implementation of a transition plan: how the current system coexists with the modernized system

Explanation: In some moment, the old system and the modernized system have to coexist, unless to be sure that the new system behaves as expected and the users get acceptance. This transition plan is just a tool implementation of the transition plan methodology provided by Xirup methodology

Priority: Important.

Related Req. Ids: GR4

Source UC: Siemens/TID

Req. ID: GR7

Description: Iterate back to earlier steps in the modernisation process to adapt to newly discovered problems

Explanation: This is another feature of Xirup iterative methodology support by tools that allow users to go back and forth while modernizing

Priority: Important.

Related Req. Ids: GR4

Source UC: Siemens/TID

Req. ID: GR8

Description: Support the execution of the possible modernisation basic actions on data.

Explanation: Support the execution of the following possible modernisation basic actions on data:

• D_1 – understanding Data semantic;

• D_2 – add new elements;

• D_3 – transform existing elements and relation among them;

• D_4 – migrate database;

• D_5 – migrate data into database;

• D_6 – migrate data in homogeneous format;

• D_7 – create new documents/artefacts

• D_8 – update documents/artefacts

• D_9 link data to related document/artifacts

Priority: Important

Related Req. Ids:

Source UC: Siemens/TI

Req. ID: GR9

Description: Support the execution of the modernisation basic actions on *ware.

Explanation: Support the execution of the following modernisation basic actions on *ware:

• W_1 understanding components semantic

• W_2 transform Components

• W_3 analyse User Interface

• W_4 re-architect User Interface

• W_5 add new elements to user interface

• W_6 add new functionalities to and customize components

• W_7 add new components

Priority: Important

Related Req. Ids:

Source UC: Siemens/TID

Req. ID: GR10

Description: Support the execution of the modernisation basic actions on process

Explanation: Support the execution of the following modernisation basic actions on process:

• P_1 understanding Process semantic

• P_2 Add New Tasks
• P_3 Modify dependencies among tasks
• P_4 Modify existing tasks.
• P_5 Transform Process

Priority: Important.

Related Req. Ids:

Source UC: Siemens/TID

Req. ID: GR11

Description: Support for assessing the cost of the modernization. Support of Xirup metrics (metrics calculation over the model).

Explanation: The responsible of the modernization should be able to estimate (preliminary estimation based on generic parameters and post modelling estimation based on model transformation distance) how much does it cost the modernization of a legacy system towards a new system, with new functionality and/or adaptation of data or xWare.

Priority: Desirable.

Related Req. Ids: GR9

Source UC: TID

Req. ID: GR12

Description: Support for reusability of legacy test cases of the TBMS over the MS.

Explanation: When a test suites are available for the TBMS, these suites should be compatible to be reused (after being transformed) within the MS

Priority: Nice to have

Related Req. Ids:

Source UC: TID

Req. ID: GR13

Description: Support for reusability of configurations/scenarios of TBMS over the MS.

Explanation: When a configuration scenarios are available for the TBMS, these configuration scenarios should be compatible to be reused (after being transformed) within the MS.

Priority: Nice to have.

Related Req. Ids:

Source UC: TID

Req. ID: GR14

Description: Undo/Action historical Facilities

Explanation: Tools should support customized-depth undo facilities to move back and forth in the history of executed user actions, to allow users to recover wrong actions

Priority: Desirable

Related Req. Ids: GR7

Source UC: D11

Req. ID: GR15

Description: Support for control versioning

Explanation: Tools should provide versioning support to manage model versions, compare models, etc

Priority: Desirable

Related Req. Ids: GR7, GR4

Source UC: D11

Req. ID: GR16

Description: Tools should implement the Xirup methodology

Explanation: Tools should implement the Xirup methodology, templates, patterns, best practices, metrics, etc. assisted by wizards.

Priority: Essential

Related Req. Ids:

Source UC: D11

Req. ID: GR17

Description: Tools should support the generation and execution of regression tests on MS

Explanation: MS must pass all tests that passed the TBMS, except of changes in functionality.

Priority: Important

Related Req. Ids:

Source UC: D11

Req. ID: GR18

Description: Tools should support an iterative modernization process, as imposed by the Xirup methodology

Explanation: This iterative process implies to work continuously on the same model edition, back and forth model amendment, model loading and saving, as well as for other tasks as TBMS model extraction, model transformation, code generation, etc.

Priority: Important

Related Req. Ids: GR4

Source UC: D11

3.2 Requirement list for K-Xtraction Tool

Req. ID: XTR1

Description: Support to open/import UML diagrams (as XMI files) that describe the legacy system or system to be modernized (STBM). UML and XMI version supported to be decided.

Explanation: Momocs tools can open (or import) existing UML models (as XMI files) of TBMS. These models will be provided as external resources (not produced by Momocs tools). This functionality is complementary to the Momocs tools ability to manage their own UML models format. These models should be transformed according with Xirup metamodel

Priority: Essential.

Related Req. Ids:

Source UC: Siemens./TID

Req. ID: XTR2

Description: Reverse Engineering support. Support to extract a TBMS model from its implementation

Explanation: Reverse Engineering tools are needed in order to understand legacy systems (TBMS) and subsystems under modernisation. Momocs Tools support to import legacy documentation/implementation of the TBMS and their transformation into a Xirup model. Tools support structured documentation as software code, xmi (uml) models, xml, xml schema (xsd), etc. Tools also support behavioural and structural model extraction from TBMS and data model structure extraction for database, xml (structured sources) and non-structured sources.

Priority: Essential

Related Req. Ids:

Source UC: TID

Req. ID: XTR3

Description: Support for building the modernized data model from TBMS data model. That implies: data model schema extraction from TBMS, data model schema transformation, data model migration (data dumping and loading)

Explanation: The specification of the modernised data model can be supported by knowledge from existing data model. Also, the migration from existing to modernised data should be supported automatically or by some assistant.

Priority: Desirable.

Related Req. Ids:

Source UC: TID

3.3 Requirement list for XSM-Analysis Tool

Req. ID: ATR1

Description: Support the Xirup model edition

Explanation: Momocs tools manage Xirup models for the TBMS and MS (they can open/edit those models) so it is essential to integrate a Metamodel Editor (e.g. UML and domain specific editors), as part of the tools

Priority: Essential

Related Req. Ids: ATR7

Source UC: Siemens/TID

Req. ID: ATR2

Description: Support for refinement/modification at different levels of abstraction.

Explanation: Momocs tools allow working with different views of models, being able to visualize, query or apply transformations on the whole model, on specific diagrams, or concrete parts (as component description), at different levels of detail (zoom in/out).

Priority: Essential

Related Req. Ids: ATR3

Source UC: Siemens/TID

Req. ID: ATR3

Description: Deep modelling of components (types and instances). Support for component's properties specification.

Explanation: Momocs tools allow detailed and extended (more than permitted by UML?) modelling of components, describing their properties and their types, etc. For instance, they permit to describe the different component types and instances of the Stackable Tray System.

Priority: Essential

Related Req. Ids: ATR1, ATR2

Source UC: Siemens. Stackable Tray System

Req. ID: ATR4

Description: Checking of model consistency, and concretely, at level of model interfaces associations

Explanation: Momocs tools check the model consistency after any model change, at external level, ensuring the compatibility between component's interfaces that are linked by associations

Priority: Essential

Related Req. Ids: RETR4

Source UC: Siemens

Req. ID: ATR5

Description: Support of specific Xirup metamodel for component modelling (metamodel based on UML2 MOF metamodel, likely a UML profile)

Explanation: Supposed that XIRUP design a metamodel extended from UML2 (for instance, as UML profile) to specify component details, the Momocs Tools will support this metamodel

Priority: Important

Related Req. Ids: TTR4

Source UC: Siemens. Stackable Tray System

Req. ID: ATR6

Description: Support to group together and manage industrial components that contribute to create new components with specific functionality.

Explanation: In a bottom -up approach, first you define a set of Industrial Components and then group them by Functionalities. This features has to be supported by Xirup representation language (Xirup metamodel), for instance, by stereotyping

Priority: Essential

Related Req. Ids: TTR6

Source UC: Siemens

Req. ID: ATR7

Description: Support the characterisation of an industrial component.

Explanation: Support the characterisation of an industrial component with at least the following information: unique identifier, father component and a set of attributes

Priority: Essential

Related Req. Ids: TTR6

Source UC:

Req. ID: ATR8

Description: Support the identification in the model of instances of same industrial component type

Explanation: All the components of a model are instances of defined industrial component types. Thanks to this type-instance concept, it must be possible to identify all the components of the same type within a model. If you want to change a component type, you need a functionality to find in the model all instances of this component

Priority: Essential

Related Req. Ids: TLR3

Source UC:

Req. ID: ATR9

Description: Support the identification in the model of instances of same industrial component type.

Explanation: All the components of a model are instances of defined industrial component types. Thanks to this type-instance concept, it must be possible to identify all the components of the same type within a model. If you want to change a component type, you need a functionality to find in the model all instances of this component

Priority: Essential

Related Req. Ids: TTR6

Source UC: Siemens

Req. ID: ATR10

Description: Support the classification of industrial components defined by their specific properties

Explanation: Momocs tools permit to define criteria to classify industrial components on the basis of components properties and attributes

Priority: Essential

Related Req. Ids: RETR1

Source UC: Siemens

Req. ID: ATR11

Description: Support the attributes classification in accordance to engineering disciplines

Explanation: Attributes within a component can be classify or groupped.

Priority: Essential

Related Req. Ids: TLR6

Source UC: Siemens

Req. ID: ATR12

Description: Support the description of the model as (or pattern) in order to secure best-practice knowledge

Explanation: some models should be described in a generic way, i.e. with parameters that can be adapted although they are initialised with standard values that cover the majority of cases. We call them engineering patterns. Engineering patterns are, in this context, design patterns for industrial

systems engineering. A pattern can be a pre-defined and adaptable model for a complex system, or for a sub-part of it?

Priority: Essential

Related Req. Ids: TTR1

Source UC: Siemens

Req. ID: ATR13

Description: Support to analyse the structure and the behaviour of the TBMS. Support for static and dynamic diagrams modelling.

Explanation: Tools helping to understand the structure and the behaviour of the system to be modernised. Structure and behaviour should be expressed in Xirup Metamodel (UML compatible, that is, able to be opened by a UML case tool once the model is exported)

Priority: Essential

Related Req. Ids:

Source UC: TID/Siemens

Req. ID: ATR14

Description: Support to specify platform specific models of the modernised system (MS). That implies support to generate PSM diagrams describing the MS.

Explanation: This is in line to what MDA defines as PSM. MS models would be instantiations of Xirup metamodel

Priority: Important

Related Req. Ids:

Source UC: TID

Req. ID: ATR15

Description: Support creation, definition, browsing, querying, composition, adaptation of software components.

Explanation: XSM Analysis tool supports the complete design iterative cycle for a software component. This component model can be stored in a model repository for later use or modification.

Priority: Essential.

Related Req. Ids:

Source UC: TID

3.4 Requirement list for Constraint Editor Tool

Req. ID: CETR1

Description: Support to specify internal dependencies between component's properties

Explanation: Momocs tools allow to specify dependencies, constraints (possibly using OCL or similar XIRUP language) between component's properties, to ensure internal consistency.

Priority: Essential

Related Req. Ids:

Source UC: Siemens

Req. ID: CETR2

Description: Support to specify external dependencies between components, through components' interfaces

Explanation: Momocs tools allow to specify dependencies, constraints (possibly using OCL or similar XIRUP language) between component's interfaces, to ensure external consistency.

Priority: Essential

Related Req. Ids:

Source UC: Siemens

Req. ID: CETR3

Description: Support to specify rules and constraints in component model

Explanation: Momocs tools allow to specify rules, constraints (possibly using OCL or similar XIRUP language) in components model

Priority: Essential

Related Req. Ids:

Source UC: Siemens

3.5 Requirement list for XSM-Visualization Tool

Req. ID: VTR1

Description: User friendly interface for model creation/edition

Explanation: Tools include a GUI Xirup Editor with facilities as drag and drop, cut and paste, forms, wizards, etc to assist the user when the model is been editing.

Priority: Important.

Related Req. Ids:

Source UC: Siemens

Req. ID: VTR2

Description: Support to produce different views of the model

Explanation: Momocs tools permit to produce different views of the same model diagram. These views are customized by the user assisted by wizards

Priority: Essential

Related Req. Ids: VTR3

Source UC: Siemens

Req. ID: VTR3

Description: Support to zoom in/out views of the model

Explanation: Momocs tools permit to zoom in/out (different level of details) into the views of a model diagram. The level of detail at each zoom scale can be customized.

Priority: Essential

Related Req. Ids:

Source UC: Siemens

Req. ID: VTR4

Description: Support for additional non-UML graphical representation of models (based on more user friendly graphical representations)

Explanation: UML graphical language is complex to understand by non experts, hence, a more readable graphical language should be supported by Momocs tools to allow model understanding by non OO Modeling experts.

Priority: Important

Related Req. Ids: ATR13

Source UC: Siemens

3.6 Requirement list for XSM Transformation Tool

Req. ID: TTR1

Description: Configuration tool for transforming of the PIM into one or two different PSMs.

Explanation: This is in line with MDA transformation pattern. But in this case there is an additional issue: it is not a transformation from one PIM to one PSM, but some components of one PIM are transformed into one PSM and others into other PSM

Priority: Desirable.

Related Req. Ids:

Source UC: TID

Req. ID: TTR2

Description: Supporting languages and tools to specify transformations that take into account: existing system, modernisation requirements and SP framework and components. Tools support model transformation specification and running, receiving as inputs the TBMS, requirements set, additional models (SP framework and components)

Explanation: Transformations are the key to automate modernization. Tools are needed to perform automatic translation of data, xWare and process semantics. These models transformation will be semi-automatic, computer assisted, from TBMS PIM to MS PSM.

Priority: Essential

Related Req. Ids:

Source UC: TID

Req. ID: TTR3

Description: Support to compare and track changes from existing to modernized system. Model mapping (between TBMS and MS)

Explanation: The developer can see what elements in the modernized system correspond to elements in the legacy system and vice versa. That implies a mapping between both models. This mapping is just another view of the model transformation.

Priority: Important

Related Req. Ids:

Source UC: TID

Req. ID: TTR4

Description: PIM-PSM model synchronization

Explanation: In case of PIM-PSM transformations, when it is detected errors in the PSM models due to errors in PIM models, it is desirable that when editing the PSM model to amend that errors, then automatically they are reflected in the PIM model. A kind of automatic reverse transformation for model-model synchronization

Priority: Nice to have.

Related Req. Ids:

Source UC: TID

Req. ID: TTR5

Description: Tools should support model transformation specification and execution

Explanation: Tools will permit to write model transformation from TBMS metamodel into the MS metamodel, provided that the tools are fed with the necessary inputs.

Priority: Essential

Related Req. Ids:

Source UC: D11

Req. ID: TTR6

Description: Tools should support MS code generation (from MS PSM) and deployment

Explanation: This is just a particular case of model transformation from PSM to PSM (code)

Priority: Important

Related Req. Ids:

Source UC: D11

Req. ID: TTR7

Description: There should be the possibility to define composition of transformations.

Explanation: Simple transformations could be combined to define more complex transformations. The iterative nature of the Xirup process could be supported by this feature, which allows to create more complex transformations incrementally.

Priority: Important

Related Req. Ids:

Source UC: TID

3.7 Requirement list for XSM Type Library and Frontend Tool

Req. ID: TLR1

Description: Support for component type selection from a component type library

Explanation: Momocs tools use a library repository of components types. Users can browse this repository and select concrete component types to be included into the model under development.

Priority: Essential

Related Req. Ids: TLR6

Source UC: Siemens/TID

Req. ID: TRL2

Description: Support for component type library management (insertion, searching, deletion, etc).

Explanation: Momocs tools permit to manage the component library repository, inserting new component types, editing exiting types, searching for types, deleting types, etc.

Priority: Essential

Related Req. Ids:

Source UC: Siemens/TID

Req. ID: TLR3

Description: Support the hierarchical organisation of component types and patterns within the library

Explanation: Support for different (tree-based, package based, etc.) hierarchical organizations

Priority: Desirable

Related Req. Ids: GR14

Source UC: Siemens/TID

Req. ID: TLR4

Description: When exchanging an industrial component, the model must identify suitable (based on the dependencies) other industrial component types within the component type library

Explanation: On a given abstraction layer (see refinement concept), I can select a component to be exchanged. Based on the dependencies, the methodology can identify all other component types from my component type library that could match. Additionally, there could be preferences (e.g. on the price or the manufacturer) to influence the choice

Priority: Essential

Related Req. Ids: NFR10

Source UC: Siemens/TID

Req. ID: TLR5

Description: Supporting tools for the reuse of components and patterns for modernization. Extensive use of the Components and Patterns Repository

Explanation: Reuse can improve robustness and performance of the modernization process. This requires support to manage a library of components and patterns for modernization. Management tools should facilitate component type and pattern specification, searching, browsing, modification, adaptation and implementation of components, data and patterns (models in general)

Priority: Essential

Related Req. Ids: TTR5

Source UC: TID

Req. ID: TLR6

Description: Tools should support model repository querying and retrieval

Explanation: Users can produce complex queries to seek for models from the model repository. Users can browse the results and retrieve the model they have selected

Priority: Essential

Related Req. Ids:

Source UC: D11

3.8 Requirement list for Requirement Elicitation Tool

Req. ID: RTER1

Description: Support for requirement elicitation. Formal specification of requirements

Explanation: There should be a tool to specify and manage requirements. This needs further elaboration by Xirup methodology.

Priority: Important

Related Req. Ids:

Source UC: Siemens/TID

Req. ID: RETR2

Description: Tools should provide support to elicit functional and non functional requirements.

Explanation: Tools should provide support to elicit functional and non functional requirements assisting the user by a set of forms and/or wizards. Requirements will be described using the appropriate Xirup methodology and/or pattern

Priority: Important

Related Req. Ids:

Source UC: D11

Req. ID: RETR3

Description: Support for model's requirement attachments

Explanation: Functional and non functional requirements are stored within a requirements repository and/or use as an attachment to the TBMS model

Priority: Desirable

Related Req. Ids:

Source UC: D11

Req. ID: RTET4

Description: Support for a requirement repository management

Explanation: Tools should manage the requirements repository (classification, storage, retrieval, modification of requirements, etc)

Priority: Desirable

Related Req. Ids:

Source UC: D11

4 Other non-functional requirements

This section elaborates non-functional requirements common for all tools. These requirements are recommendations to use common best practices for insuring interoperability, reliability and quality of the software delivery.

4.1 Interface Requirements

Req. ID: NFR1

Description: Models should be encoded in XMI format (MOF XMI 2.1 and its particular subset EMF XMI)

Explanation: Interoperability between tools requires that tools shares the same standard format for data exchange. EMF XMI (well defined subset of MOF XMI 2.1) is the most widely used format compatible with OMG MOF XMI.

Priority: Essential

Related Req. Ids:

Source UC:

Req. ID: NFR2

Description: Open Standards should be privileged.

Explanation: Usage of open standards (data, models, interfaces, tests etc) should be privileged in order to ensure maintainability and wide adoption of developed solutions.

Priority: Essential

Related Req. Ids:

Source UC:

4.2 Documentation Requirements

Req. ID: NFR3

Description: Tools should be documented with appropriate level of details

Explanation: To be adopted by large community of users, an explicit documentation of tools is required. The following documentation should be provided along with tool releases:

- Release Notes
- Installation Instructions
- User's Guide - Tutorial
- Source Code Documentation (if applicable)

Priority: Essential

Related Req. Ids:

Source UC:

4.3 Quality and Safety Assurance Requirements

Req. ID: NFR4

Description: Quality insurance requires Bug tracking system

Explanation: Bug tracking system should be deployed to keep trace of the problems and their resolution.

Priority: Essential

Related Req. Ids:

Source UC:

Req. ID: NFR5

Description: Quality insurance requires Unit Testing

Explanation: Quality insurance process requires Unit Tests to be provided along with tools releases.

Priority: Essential

Related Req. Ids:

Source UC:

Req. ID: NFR6

Description: Tools should be robust (do not crash)

Explanation: Momocs tools should reach a reasonable stability (for a prototype), therefore, they should be enough mature and stable, to allow users a normal use of their functionality.

Priority: Essential

Related Req. Ids:

Source UC: TID

4.4 Security

Req. ID: NFR7

Description: Ensure non-corruption of entry data for all the tools.

Explanation: Models are essentially important data. Tools should not corrupt data (models) provided as inputs

Priority: Essential

Related Req. Ids:

Source UC:

4.5 Architecture

Req. ID: NFR8

Description: Plugin modular design. Support based on plugins/adapters to use different modelling languages and technologies

Explanation: Tools design should be modular, adapter-based or plugin-based to allow the use of different modelling languages, representation languages. Tools should also be technology agnostic in the sense that they can not impose the adoption of any particular technology when modernizing the TBMS.

Priority: Important

Related Req. Ids:

Source UC: D11

Req. ID: NFR9

Description: Momocs tools are seamlessly integrated

Explanation: All momocs tools should be integrated each other so, from the user perspective, Momocs tools should behave as only one tool, the Momocs suite. Not only the Momocs tools will appear as a seamlessly suite, but all them will collaborate each other transparently to provide the final user requests. Tools architecture will be oriented to achieve that goal

Priority: Essential

Related Req. Ids: NFR8

Source UC:

Req. ID: NFR10

Description: Tools should be multiplatform

Explanation: Tools should be potentially executed in any Java compliant underlying platform: i.e. Windows OS, Linux/Unix, etc

Priority: Important

Related Req. Ids:

Source UC: D11

5 Conclusions

This document has collected a wide bundle of Momocs tools' functional and non-functional requirements, most of them directly derived from an analysis of the Momocs case studies. This requirement analysis has also permitted us to envisage a preliminary set of possible Momocs tools. This tools discovery has allowed us to classify the requirements in groups assigned to each individual tool. Requirements have been also scored with a priority flag that will be used for Momocs team to determine the importance of each requirement during the design and development phases of the tools. However, the real importance of this study is that it will help us to make the functional and technical design of the tools and its later development.