D2.2.a - Specification of needed views

WP2.2 - Concrete Syntax and Tool Development

Abstract: This deliverable explains different views identified by the stakeholders of FORMOSE project. These views are identified by closely studying various needs of the clients in an industrial setup, within the bounds of this project. The requirements and expectations of a typical stakeholder in requirements engineering is analyzed through the discussions with industrial partners in this consortium. Each view corresponds to a specific set of activities from the global process of requirements engineering defined through this project. Each of these specified views will be implemented within OpenFlexo.

Status: Public / ~~Confidential~~
Version: Draft / ~~Review~~ / ~~Final~~

Version history

Version	Date	Contributors	Contribution
1.0	09-03-2017	Fahad R. Golra	Drafting the specification document.
1.1	XX-03-2017	Fabien Dagnat	Corrections and Internal review.

Contents

List of Figures

1 Scope

1.1 Identification

Document Name:	Specification of needed views
Deliverable Identification Number:	D2.2.a
Project Name:	ANR Formose
File Name:	Formose-D2.2.a

1.2 System overview

FORMOSE requirements engineering methodology focuses on elicitation, specification, modeling and validation of the requirements for critical and complex systems. These requirements gathered for the critical system might be related to the product, its context, its utilization process or the development process of this product. The philosophy behind FORMOSE methodology is to gather the requirements related to all project elements and then link to them to the appropriate views for carrying out different requirements engineering activities. Some of the views are used for requirements elicitation, specification and modeling. Other views are used for the validation of requirements models, the context of the product or process, domain concepts and terminologies. All these views, shown in Figure 1, are used together to present a multi-view approach for requirements engineering.

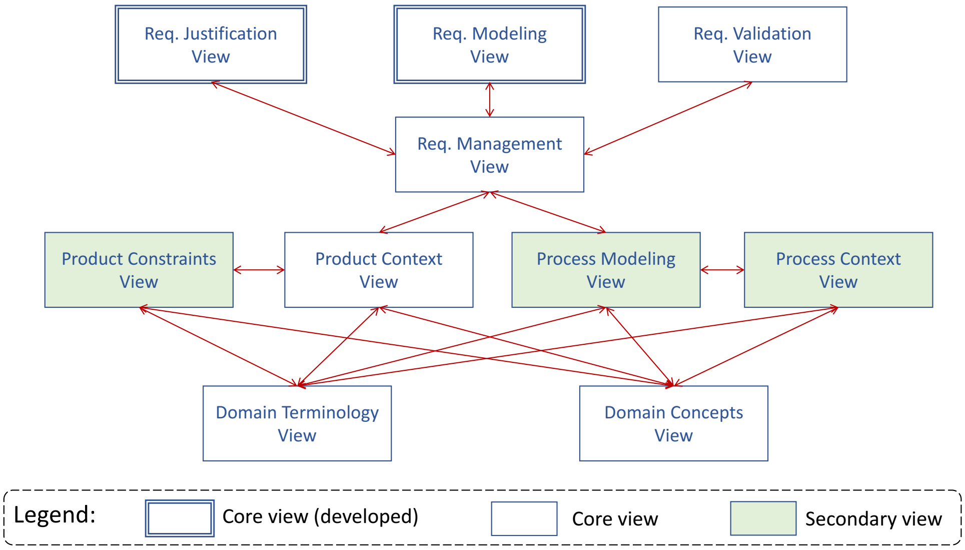


Figure 1: Views in FORMOSE

The views (to be) developed under FORMOSE project can be categorized into two parts: for product and for process. Many of the views are used for both product and process alike. For example, the *Requirements Justification View*, *Requirements Modeling View*, *Requirements Validation View* and the *Requirements Management View*. The *Domain Terminology View* and the *Domain Concepts View* are also used for both product and process.

1.3 Document overview

This document describes the requirements of the tool support (FORMOD) for the FORMOSE requirements engineering methodology. It has been organized according to the views presented to the end user. The views are categorized in *'core'* and *'secondary'*, where core views are considered mandatory for FORMOD and secondary views are optional. Each view is presented by a general description and then the related requirements are specified for it. The document presents the following views:

Core views

- Requirements Justification View
- Requirements Modeling View
- Requirements Management View
- Requirements Validation View
- Domain Terminology View
- Domain Concepts View
- Product Context View

Secondary views

- Process Modeling View
- Process Context View
- Product Constraints View

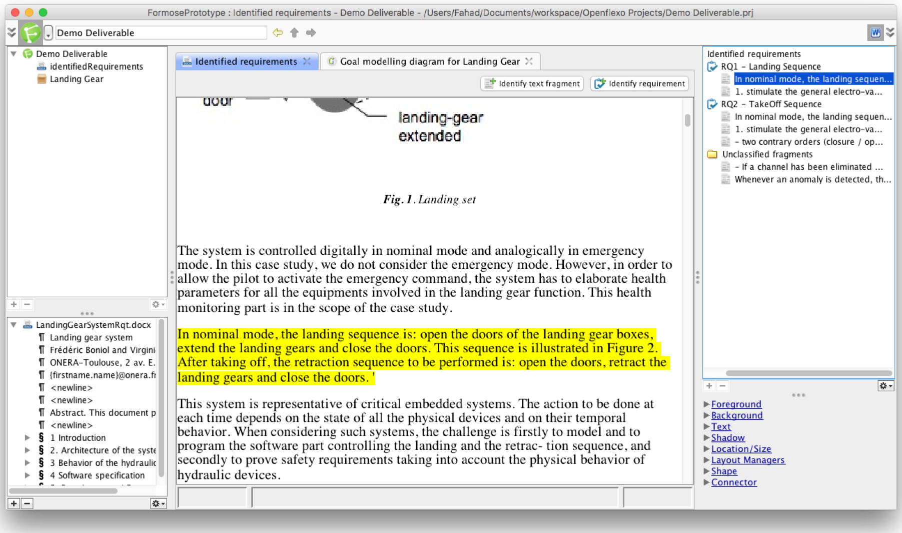
2 View Specification (Core Implemented Elements)

Out of the core views, two of them are described in this section. Both these views are already developed.

2.1 Requirements Justification View

2.1.1 General Description

Requirements justification view presents an interface, where the users can elicit requirements from some given source documents from the domain. These documents might be internal standards, minutes of the meetings, feasibility reports, legacy system documentation, *etc.* This view can also be used when some requirements are already specified, and they need to be traced back to their sources *i.e.* providing a justification of a requirement. An initial implementation of this view has already been done, as shown in Figure 2 .



FormosePrototype : Identified requirements - Demo Deliverable - /Users/Fahad/Documents/workspace/Openflexo Projects/Demo Deliverable.prj

Demo Deliverable
- Demo Deliverable
  - identifiedRequirements
  - Landing Gear

- LandingGearSystemRqt.docx
  - ¶ Landing gear system
  - ¶ Frédéric Boniol and Virgini
  - ¶ ONERA-Toulouse, 2 av. E.
  - ¶ {firstname.name}@onera.fr
  - ¶ <newline>
  - ¶ <newline>
  - ¶ Abstract. This document p
  - ¶ <newline>
  - § 1 Introduction
  - § 2. Architecture of the syste
  - § 3 Behavior of the hydraulic
  - § 4 Software specification

Fig. 1. Landing set

The system is controlled digitally in nominal mode and analogically in emergency mode. In this case study, we do not consider the emergency mode. However, in order to allow the pilot to activate the emergency command, the system has to elaborate health parameters for all the equipments involved in the landing gear function. This health monitoring part is in the scope of the case study.

In nominal mode, the landing sequence is: open the doors of the landing gear boxes, extend the landing gears and close the doors. This sequence is illustrated in Figure 2. After taking off, the retraction sequence to be performed is: open the doors, retract the landing gears and close the doors. '

This system is representative of critical embedded systems. The action to be done at each time depends on the state of all the physical devices and on their temporal behavior. When considering such systems, the challenge is firstly to model and to program the software part controlling the landing and the retrac- tion sequence, and secondly to prove safety requirements taking into account the physical behavior of hydraulic devices.

Identified requirements
- RQ1 - Landing Sequence
  - In nominal mode, the landing sequen...
  - 1. stimulate the general electro-va...
- RQ2 - TakeOff Sequence
  - In nominal mode, the landing sequen...
  - 1. stimulate the general electro-va...
  - - two contrary orders (closure / op...
- Unclassified fragments
  - - If a channel has been eliminated ...
  - Whenever an anomaly is detected, th...

Figure 2: Requirements Justification View

2.1.2 Requirements

[FOR-COR-01]

The requirements justification view shall provide the possibility to load domain documents. For the purpose of this prototype, MS Word documents and MS Excel Spreadsheets could be loaded into the view. However, an extension point should be defined to load other types of documents. This extension point should serve for future extensions of this tool.

[FOR-COR-02]

The requirements justification view shall provide basic functions like selecting and copying text from the loaded documents.

[FOR-COR-03]

The requirements justification view shall provide the ability to select a fragment of text from the loaded document and display a pointer to access this chosen text fragment.

[FOR-COR-04]

Upon selecting a text fragment, the pointer to the text fragment should be accessible to the user from the graphical view. All the pointers to selected text fragments should be organized in a way that user can perform further actions on those text fragments, using these pointers.

[FOR-COR-05]

The user shall have the possibility to view all the specified requirements in the requirements justification view.

[FOR-COR-06]

The user shall be able to identify a requirement either from an already selected fragment available through the pointer or from a portion of text selected in the loaded document. When the requirement is identified through a selected portion of text in the loaded document, a pointer to this selected text shall be kept as an identified fragment.

[FOR-COR-07]

The requirements justification view shall display the association of each requirement to the identified text fragments.

[FOR-COR-08]

The user shall be able to link the already available requirements to already identified text fragments or to portions of text from the loaded document. In the later case, the portion of text will become identified requirement and a point to it will be kept with other pointers.

2.2 Requirements Modeling View

2.2.1 General Description

Requirements modeling view presents an interface, where the users can model requirements. This view allows modeling the already existing requirements specifications. It also allows to refine/abstract the requirements, hence allowing to add further requirements to the specification. The view presents as a requirements modeling editor. The interface of the initial implementation of this view is depicted in Figure 3.

2.2.2 Requirements

[FOR-COR-09]

The requirements modeling view shall allow the development of requirement models. These requirement models need to be graphical. Hence this view shall serve as a graphical requirements modeling editor.

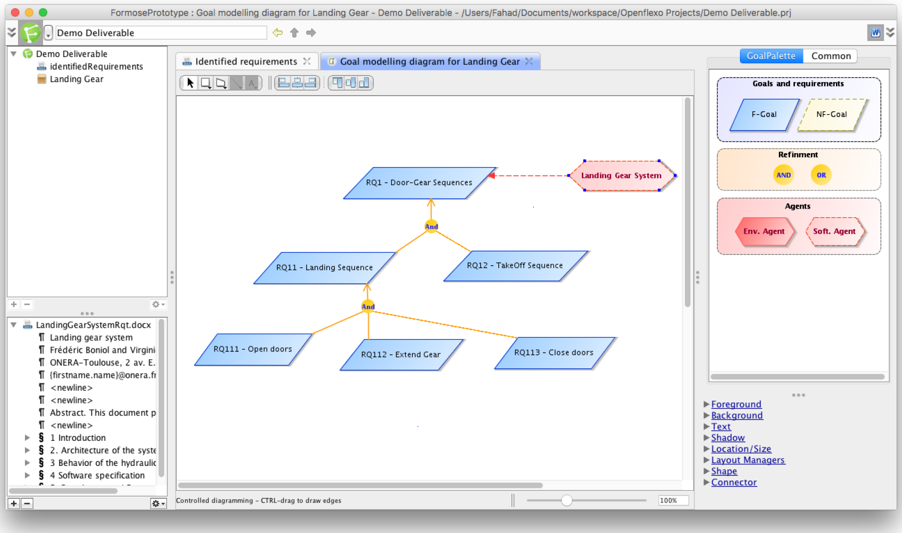


Figure 3: Requirements Modeling View

[FOR-COR-10]

As other graphical modeling editors, the requirements modeling view shall provide a main drawing area where graphical models can be developed. A panel of graphical shapes that correspond to the underlying concepts of requirements language shall be shown to the users in this view. Users should have the possibility of drag and drop different graphical shapes from this panel to the main drawing area.

[FOR-COR-11]

The graphical modeling view shall provide the possibility to alter the graphical properties of the shapes used for developing requirement models. This will help users in following the free modeling philosophy [1] for their requirement models.

[FOR-COR-12]

It shall be possible for the user to access the already specified requirement in the requirements modeling view. This shall allow users to model already specified requirements.

[FOR-COR-13]

Requirements modeling view shall offer the possibility to push the modeled requirements to the specified requirements. This way the requirements added to the model as a result of refinement or abstraction would be synchronized with the ones specified in the requirements management view.

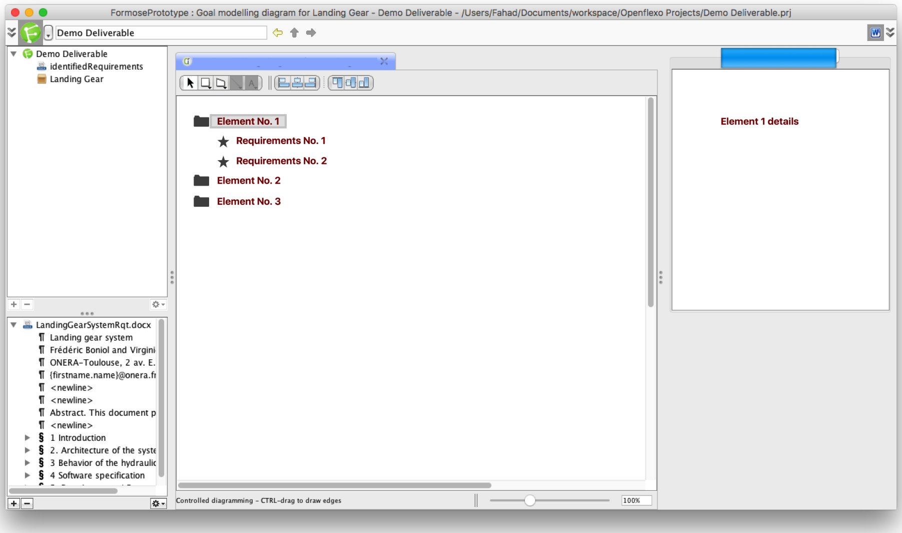
3 View Specification (Core Elements)

Core elements of this specification concern the views that are considered mandatory for the tool under development. The following core elements are not developed yet.

3.1 Requirements Management View

3.1.1 General Description

Requirements management view presents all the specified requirements of the system. These requirements can be categorized according to the elements associated to these requirements. An elements may be an process elements, project elements, system component, agent, *etc.* A rough sketch of the possible interface for this view is depicted in Figure 4. The central area of the interface is used for listing the requirements. Different panes at the sides can be used for interacting with the user. These panes can also be used to inspect the details of a requirement or an element.


FormosePrototype : Goal modelling diagram for Landing Gear - Demo Deliverable - /Users/Fahad/Documents/workspace/Openflexo Projects/Demo Deliverable.prj

Controlled diagramming - CTRL-drag to draw edges

Figure 4: Requirements Management View

3.1.2 Requirements

[FOR-COR-14]
Requirements management view shall serve as a central view of the tool, where all the specified requirements can be consulted.

[FOR-COR-15]
The user shall be able to export the specified requirements from the requirements management view in a serialized version. This will help importing the requirements into doors or rectify, in case needed.

[FOR-COR-16]
The user shall be able to link the requirements to different project elements. This shall allow requirements to be displayed as linked to the decomposition of the project elements.

3.2 Requirements Validation View

3.2.1 General Description

Requirements validation view presents the specified requirements of the system and the corresponding formalisms in B-method. This view might have one or two different interfaces. It will be used to graphically view which parts of the model are validated and which parts are not. This view shall also allow interacting with the formal specification of the requirements.

3.2.2 Requirements

[FOR-COR-17]
The requirements validation view will link the requirements to B solver. This will be used to check if the refinement graph of the requirements does not create any unreachable requirements. This view shall also help in identifying the requirements conflict to some extent.

[FOR-COR-18]
Requirements validation view shall show the validated and non validated parts of the SysML kaos modeling diagram. This validated and non-validated parts of the model will be displayed in a graphical form i.e. color codes or graphical icons shall be used, so that the user can easily identify non-validated parts of the model.

[FOR-COR-19]
The graphical user interface used for the requirements validation view shall display the corresponding B source code for the requirements model in question. A selection mechanism for highlighting the corresponding code for an element in requirements model shall optionally be added to the graphical interface.

[FOR-COR-20]
The solver should not run continuously for each change made to the requirements model. The requirements validation view shall provide the possibility to start the validation process for the requirements model.

[FOR-COR-21]
The results returned by the solver should be displayed to the user through the requirements validation view. Part of the results will be shown through graphical interface (FOR-COR-18), and the rest of them should be consultable in textual form for details reasoning.

[FOR-COR-22 / Optional]
The requirements validation view shall provide the possibility to edit the B code directly from the provided interface.

[FOR-COR-23 / Optional]
It should be possible for the user to edit the requirements model directly from the requirements validation view.

3.3 Domain Terminology View

3.3.1 General Description

This view shall present an interface where a user can interact with the ontologies. Concepts of the ontologies can be viewed and edited in this view. The interface will load the ontologies related to the terminologies used in the development project.

3.3.2 Requirements

[FOR-COR-24]
The domain terminology view shall allow the user to develop a knowledge base of terminologies involved in critical and complex software development projects. This knowledge base shall be developed using ontologies.

[FOR-COR-25]
The user shall be able to load the existing ontologies into the tool, using domain terminology view. The user shall also have the possibility to edit the ontologies directly from this view.

3.4 Domain Concepts View

3.4.1 General Description

This view shall present an interface where a user can interact with the ontologies. Concepts of the ontologies can be viewed and edited in this view. The interface will be used for the ontologies of the concepts used in the development project.

3.4.2 Requirements

[FOR-COR-26]
The domain concepts view shall allow the user to develop a knowledge base of the concepts involved in critical and complex software development projects. This knowledge base shall be developed using ontologies.

[FOR-COR-27]
The user shall be able to load the existing ontologies into the tool, using domain concepts view. The user shall also have the possibility to edit the ontologies directly from this view.

3.5 Product Context View

3.5.1 General Description

Product context view presents an interface to develop and edit the context diagrams for the system under development. It presents a graphical editor for the development of these models. The interface will have a palette of concepts and a main drawing area for the development of models, as shown in Figure 5.

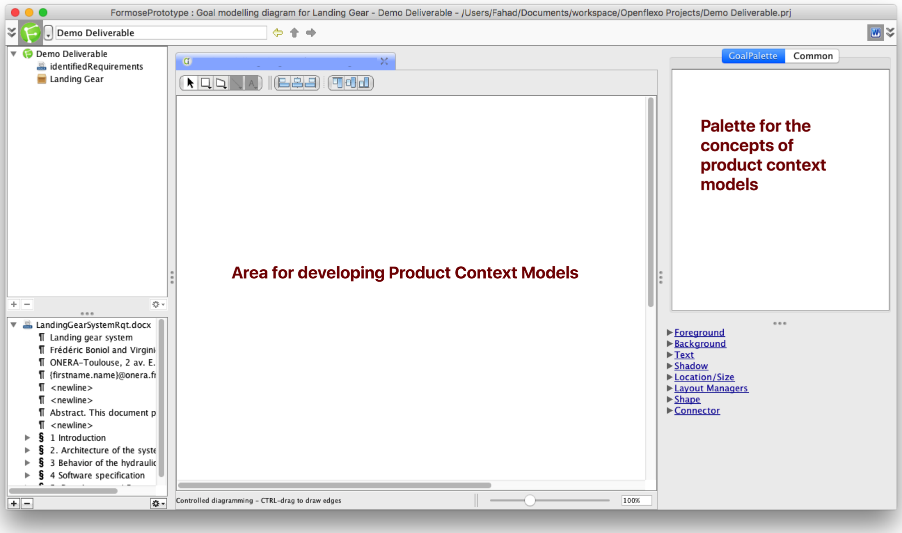

FormosePrototype : Goal modelling diagram for Landing Gear - Demo Deliverable - /Users/Fahad/Documents/workspace/Openflexo Projects/Demo Deliverable.prj
Demo Deliverable
Demo Deliverable
identifiedRequirements
Landing Gear
LandingGearSystemRqt.docx
Landing gear system
Frédéric Boniol and Virgini
ONERA–Toulouse, 2 av. E.
{firstname.name}@onera.fr
<newline>
<newline>
Abstract. This document p
<newline>
§ 1 Introduction
§ 2. Architecture of the syste
§ 3 Behavior of the hydraulic
§ 4 Software specification
Area for developing Product Context Models
Controlled diagramming – CTRL-drag to draw edges
100%
GoalPalette
Common
Palette for the concepts of product context models
Foreground
Background
Text
Shadow
Location/Size
Layout Managers
Shape
Connector

Figure 5: Product Context View

3.5.2 Requirements

[FOR-COR-28]

The product context view shall provide an editor for the development of product context models. These product context models are graphical. Hence this view shall serve as a graphical product context modeling editor.

[FOR-COR-29]

The product context models use the formalisms of SysML block diagram. Thus, the graphical editor offered by this view shall follow the metamodel of SysML block diagrams.

[FOR-COR-30]

The product context view shall be able to access the domain concepts and terminologies for the development of product context models. These concepts and terminologies will be available to the user within the product context view.

[FOR-COR-31]

The product context modeling view shall provide a main drawing area where graphical models can be developed. A panel composed of graphical shapes correspond to the concepts of product context metamodel shall be available to the users. Users should have the possibility of drag and drop different graphical shapes from this panel to the main drawing area.

[FOR-COR-32]

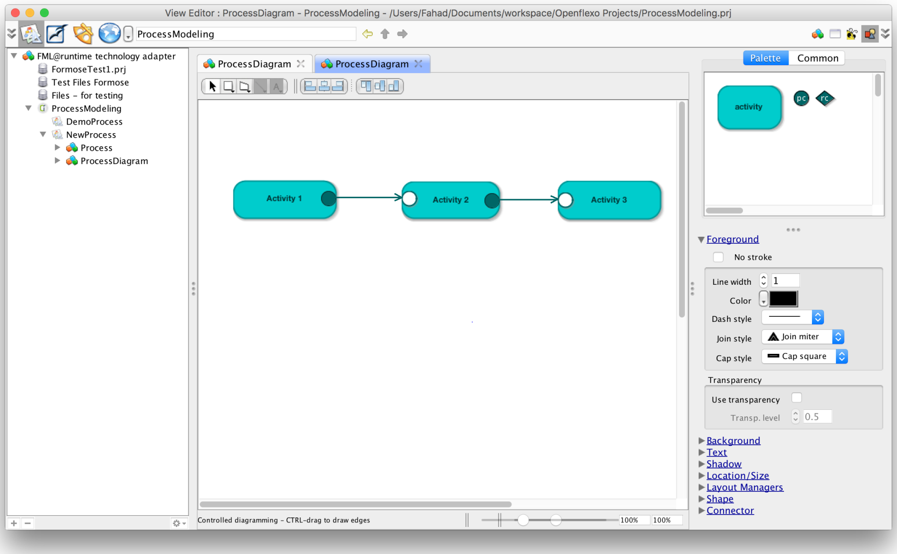
The product context view shall provide the possibility to edit the graphical properties of the shapes used for developing the product context models.

4 View Specification (Secondary Elements)

4.1 Process Modeling View

4.1.1 General Description

Process modeling view is used for developing process models for the development of the concerned system under development. The interface of this view is a graphical editor for the process modeling language defined by the methodology. The graphical editor will have a palette for process model concepts and a main drawing area for the development of process models, as shown in Figure 6.



View Editor : ProcessDiagram - ProcessModeling - /Users/Fahad/Documents/workspace/Openflexo Projects/ProcessModeling.prj

Controlled diagramming - CTRL-drag to draw edges

No stroke □
Line width 1
Join style Join miter
Cap style Cap square
Use transparency □
Transp. level 0.5

Figure 6: Process Modeling View

4.1.2 Requirements

[FOR-SEC-33]

The process modeling view shall serve for process specification. The processes will be specified using graphical models. This view shall provide a graphical editor for the development of graphical process models.

[FOR-SEC-34]

The process modeling view shall provide the possibility to access the key activities performed by the FORMOSE tool. These activities have defined inputs and outputs, which shall be made visible to the user.

[FOR-SEC-35]

A user shall be able to develop a process model using the activities already available. This process model shall define the workflow followed the tool for a specific project.

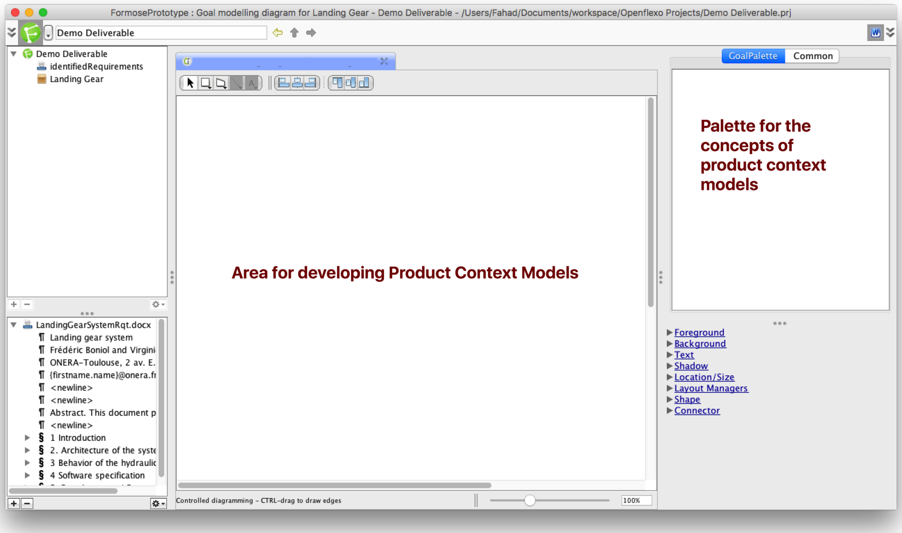
[FOR-SEC-36]

The process model developed by the user shall define the methodology and views used for a specific project. Hence, different views of the FORMOSE tool will be configured according to this process model.

4.2 Process Context View

4.2.1 General Description

Process context view presents an interface to develop and edit the context diagrams for the development process used in a project. It presents a graphical editor for the development of these models. The interface will have a palette of concepts and a main drawing area for the development of models, as shown in Figure 7.



FormosePrototype : Goal modelling diagram for Landing Gear - Demo Deliverable - /Users/Fahad/Documents/workspace/Openflexo Projects/Demo Deliverable.prj

Demo Deliverable

- Demo Deliverable
  - identifiedRequirements
  - Landing Gear

- LandingGearSystemRqt.docx
  - ¶ Landing gear system
  - ¶ Frédéric Boniol and Virgini
  - ¶ ONERA-Toulouse, 2 av. E.
  - ¶ {firstname.name}@onera.fr
  - ¶ <newline>
  - ¶ <newline>
  - ¶ Abstract. This document p
  - ¶ <newline>
  - § 1 Introduction
  - § 2. Architecture of the syste
  - § 3 Behavior of the hydraulic
  - § 4 Software specification

Area for developing Product Context Models

Controlled diagramming - CTRL-drag to draw edges

Palette for the concepts of product context models

Figure 7: Process Context View

4.2.2 Requirements

[FOR-SEC-37]

The process context view shall allow the development of graphical models for the actors that participate in the development product models. This view shall provide a graphical editor for the development of graphical process context models.

[FOR-SEC-38]

The process context view shall provide the possibility to access the concept and terminology ontologies specified in the domain concepts view and the domain terminology view.

4.3 Product Constraints View

4.3.1 General Description

Product constraints view presents an interface to interact with the constraints related to the product under development. These constraints can be taken from different sources of information.

4.3.2 Requirements

[FOR-SEC-39]
The product constraints view shall offer the possibility to model the domain constraints. These constraints can be the product standards to be followed, the rules and regulations for the product, etc.

[FOR-SEC-40]
The product constraints view shall allow linking these constraints to individual specified requirements managed by the requirements management view. Each constraint be traceable from the requirements management view.

Views	Product		Process	
	specification	domain	specification	domain
Requirements Justification View	x	x	x	x
Requirements Modeling View	x	x	x	x
Requirements Management View	x	x	x	x
Requirements Validation View	x	x	x	x
Domain Terminology View	x	x	x	x
Domain Concepts View	x	x	x	x
Product Context View		x		
Process Modeling View			x	
Process Context View				x
Product Constraints View	x	x		

Table 1: Relationship of views to project elements

5 Summary

In this document, we outlined the requirements specifications for the tool support to be implemented under the ANR project *FORMOSE*. This tool is the implementation of the *FORMOSE* requirements engineering methodology for critical and complex systems. This methodology follows a multi-view approach for requirements engineering. Requirements of the tool are presented according to the different views that are either already developed or are planned to be developed. These views are categorized into core and secondary views. The core views shall be implemented in this project on a priority basis. The secondary views are the perspectives which might or might not make into the final implementation, depending on the availability of time at hand.

The different views specified in this document concern a typical development project of a critical and complex software and systems. Table 1 categorizes the project elements into product and process elements. Each of these elements are further categorized into specification elements or the domain elements. Specification elements concern the core software product or process, whereas domain elements concern the domain or context of these core elements. The matrix presented in this table relates each view to this project element decomposition.

References

[1] Fahad R Golra, Antoine Beugnard, Fabien Dagnat, Sylvain Guerin, and Christophe Guychard. Using free modeling as an agile method for developing domain specific modeling languages. In *Proceedings of the ACM/IEEE 19th International Conference on Model Driven Engineering Languages and Systems*, pages 24–34. ACM, 2016.