



SCENARIO - DOMAIN REQUIREMENTS ENGINEER

DOMAIN REQUIREMENTS ENGINEER

This perspective should analyze and specify the variable and common requirements of the product portfolio in order to facilitate the vision of the application that will be developed for different stakeholders. This view is given by the use case, class, activity and sequence diagrams that together define the functionalities, exchange of messages and activities of the systems.

To diagrams correctly express the user requirements without inconsistencies, you must review such diagrams and elements. To achieve your goal, perform the steps outlined below to inspect each of the informed diagrams. When you find a defect in one of the steps, fill in the Defect Identification Form indicating the diagram, the step item (number question), and the element and the defect found.

LOCATE USE CASE DIAGRAM AND REQUIREMENTS SPECIFICATION

Step 1

inspection of use case diagram

Step 2

LOCATE CLASS DIAGRAM AND REQUIREMENTS SPECIFICATION

Step 3

inspection of class diagram

Step 4

LOCATE CLASS DIAGRAM AND REQUIREMENTS SPECIFICATION

Step 5

Step 6

Step 7

Step 8

Inspection of sequence diagram



LOCATE USE CASE DIAGRAM AND REQUIREMENTS SPECIFICATION

The use case diagram in Domain Engineering must correctly describe the set of features of all systems that can be configured from SPL. Read the requirements specification carefully. While reading, make a list of all the requirements and actors specified in the document, placing a notation with the type of stereotype of the element (mandatory, optional, etc.) and their possible relationships. Compare the list made with the use case diagram to ensure that there is no inconsistency between them. To do this, answer the questions that follow.

Step 1	Consider actors and use cases as elements in this step. For each element in the use-case diagram, check for matching elements in the list made. Mark the element after its analysis, to prevent it from being analyzed again.	
	1.1	Does the name of the element correctly express functionality?
	1.2	Does this element correspond to a feature/actor that was not defined in the requirements document? If so, disregard him and his relationships for the next steps and move on to the next element.
	1.3	Has the functionality described by that element already been specified by another element? If so, disregard the element that is incorrect and its relationships to the next steps and move on to the next element.
	1.4	Is the element in the use-case diagram stereotyped?
	1.5	In the use case diagram, if the element is optional or mandatory, has it been specified with the correct stereotype (<<optional>> or <<mandatory>>)?
	1.6	Was the element that represents a point of variation marked with the <<variationPoint>> stereotype?
	1.7	For each element of the use case diagram, check and analyze their relationships to answer the questions that follow. Remember to mark the relationships already verified to avoid re-analysis.
		1.7.1 Does the relationship comply with the requirements specification? Check if there is really a relationship between the elements for the context.
		1.7.2 Was the relationship identified as an extension (<<extend>>) or inclusion (<<include>>) erroneously according to the requirements specification?
		1.7.3 Were the inclusion relationships between the elements specified with the <<include>> stereotype?
		Note: SMarty suggests that inclusion relationships are associated with mandatory (<<mandatory>>) or optional (<<optional>>) variants.
		1.7.4 If the element requires another, has the relationship between them been stereotyped with <<requires>>?
	1.7.5	If the relationship is mutual exclusion, is the relationship stereotyped in the use case diagram with <<mutex>>?
	1.8	Was there any relationship missing for this element that was not specified in the use case diagram?
Step 2	After analyzing all the elements in the previous step (all marked as visited), check and analyze the questions that follow.	
	2.1	If the element is of the optional type or point of variation (<<optional>> or <<variationPoint>>), check and analyze their relationships and stereotypes to answer the questions that follow.



		2.1.1	Is there a variability notation (<<variability>>) associated with the element?
		2.1.2	Do the variants specified for this variability have the correct variant notation (<<OR>>, <<XOR>> or <<optional>>)?
		2.1.3	Do the <<OR>> and <<XOR>> variants related to the variation point have the relationship <<extend>> to the associated variation point?
	2.2	The <<variability>> stereotype represents variability through a UML comment. For each of these comments defined in the diagrams, locate the comment, analyze it and answer the questions below.	
		2.2.1	Are the variants defined in the <i>variants set</i> really variants for this element? If any are not, disregard it for the next questions.
		2.2.2	Are there any variants defined in the collection of variants that are not described in the use case diagram?
		2.2.3	Are all variants related to this element defined with (<<OR>>), (<<XOR>>) or (<<optional>>) in the collection of variants of the meta-attribute variants with their correct name?
		2.2.4	Check the type of the associated variants:
			• If they are of type <<optional>>, minSelection = 0 and maxSelection = 1?
			• If they are of type <<OR>>, minSelection = 1 and maxSelection = total of variants ?.
			• If they are of type <<XOR>>, minSelection = maxSelection = 1?
	2.3	Are there still items on your list that were not specified by any elements? That is, it is missing from the use case diagram (If they are variants that have already been identified in the previous step, disregard).	



LOCATE CLASS DIAGRAM AND REQUIREMENTS SPECIFICATION

The class diagram for your role should, in a conceptual way, present the class structure and its relationships for all SPL systems in the business domain. Read the requirements specification carefully and, during the reading, make a list with the mentioned objects and the data that characterize this object. Compare the list made with the class diagram to ensure that there is no inconsistency between the classes and the user's requirements. To do this, answer the questions that follow.

Step 3	For each class in the class diagram, check the classes, the attributes, the relationships between the classes with the list made and answer the questions that follow. Check the class after its analysis, to avoid being analyzed again.	
	3.1	Does the class name correctly express the objects of this class?
	3.2	Does this class correspond to an object that has not been defined in the requirements document? If so, disregard him and his relationships for the next steps and move on to the next element.
	3.3	Is this class in redundancy with another one already specified in the class diagram? If so, disregard him and his relationships for the next steps and move on to the next element.
	3.4	Is the class stereotyped in the class diagram?
	3.5	If the element is optional or mandatory, has it been specified with the correct stereotype (<<optional>> or <<mandatory>>)?
	3.5	If the classes are grouped into packages, check and answer the questions that follow.
		3.5.1 Has the package name been defined and expresses the grouping correctly? If you have already filled out this defect for this package on the form you do not need to fill it out again
		3.5.2 Is the class in the correct package?
	3.6	For each relationship for this class, check the classes that make up this relationship, to ensure that the relationship between them is in accordance with the requirements specification. To do this, answer the questions that follow and mark the relationships already analyzed.
		3.6.1 Is this relationship in accordance with the requirements specification? Check if there is really a relationship between the elements for the context.
		3.6.2 Is the cardinality of this relationship correct according to the requirements specification?
		Check the type of relationship to ensure that the classes are stereotyped and correct according to the SMarty approach:
		3.6.3 • Generalization: Are the most general classifiers points of variation (<<variationPoint>>) and the most specific, variants?
		• Realization of interface: Are the specifications points of variation (<<variationPoint>>) and the implementations are variants?
		• Aggregation or Composition: Are the instances typed with diamonds (filled or not filled) are points of variation (<<variationPoint>>) and associated instances are variants?
	3.6.4	Are classes that require another related to the <<requires>> stereotype?
	3.6.5	Are mutually exclusive classes related to the <<mutex>> stereotype?
	3.7	Was there any relationship missing for this class that was not specified in the class diagram?



Step 4	After analyzing all the classes in the previous step (all marked as visited), check and analyze the questions that follow.	
	4.1	The control classes manage the activities of the class. For each class of this type that is stereotyped with <<optional>> and/or <<variationPoint>>. Check it to ensure that the variability/variant notations are correct according to the requirements specification. To do this, go to each of these classes and answer the questions that follow.
		4.1.1 Is there a UML note that represents variability (<<variability>>) associated with the control class?
		4.1.2 Have all the variants been defined and are they with the correct variant notation (<<OR>>, <<XOR>> or <<optional>>)?
	4.2	The <<variability>> stereotype represents variability through a UML comment. For each of these comments defined in the diagrams, locate the comment, analyze it and answer the questions below.
		4.2.1 Are the variants defined in the <i>variants</i> set really variants for this element? If any are not, disregard it for the next questions.
		4.2.2 Are there any variants defined in the collection of variants that are not described in the class diagram?
		4.2.3 Are all variants related to this element defined with (<<OR>>), (<<XOR>>) or (<<optional>>) in the collection of variants of the meta-attribute variants with their correct name?
		4.2.4 Check the type of the associated variants:
		• If they are of type <<optional>>, minSelection = 0 and maxSelection = 1?
		• If they are of type <<OR>>, minSelection = 1 and maxSelection = total of variants ?.
		• If they are of type <<XOR>>, minSelection=maxSelection=1?
	4.3	Are there still items on your list that were not specified by any elements? That is, it is missing from the class diagram (If they are variants that were already identified in the previous step, disregard).



LOCATE THE SEQUENCE DIAGRAM AND REQUIREMENTS SPECIFICATION

The sequence diagram in Domain Engineering must express the interaction of the system: the exchange of messages between the objects of the systems that can be configured from an SPL. With the sequence diagram in hand, check each of the described objects/actors and look for the corresponding class in the class diagram (if defined) or in the requirements specification. For each of them, check the relationships and messages that are exchanged between the elements to ensure that they are consistent with the class diagram/requirements specification. Then answer the questions that follow.

Step 5	Consider the "object heads" and the lifeline actors in the sequence diagram as an element. For each one of them, check the messages and stereotypes given to them in order to answer the questions that follow.	
	5.1	Does the name of the element correctly express the object/actor?
	5.2	Is the element represented in any system class?
	5.3	Is the element part of this interaction according to the Requirements Specification? If not, disregard all the lifeline and their messages for the next steps and go to the next element.
	5.4	Is the element redundant with another defined element? If so, please disregard all the lifeline and their messages for the next steps and go to the next element.
	For each of the messages defined in the lifeline of this element, analyze it and answer the questions that follow	
	5.5.1	Is the message named?
	5.5.2	Is the interaction represented by this message described in the requirements specification? If not, disregard it and go to the next message.
	5.5	5.5.3 Does the name of the message correctly express the information being transmitted?
	5.5.4	Is the order of the message correct according to the requirements specification?
	5.5.5	Check the other messages on this lifeline. Is this message redundant with another? If so disregard it and go to the next message.
	5.5.6	Messages that are not directly related to a variability and its elements, do not need a stereotype and are considered mandatory. Is the message of this type stereotyped?
	5.6	Have all the important messages for this element been defined in the sequence diagram according to the requirements specification?
	5.7	Are elements that require the presence of another related to the <<requires>> stereotype?
	5.8	Are mutually exclusive elements related to the <<mutex>> stereotype?
Step 6	Para cada elemento alternativo no diagrama de sequência como o CombinedFragment com interactionOperator "alt"(alternative) e elemento interactionUse "ref", verifique seus estereótipos e as mensagens relacionadas a este elemento para responder as questões que seguem.	
	6.1	Is the element defined with the <<variationPoint>> stereotype?
	6.2	Is there a UML notation that represents variability (<<variability>>) associated with this/CombinedFragment element?
	6.3	Are the variants corresponding to the messages stereotyped correctly? Check the stereotypes of the variant messages for this variability.



		<ul style="list-style-type: none"> • For variants of the interactionUse element "ref": <<OR>>
		<ul style="list-style-type: none"> • For variants with interactionOperator "alt": <<XOR>>
Step 7		For each optional element in the sequence diagram, such as: combinedFragment with interactionOperator "opt" (optional) and exchange of messages between two non-mandatory objects or between one mandatory object and one not, check their stereotypes and messages related to it to answer the questions that follow.
	7.1	Is there a UML notation that represents variability (<<variability>>) associated with this/CombinedFragment element?
	7.2	Are the variants corresponding to the/CombinedFragment messages correctly stereotyped with <<optional>>?
	7.3	For elements marked with <<optional>> are the lifelines that are part of the CombinedFragment also stereotyped with <<optional>>?
Step 8		The <<variability>> stereotype represents variability through a UML comment. For each of these comments defined in the diagrams, go to the comment, analyze it and answer the questions below.
	8.1	Are the variants defined in the variants set really variants for this element? If any are not, disregard it for the next questions.
	8.2	Do all variants in the variant set have an associated lifeline in the sequence diagram?
	8.3	Are all variants related to this element defined with (<<OR>>), (<<XOR>>) or (<<optional>>) in the collection of variants of the meta-attribute variants with their correct name?
	8.4	Check the type of the associated variants:
		<ul style="list-style-type: none"> • If they are of type <<optional>>, minSelection = 0 and maxSelection = 1?
		<ul style="list-style-type: none"> • If they are of type <<OR>>, minSelection = 1 and maxSelection = total of variants ?. • If they are of type <<XOR>>, minSelection=maxSelection=1?