



SCENARIO - DOMAIN ASSET MANAGER

DOMAIN ASSET MANAGER

This perspective interacts with all Domain Engineering activities to manage the versions and variants of the artifacts produced in this phase. Therefore, the Domain Asset Manager must ensure that the versions of the use case, class, component, sequence, and features diagrams are valid and traceable.

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 THE DIAGRAM TO BE INSPECTED

Step 1

Step 2

Step 3

inspection all SMarty diagrams and feature diagram

LOCATE ALL SMARTY DIAGRAMS TO BE INSPECTED

Step 4

inspection all SMarty diagrams



LOCATE THE DIAGRAM TO BE INSPECTED

Please locate the first version of the SPL diagram (oracle), check the change log for the new version of the SPL. Correct the diagram by deleting or adding the new elements as described in the record (remember the guidelines for defining the diagrams). Then, locate the version that will be inspected and compare it with the oracle to avoid inconsistency between them, marking the elements as they are being inspected. To do this, answer the questions that follow.

| | | |
|--------|--|--|
| Step 1 | For each element of the diagram being inspected, check it and mark it as visited when you finish the questions about it from Step 1: | |
| | 1.1 | Is it possible to trace the element to the oracle? If not, disregard it with your relationships and move on to the next element (remember that elements added in the change log will not appear in the oracle) |
| | 1.2 | Does the name of the element in the second version differ/is abbreviated from the first version? |
| | 1.3 | In the second version, was the element's stereotype defined? |
| | 1.4 | Is the element stereotype in the second version correct? |
| | 1.5 | For each relationship/message in this element, check and answer the questions that follow. Remember to mark the relationships already analyzed. |
| | | 1.5.1 Does this relationship/message exist in the oracle? If not, skip the next questions and move on to the next one. |
| | | 1.5.2 If so, has the relationship/message stereotype been defined correctly? |
| | | For the sequence diagram, check the messages: |
| | | 1.5.3 <ul style="list-style-type: none"> • Is the order of the message correct according to the oracle? • If defined, are the input and output parameters correct according to the oracle? |
| | 1.6 | If you are inspecting the class diagram, analyze the attributes and methods of the classes to answer the following questions: |
| | | 1.6.1 If so, are the attributes of this class correct according to the oracle? Check the name and type to answer the question. |
| | | 1.6.2 If so, are the methods of this class correct according to the oracle? Check the input and output parameters and their types to answer the question. |
| | 1.7 | If you are inspecting the component diagram, review the interfaces and components defined in the classifier format to answer the following question: |
| | | 1.7.1 Are all component/door/interface variants visible in the operating compartment as well as in the oracle? Also check for stereotypes |
| Step 2 | If you are inspecting any SMarty diagram. For each element with a related variability (<<variability>>), go to UML notation and look for it in the second version: | |
| | 2.1 | Is the variability notation present in the inspected version? |
| | 2.2 | Is the name (meta-attribute name) of the variability redundant with another already defined? |
| | 2.3 | Does the name (meta-attribute name) of the variability a reflect correctly? |
| | 2.4 | Are minSelection and maxSelection set correctly? Remember to check if a new variant has been inserted according to the change log. |
| | 2.5 | Check the associated variants. Are they present with their correct name as in the oracle? |
| | 2.6 | Check the alllowsAddingVar meta-attribute. Is it correct according to the oracle? |
| | 2.7 | Check the bindingTime meta-attribute. Is it correct according to the oracle? |
| | 2.8 | Check the realizes + and realizes- meta-attributes. Are they specified in the second version, as well as in the first? |



| | | |
|--------|--|---|
| Step 3 | After analyzing all the elements in the previous steps (all elements of the inspected diagram marked as visited), check and analyze the questions that follow in the oracle diagram. | |
| | 3.1 | Is any element of the oracle not present in the inspected version? |
| | 3.2 | Is there any relationship of the oracle not present in the previewed version? |

| LOCATE ALL SMARTY DIAGRAMMS TO BE INSPECTED | | |
|---|---|---|
| The realizes+ and realizes- meta-attributes allow traceability in SMarty diagrams. For each inspected diagram, check which diagrams it is traceable to. To ensure that it is really possible to trace these elements from one diagram to another, please review the diagram and answer the following questions. | | |
| Step 4 | Locate the diagram to be inspected. For each variability (UML comment stereotyped with <<variability>>), locate the elements realizes+ and realizes-. | |
| | • realizes+ should contain the name of variability(ies), thus locate the higher level diagrams with such variability(ies) | |
| | • realizes- should contain the name of variability(ies), thus locate the lower level diagrams with such variability(ies) | |
| | 4.1 | Did you find (match) the variability(ies) informed in the realizes+ or realizes- attributes? |
| | 4.2 | Does the variability in the corresponding diagram(s) realizes the variability of source inspected diagram? |
| | 4.3 | Is there any missing variabilities in the corresponding diagram(s), informed in the source inspected diagram? |