Project Model-based Traceability

Team MODEL INC.

Use Case Report V1.2

Version History

#	Date	Prepared By	Description	Reviewed By
1.0	10/10/2016	Aishwarya	Initial Draft	Aparrnaa & Thejesvini
1.1	5/12/2016	Thejesvini	Made some changes to view traceability use case	Aishwarya
1.2	12/02/2016	Aparrnaa	Refined Use Cases	Team

Contents

1. Introduction

a. Project Overview

Model-based traceability is a tool which would aid the project teams in pulling the reliable project artifacts for a change request within a minute to meet the client's SLA period. The tool will be built in such a way that model manager is provided with a user interface using which he will be able to create model, associate properties based on the CMMI level 5 standard and assign user roles and responsibilities. Currently, wrong artifacts are mapped up say for example, business use cases being mapped to system test cases since there are no user level restrictions. Because of this, when there is a change request it gets difficult to track all the right artifacts to the corresponding requirement. Hence, with the model manager tool a desktop based application in place project manager can define the model and assign user roles and restrictions.

Developers and other team members would be provided with a traceability web UI wherein he can create project instances(artifacts) and link them up. Using the standard model defined in the model manager tool, the project team can create project instances(artifacts) based on their respective roles and responsibilities.

Consider a project model defined in model manager tool is - Requirement → Design → Test Case and it is defined with properties for each element. The model manager would then assign roles and restrictions. Based on this, developer X can create a requirement artifact *"R098 Implement Search Engine"* and link it up with design artifact "*D098 Search Engine Design*" only if he has given the rights to do so.

This application will also provide an option to view traceability and generate project reports which can then be used for creating the patch for a change request. While viewing the traceability, all the related artifacts of base and customized version for each client have to be listed. For this reason, model manager is provided with Import Project Model option using which he can copy the project details from the base version and later customize it.

By restricting the users, we are able to achieve reliability,

- Project manager can benefit by ensuring that unauthorized access to create artifacts could be minimized.
- Developers do not have the access to create or link up wrong artifacts.

b. Purpose of the document

This document captures the requirements provided by the customer and describes the actors and the scenarios involved in the project.

c. Audience

The audience of this document includes,

Mentor - Professor Sivakumar S

Client – SSK Private Limited

MODEL Inc. Team Members

- Aparrnaa R
- Ajay Akkineni
- Arul Dhana Saam Prakash
- Aishwarya A
- Dinesh Parthiban
- Sriram J
- Thejesvini R

2. Actors

The various actors involved in the project are:

Actor Specification
Actor Name: Model manager
Description: Model manager can be an Architect / the Project Manager within the project team who has the authority to create/update/delete the project model and define the properties for each element. He also defines the roles and responsibility of each member associated with the project. He will be the single point of contact and owner for the project model. Security and Reliability will be promoted if the model manager duly performs his task.

Actor Specification
Actor Name: Developers
Description: A developer can be anybody who belong to the project team. He/She would use traceability tool to create project instances(artifacts) and link them up based on their role and restrictions. He can also view the traceabilities pertaining to the change request and generate report. The results returned from the traceability tool will be used by him to identify the patch for other clients.

3. Use case model

Below is the use case diagram which shows different actors and the use cases involved in the project.

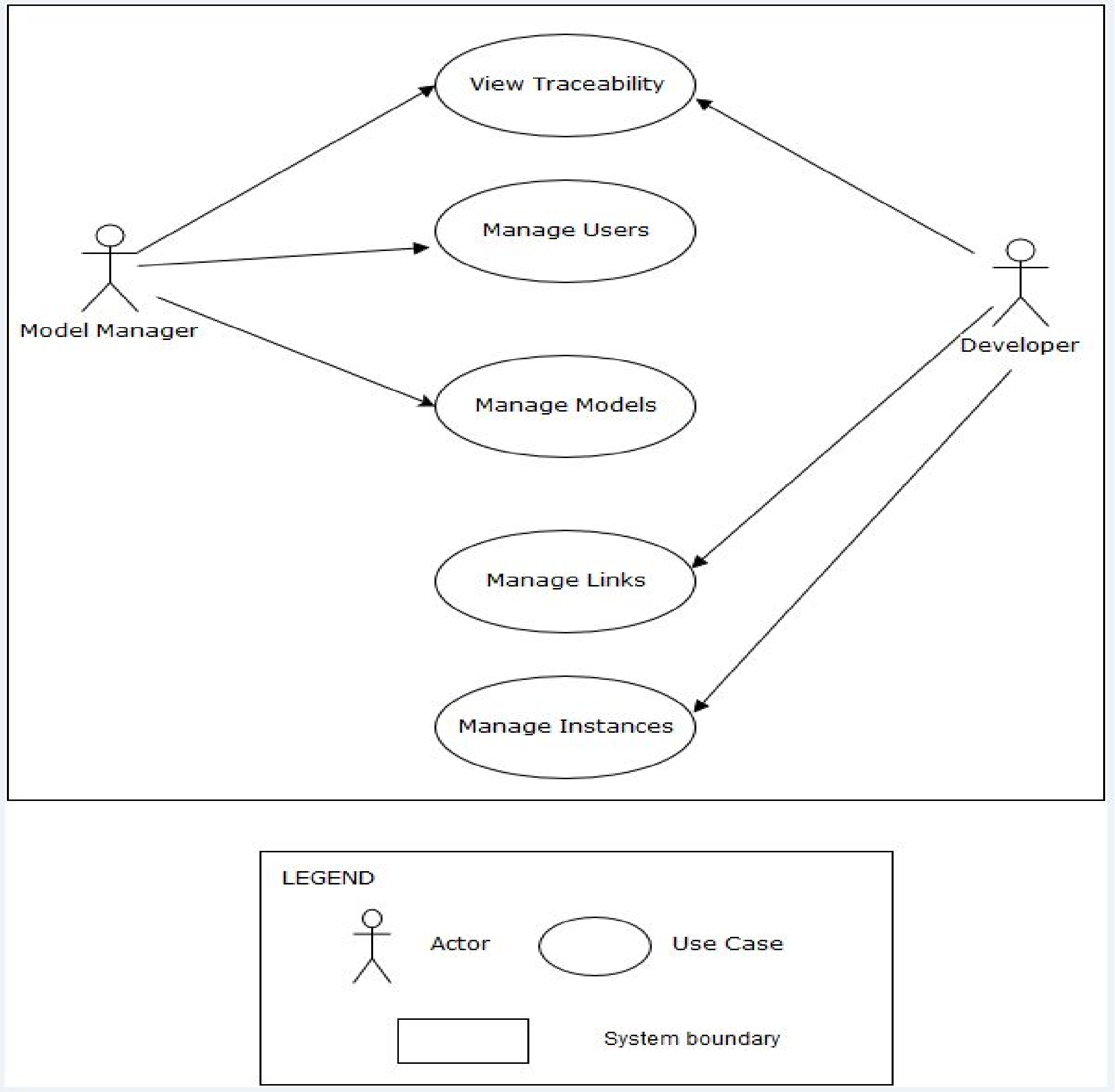

Fig I: Model-based Traceability Use case diagram

These use cases were arrived at based on the stakeholder interactions and his needs. Backup and restore model were identified as system level use cases but after post architecture estimation we have reduced the scope of the project and have taken backup and restore model use cases for future enhancements.

4. Use case specification

Below use case specification describes various use cases of the system with exceptional and alternate flows.

- **Manage Models**

Use case name	**Manage Models**
Use case ID	UC #1
Primary actor(s)	Model manager
Secondary actor(s)	None
Brief Description	The model manager can create, update, delete and save a project model. He will be able to associate properties for all the entities defined in the model. This use case gets triggered when the model manager wants to performs one of the below mentioned operations: • He has created a new project, and wants to define a model for that project. • When he wants to create, or edit the properties associated with each entity defined in the model. • He wants to import an existing project model and make necessary modifications. • IF the model manager would like to view the model for an existing project for which he has been assigned, then he can view the model or make any changes to the model entities or the properties of the model entities. • IF the model already created for the project is no longer required, the model manager can delete the model along with its associated properties. The model manager can perform one of the above-mentioned operations.
Pre-conditions	The model manager has access to the Model Manager tool and has the privileges to create a new project or open an existing project for modifications.
Flow of events	Any one of the following alternate flows might get triggered based on whatever operation the model manager wants to perform.
Alternate flows and exceptions	**Create Project Model:** IF the model manager wants to "Define Model" for a new project which he has created, then refer AF #1 **Import/Copy Project Model:** When the model manager wants to "Import Model" for a new project which he has created, then refer AF #2

	Delete Project Model: When the model manager wants to "Delete Project Model" completely, such that he is no longer responsible for maintaining the project. He opens up the corresponding project form the dropdown listed after he gets authenticated. The system provides a delete option which he clicks to completely remove the project from the dropdown listing and make it unavailable for further maintenance. **Save Project Model:** 1. During all the alternate flow mentioned above, the model manager can choose to save the project model for later purpose. The system provides a save option in the UI which the manager can click to persist the model 2. If system does not respond for the save option, corresponding feedback will be provided to the manager and state-of-the-system is maintained.
Post-conditions	The model entities and properties are updated in the system.
Non-behavioral requirements	The model manager must be able to configure the model entities and properties according to his project and be given with an interactive UI for defining/editing the model.
Assumptions	1. The employee chosen as the model manager for the project would be a trusted one and would adhere to CMMI level 5 standards as closely as possible.
Priority	High
Issues	
Source	Stakeholder Meeting MOM- 23/09/2016

Alternative Name	Create Project Model
ID	AF #1
Primary Actor	Model manager
Secondary Actor	None
Brief Description	This alternate flow is defined for the model manager who wants to "Define Model" for a new project which he has created. He would also associate properties for each entity in the project model.
Preconditions	1. The manager has created a new project 2. The manager chooses to define the model from scratch 3. An empty canvas is presented to the model manager where he will be able to view the list of entities as per CMMI level 5 standard. {Note: The system shall provide a model canvas editor with an interactive UI where he can define the project model.}

Flow of events	1. Model manager creates an entity (requirement, design, code, test case, or etc) for the project model by providing the entity name and description. 2. The model manager has to define the properties for the entity created. He can choose to customize the standard CMMI level 5 properties provided by the system or create new properties on his own. {Note: The system shall store the standard properties for every entity which will be made available to the model manager once he has created an entity} 3. Step 2 would end successfully, when the manager has defined all the properties for a particular entity. 4. Repeat step 1 and 2 until all other entities (requirement, design, code, test case, or etc.) to be present in the project would be created. 5. Model manager then appropriately provides the linkages between the entities that are created above. 6. He then saves the project model once he thinks that everything is precise and accurate or when he wishes to resume his work at later point in time. 7. This alternate flow ends successfully when the system provides the feedback that "*The project model has been created successfully*" to the model manager.
Postconditions	The project model is saved in the data store and will be available for the later purpose of defining the project instances and to link them up.
Priority	High
Non-behaviorial requirements	None
Assumptions	1. The model manager has gone through the user manual and knows how to work with the interactive UI to create the project model. 2. This flow would get initiated when a new project request is sent to the model manager
Issues	1. Manager faces an inherent lag while defining the entities or while linking them up. 2. When the project model is not saved successfully then the system has to make sure that the model created by the model manager does not get erased. Corresponding feedback has to be provided to the manager and the system should restore its state automatically.
Source	Stakeholder Meeting MOM- 23/09/2016

Alternative Name	Import Project Model
ID	AF #2
Primary Actor	Model manager

Secondary Actor	None
Brief Description	This alternate flow is defined for the model manager who wants to "Import Project Model" from an existing project and wants to customize it for the new project request. Modifications will be done to the entities, or the linkages, or the properties which were already defined in the version that it is derived from.
Preconditions	1. The manager has a new project which is quite similar to the already existing and chooses to import it. 2. The model editor canvas would be populated with the existing project model, properties and linkage details chosen by the manager.
Flow of events	Once the project model is made available in the system for modification, the model manager would perform on of the following operation: 1. Model manager might want to incorporate a new entity into the already existing project model. In which case, he can would define the entity by providing the entity name and description. 2. The model manager would want to modify or define the properties for the new entity that he created in step 1, in which case the system would provide the standard CMMI level 5 properties or create new properties on his own. 3. Model manager might want to edit the already linked entities in the project model. 4. He then saves the project model once he thinks that everything is precise and accurate or in case he wishes to resume his work at later point in time. 5. This alternate flow ends successfully when the system provides the feedback that "*The project model has been saved successfully*" to the model manager.
Postconditions	The project model is saved in the data store and will be available for the later purpose of defining the project instances and to link them up.
Priority	High
Non-behaviorial requirements	None
Assumptions	3. The model manager has gone through the user manual and knows how to work with the interactive UI to import the project model from already existing model. 4. This flow would get initiated when a new project request is sent to the model manager and the process followed is similar to already defined project model.
Issues	When the project model is not saved successfully then the system has to make sure that the model created by the model manager does not get erased. Corresponding feedback has to be provided to the manager and the system should restore its state automatically. The exception has to be handled appropriately by the system.
Source	Stakeholder Meeting MOM- 23/09/2016

- **Manage Users**

Use case name	**Manage Users**
Use case ID	UC #2
Primary actor(s)	Model manager
Secondary actor(s)	None
Brief Description	The model manager would want to create user roles and update them whenever required. He can also assign access permissions to each entity defined in the project model. This use case requirement is the critical functionality that has to be developed to make the system more reliable and prevent the development team in linking up wrong artifacts. For instance, if developer X This use case gets triggered when the project model is defined and the manager wants to create the roles and restrictions for each member in the project team such that unauthorized manipulation could be prevented.
Pre-conditions	The model manager has created a new project model for a particular project or he wishes to update the user roles and permissions for an already existing project.
Flow of events	1. The model manager adds users to his project. 2. The model manager defines the roles that are present in his project. 3. He can either choose from the given roles or add a new role. 4. Once the model manager has defined the project roles, he can assign specific users to these roles. 5. The model manager assigns specific roles to each user involved in the project, and provides read/write permissions to entities in the model.
Alternate flows and exceptions	**Create User Roles:** IF the model manager wants to “Create User Roles” then, the system provides **Update User Roles:** When the model manager wants to “Update User Roles” then,
Post-conditions	The users for a project and their corresponding access permissions to model entities are updated. Based on these least privilege restrictions developer can view only what is essential and necessary for him.
Non-behavioral requirements	Role-based user restrictions have to be incorporated by providing read and write permissions.
Assumptions	
Priority	High
Issues	
Source	Stakeholder Meeting MOM- 23/09/2016

- **Manage Instances**

Use case name	**Manage Instances**
Use case ID	UC #3
Primary actors	Developers
Secondary actors	Model manager
Brief Description	Other users apart from Project Manager create, update and delete instances in the model.
Preconditions	The model for the particular project has been created. The meta properties of the instances are visible to the user.
Flow of events	1. The user views the existing instances accessible to him (according to his role in the project). 2. IF the user wants to create instances, he views the meta properties of the instances, fills in values for each of these accordingly and saves them as a separate instance. 3. IF the user wants to create another instance, he repeats step 2. 4. IF the user wants to edit or delete the intended instances, he selects the edit or delete option.
Post conditions	The instances for the particular entity are updated.
Priority	High
Alternate flows and exceptions	-
Non-behavioral requirements	The user must find it easy to create, update and delete instances without having to remember the properties details. The user must not be able to make changes in any of the instances that are out of his access.
Assumptions	1. The user must be able to view the previous and next instances related to the entity he is working on.

- **Manage linkages/links**

Use case name	**Manage linkages/links**
Use case ID	UC #4
Primary actors	Developers
Secondary actors	Model manager
Brief Description	Other users apart from Project Manager links the instances that are accessible to them.
Preconditions	The relevant instances are created and reviewed by the users.
Flow of events	1. The user views the instances phase-wise. 2. He links the intended instances of one entity to the instances of the

	previous or next entity. 3. The user verifies whether he has linked the instances correctly. 4. IF the user feels he has linked up two instances incorrectly, then he edits or deletes that particular link between these instances.
Post conditions	The instances accessible to the user have been linked and reviewed.
Priority	High
Alternate flows and exceptions	-
Non-behavioral requirements	The user must find it easy to link the instances. He must also find it easy to edit these links when needed. The user must not be able to edit links that are outside of his access.
Assumptions	1. The user must be able to view the previous and next artifacts related to the artifact he is working on.

- **View Traceability**

Use case name	**View traceability**
Use case ID	UC #5
Primary actors	Developers, model manager
Secondary actors	-
Brief Description	The users view the traceability of the project instances and the associated linkages between them. Traceability also includes the instances (and linkages) from its parent and/or child project versions if any.
Preconditions	The model has been defined by the manager. Project instances are created and linked by the developers.
Flow of events	1. Developer logs into the model-based traceability tool. 2. System displays all the projects that the developer is authorized to work on. 3. Developer selects one project. 4. System provides the instances pertaining to that project along with the instance descriptions. 5. The user can also additionally give the name of a particular instance to view traceability. 6. The user can now view the traceability of the project across modules.
Post conditions	The traceability of the project is seen.
Priority	High
Alternate flows and exceptions	-
Non-behavioral requirements	The traceability must be easily understood when viewed by the users. Traceability pulls up more records(say 10000 +) how to display the details to the developer/manager such that it is easy to understand and exported into pdf.
Assumptions	The users can view the names of the projects and related instances for guidance.