# Validation of MT-DSL Transformations

CSI 6900 Project Description, Winter 2014

Supervisors: Daniel Amyot (EECS)

Student: Xinyu Zhao (7064188)

## Abstract

Requirements are a key aspect of software development, which are related to other software artefacts, such as designs, test cases and documentation[1]. In order to track the relationships between requirement artefacts and these other software artefacts, a new Domain-Specific Language was introduced in Anisur Rahman's paper, which can describe the concepts of a modeling language in order to be traced in a Requirement Management System (RMS), such as Rational DOORS [2].

The MT-DSL and the tools are demonstrated for model importing and evolution scenarios (re-importing) with one URN model and three AoURN models[3, 4]. This report contributes a validation to the MT-DSL work.

## Table of Contents

# 1. Introduction

The MT-DSL work deserves additional validation, the editor and a library of import and maintenance functions from Anisur Rahman's work will be used in this report. Three large AoURN models (radio-solution, CCCMS and YouKeyKnowsv7) are validated and the results shown in the following chapters.

In Chapter 2, two MT-DSL models will be presented with short explanations. In Chapter 3, the process of generating the DXL library and an overview of the Generated DXL library for jUCMNav/URN models[5, 6] will be given. Experiments and validations of a URN model (HelloWorld) and three large AoURN models are in Chapter 4, including test scenarios, imported results and re-importing changes. Bugs and some important issues in MT-DSL will discuss in Chapter 5.

**Problem Statement**

We need to further validate the MT-DSL work, the module 'Concern' and 'Scenario' should be added as modules in the MT-DSL description to generate the DXL library. Plus the related Eclipse plugin should be modified as well.

For the existing modules such as 'ActorRef' and 'IntentionalElementRef', there are some attributes are not important any more which should be removed. For instance, the 'Height' and 'Weight' from these two modules don't deserve to be traced. On the other hand, there are some attributes are not included before, however, they are important to be traced (e.g. 'important Type' and 'important Quantitative'). In this report, these new important attributes are added to be traced and validated.

# 2. Model Traceability Domain-Specific Language Models

In this chapter, we will introduce two MT-DSL models, which are presented in Appendix A and B. The MT-DSL model of Appendix A describes a subset of jUCMNav metamodel which can be supported by DXL library.The MT-DSL file is the only one need to be changed when any modification is applied to the existing

DXL library [1]. Compare with the original MT-DSL description of URN, I added ‘Concern’ and ‘Scenario’ modules in this MT-DSL model as mentioned above so they can be traced and imported in DOORS. The attributes ‘importance Type’ (string) and ‘importance Quantitative’ (int) are added to the modules ‘Actor’, ‘IntentionalElemen’t and ‘Contribution’. Plus, the modules ‘map’ and ‘grlDiagram’ have new association to concern by using the ID and name of ‘Concern’.

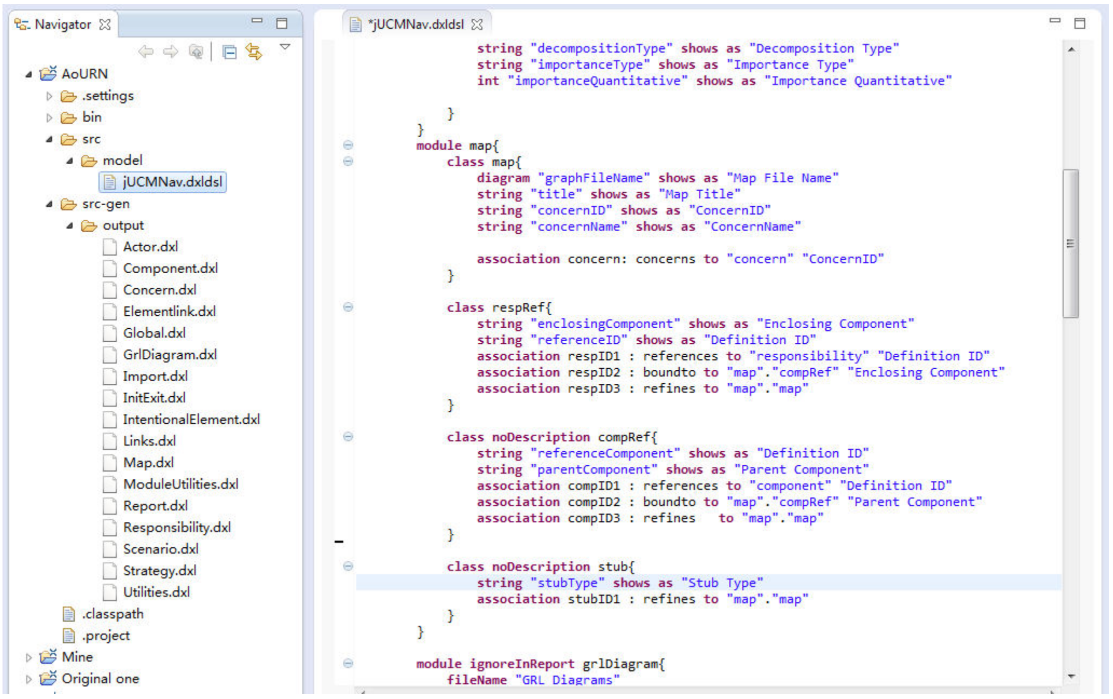
The following partial MT-DSL from appendix A shows the several new modules, attributes and associations.

```
module concern{
                class concern{
                    string "condition label" shows as "Condition Label"
                    string "condition expression" shows as "Condition Expression"
                }
          }

module ignoreInReport actor{
                class actor{
                    string "importanceType" shows as "importance Type"
                    string     "importanceQuantitative"     shows     as     "importance
Quantitative"
                }
          }

module map{
                class map{
                    diagram "graphFileName"    shows as "Map File Name"
                    string "title" shows as    "Map Title"
                    string "concern"
                    association map1: concerns to "concern"."concern" "concern"
                }

module scenario{
                class scenario{
                }
          }
```

The appendix B includes another MT-DSL model, which includes new modules, attributes and associations. For example, ‘starPoint’, ‘endPoint’ as classes in module "map" can be tracked and imported in this MT-DSL description. The related associations are also added in this MT-DSL description.

## 3. Traceability Library Generation for Importing jUCMNav Models in DOORS.

In this chapter, we will describe the DXL traceability library generation according to the MT-DSL description. The full version of MT-DSL description is presented in Appendix A, which generated the DXL library with the part of AoURN which needs to be traced and imported in DOORS. Figure 1 shows the outline view of AoURN models and links described in MT-DSL.



```
            string "decompositionType" shows as "Decomposition Type"
            string "importanceType" shows as "Importance Type"
            int "importanceQuantitative" shows as "Importance Quantitative"

        }
    }
    module map{
        class map{
            diagram "graphFileName" shows as "Map File Name"
            string "title" shows as "Map Title"
            string "concernID" shows as "ConcernID"
            string "concernName" shows as "ConcernName"

            association concern: concerns to "concern" "ConcernID"
        }

        class respRef{
            string "enclosingComponent" shows as "Enclosing Component"
            string "referenceID" shows as "Definition ID"
            association respID1 : references to "responsibility" "Definition ID"
            association respID2 : boundto to "map"."compRef" "Enclosing Component"
            association respID3 : refines to "map"."map"
        }

        class noDescription compRef{
            string "referenceComponent" shows as "Definition ID"
            string "parentComponent" shows as "Parent Component"
            association compID1 : references to "component" "Definition ID"
            association compID2 : boundto to "map"."compRef" "Parent Component"
            association compID3 : refines   to "map"."map"
        }

        class noDescription stub{
            string "stubType" shows as "Stub Type"
            association stubID1 : refines to "map"."map"
        }
    }

    module ignoreInReport grlDiagram{
        fileName "GRL Diagrams"
```

**Figure 1** MT-DSL Description of the AoURN

Table 1 displays the generated DXL library for this MT-DSL description of jUCMNav/AoURN model, which includes the size of each file (lines of code), number of DXL functions and description.

| Library File Name | Size | Func | Description of DXL Library File |
|---|---|---|---|
| Actor.dxl | 50 | 1 | DXL file for |

| | | | |
|---|---|---|---|
| | | | module Actor |
| Component.dxl | 44 | 1 | DXL file for module Component |
| Concern.dxl (New) | 48 | 1 | DXL file for module Concern |
| Elementlink.dxl | 105 | 2 | DXL file for module Elementlink |
| GrlDiagram.dxl | 189 | 4 | DXL file for module GrlDiagram |
| IntentionalElement.dxl | 56 | 1 | DXL file for module IntentionalElement |
| Map.dxl | 192 | 4 | DXL file for module Map |
| Responsibility.dxl | 41 | 1 | DXL file for module Responsibility |
| Scenario.dxl(New) | 41 | 1 | DXL file for module Scenario |
| Strategy.dxl | 42 | 1 | DXL file for module Strategy |
| Import.dxl | 196 | 2 | This includes the DXL function beginImport to start the import process. |
| InitExit.dxl | 593 | 11 | Contains all the DXL functions to initialize and finalize the import process (including GUI interactions) |
| ModuleUtilities.dxl | 169 | 7 | Includes the helper DXL functions that are invoked during the model import process in DOORS. |
| Utilities.dxl | 24 | 0 | Contains the list of import statements to import all other library files. |
| Links.dxl | 224 | 3 | Contains DXL library code for the links described in the modules. |
| Report.dxl | 200 | 6 | Contains generated |

| | | | |
|---|---|---|---|
| | | | DXL code for creating a report at the end of the import process. |
| Global.dxl | 134 | 0 | Declares global variables used in all DXL files in the library. |

**Table 1** Generated DXL library for jUCMNav/URN models

# 4. Experiments and Validation

In this chapter, we will present four experiments, which include one small URN model and three large AoURN models to validate the tools by using the MT-DSL description to generate the DXL library, exporting DXL files from jUCMNav diagrams, importing jUCMNav image files in DOORS by using the DXL script. The MT-DSL description for the AoURN model is provided in Appendix A, the generated DXL library for the subset of AoURN is described Chapter 3 and the overview of it is shown in Table 1.

## 4.1 URN Model

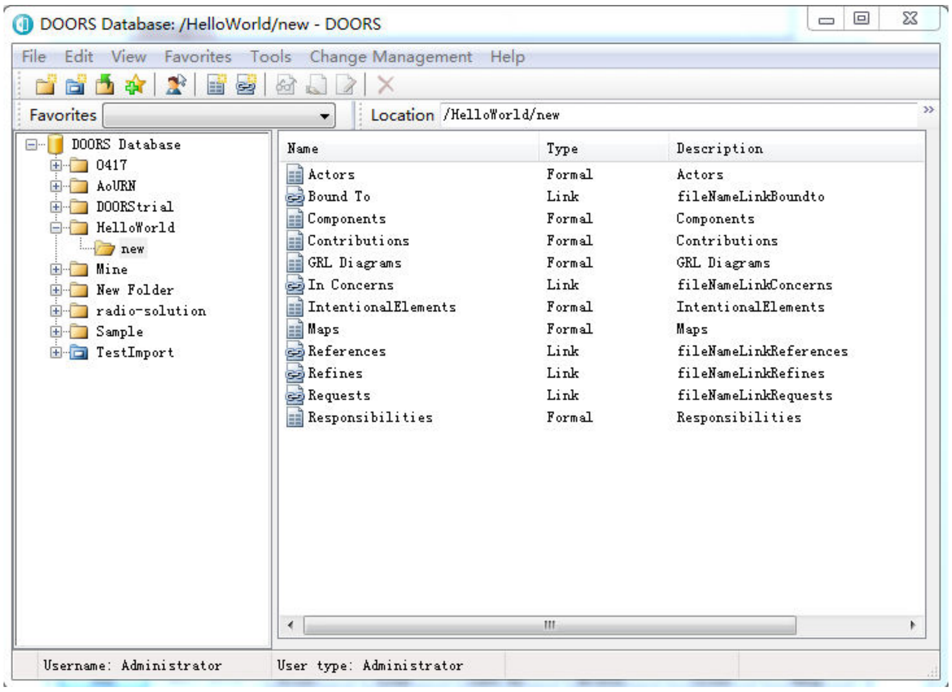
In this section, a small URN model will be given as the first experiment. This model contains 1 UCM Map and 1 GRL Diagram and other modules and associations. The overview of generated DXL library is described in Chapter 3. The full version of MT-DSL is presented in Appendix B.

### 4.1.1 Test Scenario

The following DXL script is used to invoke the generated DXL library and to import the URN model in DOORS, which is exported automatically from jUCMNav.

```
#include "addins/dsl/lib//Utilities.dxl"
pragma runLim, 0

beginImport( "HelloWorld" )

actor( "27", "Vendor", "" )
actor( "29", "Customer", "" )
actor( "31", "Database", "" )

component( "39", "Customer", "", "Team" )
component( "41", "Vendor", "", "Team" )
component( "45", "Database", "", "Team" )

responsibility( "288", "Check Availability", "" )
responsibility( "294", "Provide Information of Items", "" )
responsibility( "298", "Provide Items", "" )

intentionalElement( "21", "Provide Info", "" )
intentionalElement( "23", "Sell", "" )
intentionalElement( "25", "Buy", "" )

contribution("34", "Contribution34", "", "contribution", "Help", 25,"0", "23", "25" )
contribution("37", "Contribution37", "", "contribution", "Help", 25,"0", "21", "23" )

grldiagram( "2", "GRLGraph2", "", "D:/Study/Winter 2014/Project/Report/0506
Improvement/AoURN Export/HelloWorld-GRLGraph2-GRLGraph2.bmp",
"GRLGraph2", "", "" )
    actorRef( "28", "ActorRef28", 529,50,150,118,"27" )
    actorRef( "30", "ActorRef30", 145,-87,140,126,"29" )
    actorRef( "32", "ActorRef32", 220,167,149,115,"31" )
    intentionalElementRef( "22", "Provide Info", "", 238,197,"32", "21" )
    intentionalElementRef( "24", "Sell", "", 557,87,"28", "23" )
    intentionalElementRef( "26", "Buy", "", 160,-50,"30", "25" )

map( "3", "UCMmap3", "", "D:/Study/Winter 2014/Project/Report/0506
Improvement/AoURN Export/HelloWorld-Map3-UCMmap3.bmp", "UCMmap3" )
    respRef( "289", "Check Availability", "", 485,205,"42", "288" )
    respRef( "295", "Provide Information of Items", "", 758,285,"46", "294" )
    respRef( "299", "Provide Items", "", 516,299,"42", "298" )
    compRef( "40", "Customer", 126,152,222,211,"39", "" )
    compRef( "42", "Vendor", 421,149,214,213,"41", "" )
    compRef( "46", "Database", 737,150,206,215,"45", "" )
    startPoint( "58", "Order Items", "" )
    endPoint( "60", "Out of Stock", "" )
    endPoint( "283", "", "" )

endImport
```

The result of this URN Model is shown as follow, Figure 2 shows the DOORS formal and link modules (associations) from it. We can see the modules such as ‘Actors’, ‘IntentionalElements’ and associations such as ‘References’ are imported correctly.


DOORS Database: /HelloWorld/new - DOORS
File Edit View Favorites Tools Change Management Help
Favorites
Location /HelloWorld/new

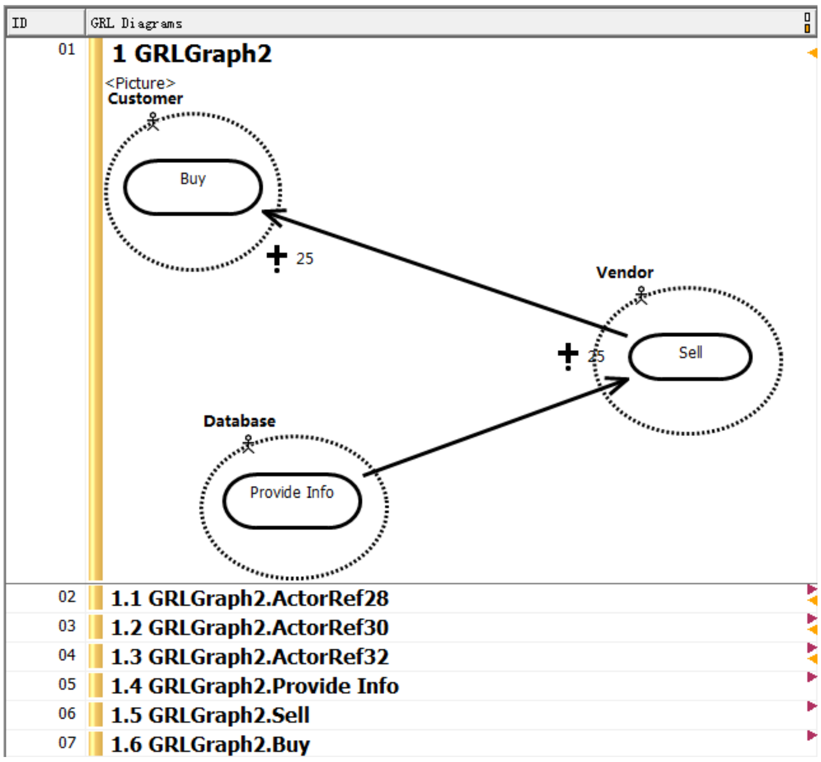
- DOORS Database
  - 0417
  - AoURN
  - DOORStrial
  - HelloWorld
    - new
  - Mine
  - New Folder
  - radio-solution
  - Sample
  - TestImport

| Name | Type | Description |
|---|---|---|
| Actors | Formal | Actors |
| Bound To | Link | fileNameLinkBoundto |
| Components | Formal | Components |
| Contributions | Formal | Contributions |
| GRL Diagrams | Formal | GRL Diagrams |
| In Concerns | Link | fileNameLinkConcerns |
| IntentionalElements | Formal | IntentionalElements |
| Maps | Formal | Maps |
| References | Link | fileNameLinkReferences |
| Refines | Link | fileNameLinkRefines |
| Requests | Link | fileNameLinkRequests |
| Responsibilities | Formal | Responsibilities |

Username: Administrator  User type: Administrator

**Figure 2** Imported URN Model in DOORS

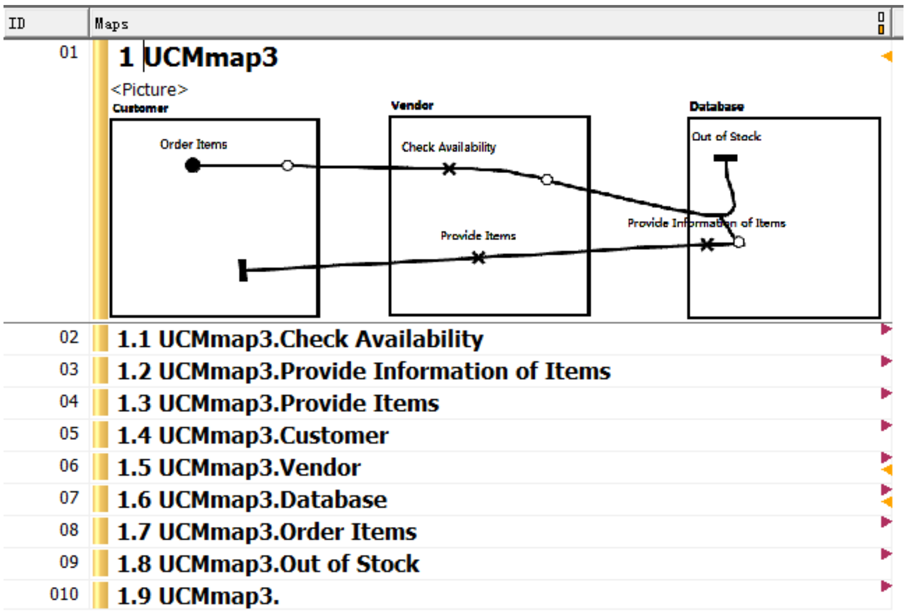
Figure 3 shows the GRL Diagram (#ID2), which contains 3 ‘actorRef’, 3 ‘intentionalElementRef’ and the association ‘References’ to Actors.



| ID | GRL Diagrams |
|---|---|
| 01 | 1 GRLGraph2 |
| 02 | 1.1 GRLGraph2.ActorRef28 |
| 03 | 1.2 GRLGraph2.ActorRef30 |
| 04 | 1.3 GRLGraph2.ActorRef32 |
| 05 | 1.4 GRLGraph2.Provide Info |
| 06 | 1.5 GRLGraph2.Sell |
| 07 | 1.6 GRLGraph2.Buy |

**Figure 3** GRL Diagram module in DOORS

The following figure shows the imported ‘UCM Map’ module, which includes the content of the image file “HelloWorld-Map3-UCMmap3.bmp”. The attributes include ‘respRef’, ‘compRef’, ‘startPoint’ and ‘endPoint’.



| ID | Maps |
|---|---|
| 01 | **1 UCMmap3** <Picture> |
| 02 | **1.1 UCMmap3.Check Availability** |
| 03 | **1.2 UCMmap3.Provide Information of Items** |
| 04 | **1.3 UCMmap3.Provide Items** |
| 05 | **1.4 UCMmap3.Customer** |
| 06 | **1.5 UCMmap3.Vendor** |
| 07 | **1.6 UCMmap3.Database** |
| 08 | **1.7 UCMmap3.Order Items** |
| 09 | **1.8 UCMmap3.Out of Stock** |
| 010 | **1.9 UCMmap3.** |

**Figure 4** Map module in DOORS

Figure 5 shows the link ‘Reference’ and I chose the one between GRL Diagram and Actors. The association is connected by ‘actorRef’, which is contained in ‘GRL Diagram’.

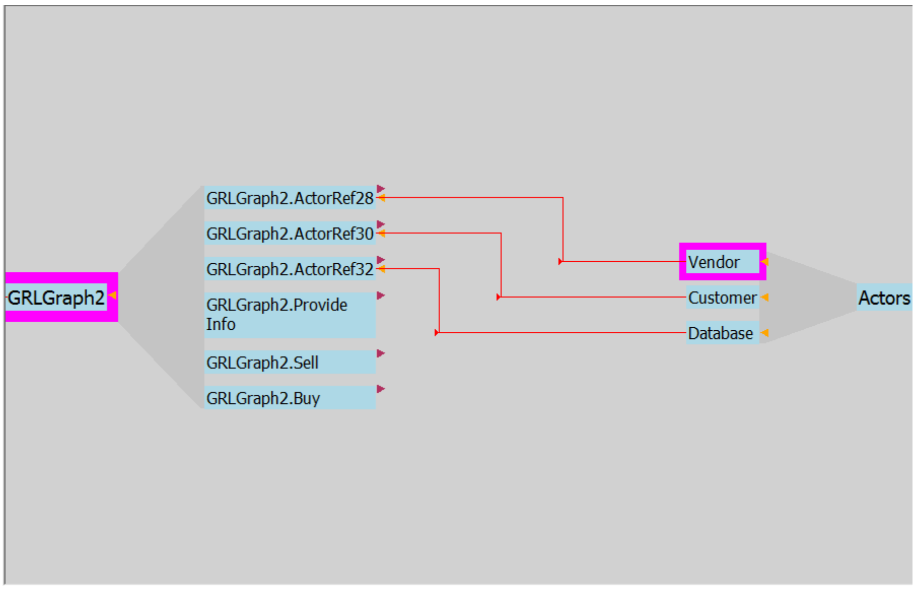
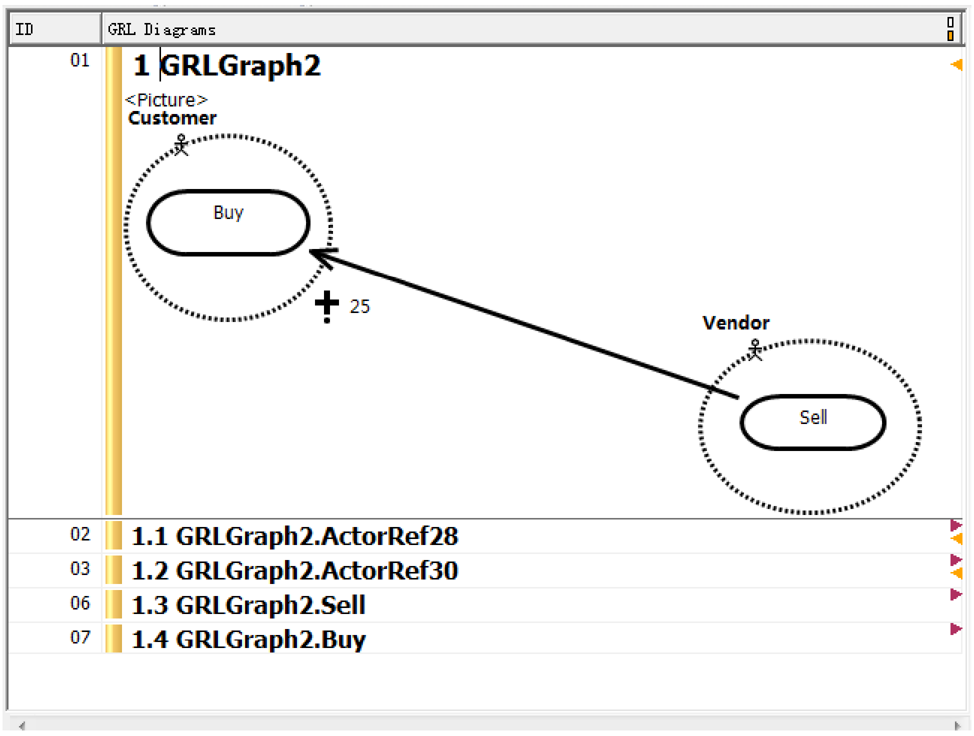


**Figure 5** ‘Reference’ links between GRL Diagrams and Actors

### 4.1.2 Re-importing Changes

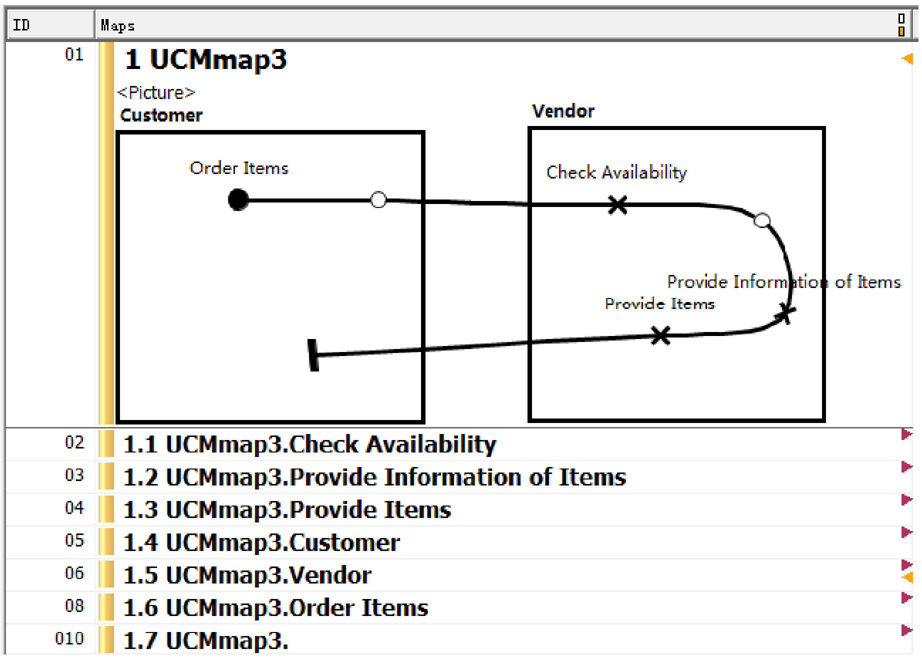
In this section, I will modify some parts of the original jucm file and import the model to DOORS again by using the new generated DXL script. I deleted the actor ‘Database’ and the goal ‘Provide Info’ in GRL Diagram and re-import it in DOORS. The following figure shows the new GRL Diagram after re-importing.



| ID | GRL Diagrams |
|---|---|
| 01 | 1 GRLGraph2 <br> <Picture> |
| 02 | 1.1 GRLGraph2.ActorRef28 |
| 03 | 1.2 GRLGraph2.ActorRef30 |
| 06 | 1.3 GRLGraph2.Sell |
| 07 | 1.4 GRLGraph2.Buy |

**Figure 6** GRL Diagram module in DOORS after re-importing

For UCM Map, I deleted the component ‘Database’ and the related contribution and re-imported it in DOORS. Figure 7 shows the UCM Map in DOORS after re-importing.



| ID | Maps |
|---|---|
| 01 | **1 UCMmap3**<br>&lt;Picture&gt; |
| 02 | **1.1 UCMmap3.Check Availability** |
| 03 | **1.2 UCMmap3.Provide Information of Items** |
| 04 | **1.3 UCMmap3.Provide Items** |
| 05 | **1.4 UCMmap3.Customer** |
| 06 | **1.5 UCMmap3.Vendor** |
| 08 | **1.6 UCMmap3.Order Items** |
| 010 | **1.7 UCMmap3.** |

**Figure 7** UCM Map module in DOORS after re-importing

Lastly, the link ‘References’ changed as well after I did the changes to UCM Map and GRL Diagram. The following figure shows this link after re-importing.

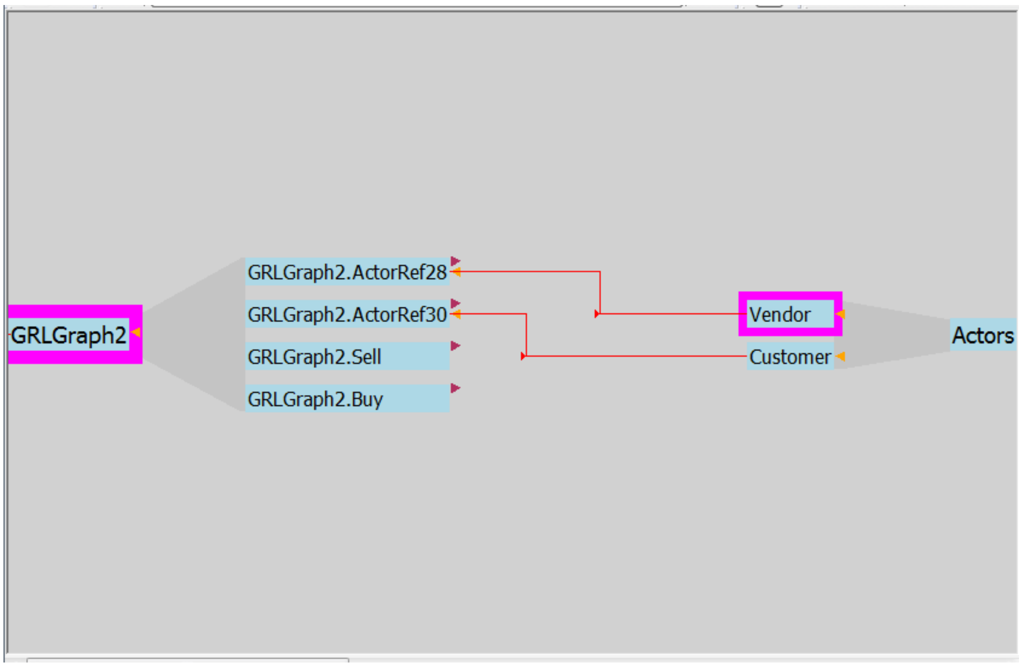


**Figure 8** ‘Reference’ links between GRL Diagrams and Actors after re-importing

### 4.2 Radio-solution Model

This model contains 8 ‘Map’ modules and 7 ‘Concern’ modules, we choose the map with name ”Power” and “Select Band” as test scenarios to validate invoking the generated DXL library and importing the subset of this model

### 4.2.1 Test Scenarios

For this part, I used the following DXL script which is exported from jUCMNav to invoke the generated library and to import the radio-solution model in DOORS. I can import only one map at one time to DOORS. Plus this model includes the new modules: ‘Concern’ and ‘Scenario’, new associations between ‘Concern’ and ‘map’ and ‘grlDiagram’.

```
#include "addins/dsl/lib//Utilities.dxl"
pragma runLim, 0

beginImport( "radio-solution" )

component( "11", "user", "", "Team" )
component( "13", "radio", "", "Team" )

concern( "4614", "Autotune", "", "", "" )
concern( "4615", "Display", "", "", "" )
concern( "4616", "Memory", "", "", "" )
concern( "4617", "Power", "", "", "" )
concern( "4618", "Select Band", "", "", "" )
concern( "4619", "Tune", "", "", "" )
concern( "21730", "Remember Settings", "", "", "" )

responsibility( "63", "band", "" )
responsibility( "101", "setBand(AM)", "" )
responsibility( "103", "setBand(FM)", "" )
responsibility( "140", "freqDown", "" )
responsibility( "142", "freqUp", "" )
responsibility( "146", "storeFreq", "" )
responsibility( "229", "search", "" )
responsibility( "332", "preset", "" )
responsibility( "371", "memory", "" )
responsibility( "694", "storeBand", "" )
responsibility( "3890", "checkSignal", "" )
responsibility( "6492", "power", "" )
responsibility( "10617", "setEnabled", "" )
responsibility( "10619", "setNotEnabled", "" )
responsibility( "12072", "setFreq(UP)", "" )
responsibility( "12942", "setBandToStored", "" )
responsibility( "12944", "setFreqToStored", "" )
responsibility( "13187", "rememberSettings", "" )
responsibility( "21473", "retrieveSettings", "" )
responsibility( "22332", "setFreq(DOWN)", "" )
responsibility( "22681", "activeAT", "" )
responsibility( "22683", "notActiveAT", "" )
responsibility( "23051", "standby", "" )
responsibility( "23536", "setStandbyOn", "" )
responsibility( "23538", "setStandbyOff", "" )
responsibility( "24456", "setFreq", "" )
responsibility( "24486", "adjustmentUp", "" )
responsibility( "24488", "adjustmentDown", "" )

map( "6350", "Power", "", "D:/Study/Winter 2014/Project/URN-Export/0502a/radio-solution-Map6350-Power.bmp", "Power", "Power" )
    respRef( "6493", "power", "", "6486", "6492" )
    respRef( "10618", "setEnabled", "", "6411", "10617" )
    respRef( "10620", "setNotEnabled", "", "6411", "10619" )
    stub( "22803", "Initialize Settings", "static" )
    stub( "22805", "Remember Settings", "static" )
    stub( "34651", "Abort Autotune", "static" )
    stub( "34942", "Abort Standby", "static" )
    compRef( "6411", "radio", "13", "" )
    compRef( "6486", "user", "11", "" )

scenario( "3794", "BandType", "" )
scenario( "3812", "TuneDirectionType", "" )

endImport
```

This script describes the radio-solution model is imported in DOORS, which includes two components, 7 concerns, 2 scenarios,28 responsibilities and 1 UCM objects. There is one image need to be imported in RMS which is exported by jUCMNav. The following table describes details of this image.

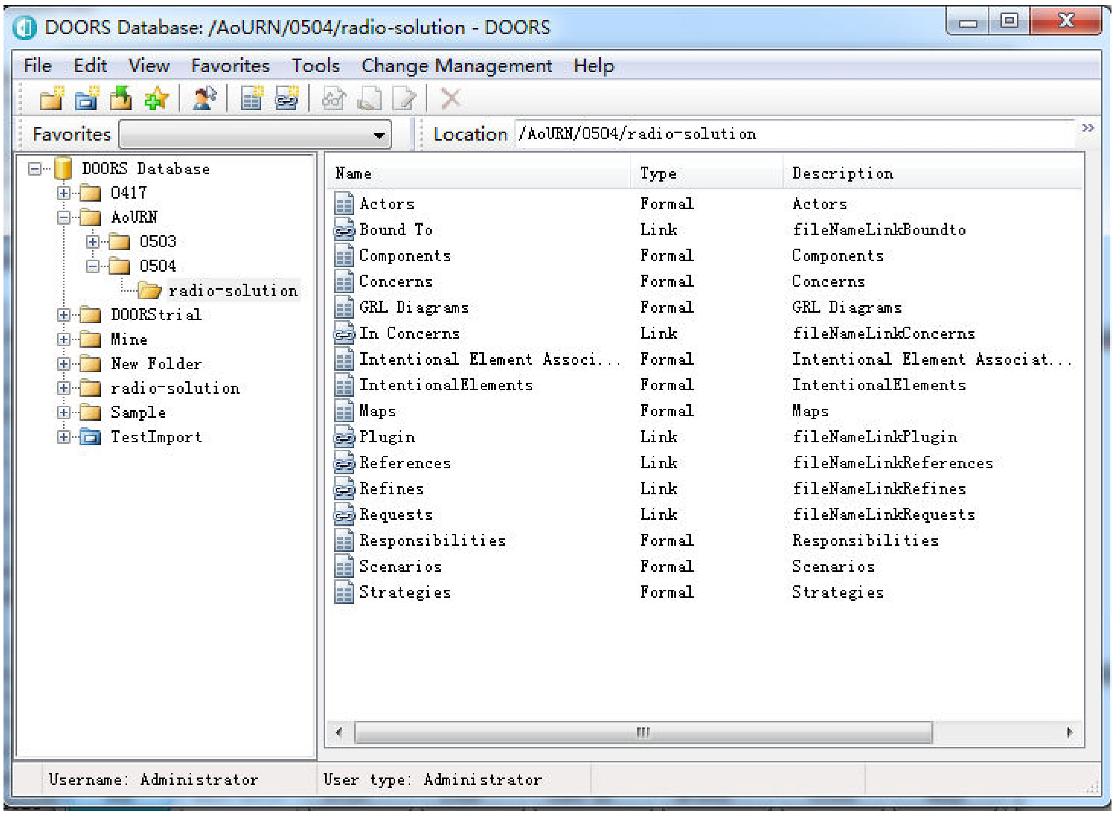
| | |
|---|---|
| AoURN object | map |

| | |
|---|---|
| ID | 6350 |
| Name | Power |
| Description | Null |
| Map File Name | radio-solution-Map6350-Power.bmp |
| Map Title | Power |
| Name of Concern | Power |

**Table 2** Details of map imported by DXL script for radio-solution model

### 4.2.2 Results

The following figure shows the DOORS formal and link modules which are imported from jUCMNav.

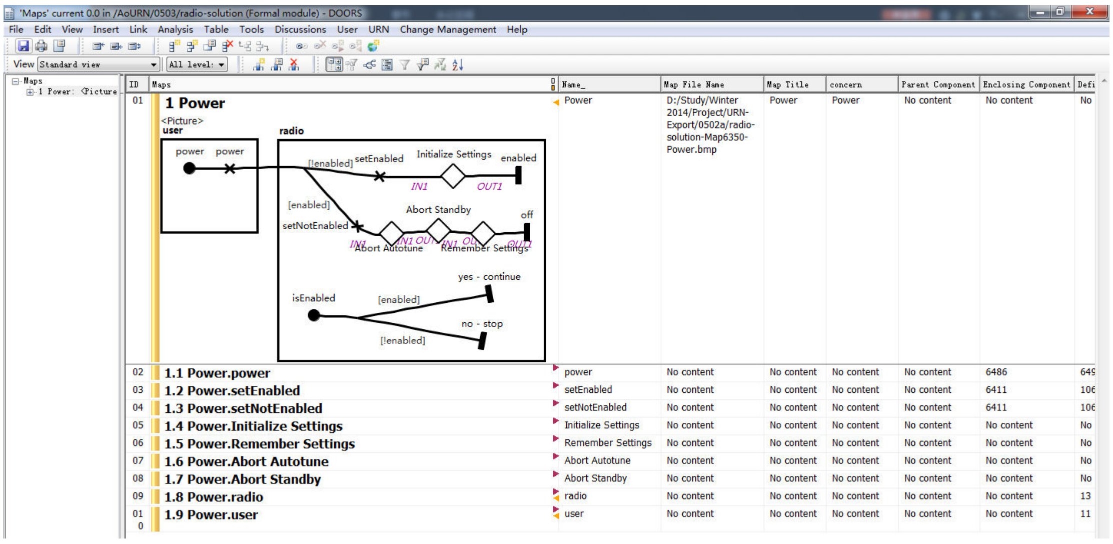

DOORS Database: /AoURN/0504/radio-solution - DOORS

| Name | Type | Description |
|---|---|---|
| Actors | Formal | Actors |
| Bound To | Link | fileNameLinkBoundto |
| Components | Formal | Components |
| Concerns | Formal | Concerns |
| GRL Diagrams | Formal | GRL Diagrams |
| In Concerns | Link | fileNameLinkConcerns |
| Intentional Element Associ... | Formal | Intentional Element Associat... |
| IntentionalElements | Formal | IntentionalElements |
| Maps | Formal | Maps |
| Plugin | Link | fileNameLinkPlugin |
| References | Link | fileNameLinkReferences |
| Refines | Link | fileNameLinkRefines |
| Requests | Link | fileNameLinkRequests |
| Responsibilities | Formal | Responsibilities |
| Scenarios | Formal | Scenarios |
| Strategies | Formal | Strategies |

Username: Administrator
User type: Administrator

**Figure 10** 'radio-solution' model imported in DOORS

The following figure shows the imported UCM Map module, which includes the content of the image file ‘radio-solution-Map6350-Power.bmp’. The new attribute: name of concern is imported as expected.


'Maps' current 0.0 in /AoURN/0503/radio-solution (Formal module) - DOORS

File Edit View Insert Link Analysis Table Tools Discussions User URN Change Management Help

View: Standard view | All level:

Maps
1 Power: <Picture

| ID | Maps | Name_ | Map File Name | Map Title | concern | Parent Component | Enclosing Component | Defi |
|---|---|---|---|---|---|---|---|---|
| 01 | 1 Power <Picture> | Power | D:/Study/Winter 2014/Project/URN-Export/0502a/radio-solution-Map6350-Power.bmp | Power | Power | No content | No content | No |
| 02 | 1.1 Power.power | power | No content | No content | No content | No content | 6486 | 649 |
| 03 | 1.2 Power.setEnabled | setEnabled | No content | No content | No content | No content | 6411 | 106 |
| 04 | 1.3 Power.setNotEnabled | setNotEnabled | No content | No content | No content | No content | 6411 | 106 |
| 05 | 1.4 Power.Initialize Settings | Initialize Settings | No content | No content | No content | No content | No content | No |
| 06 | 1.5 Power.Remember Settings | Remember Settings | No content | No content | No content | No content | No content | No |
| 07 | 1.6 Power.Abort Autotune | Abort Autotune | No content | No content | No content | No content | No content | No |
| 08 | 1.7 Power.Abort Standby | Abort Standby | No content | No content | No content | No content | No content | No |
| 09 | 1.8 Power.radio | radio | No content | No content | No content | No content | No content | 13 |
| 010 | 1.9 Power.user | user | No content | No content | No content | No content | No content | 11 |

**Figure 11** Map module from radio-solution model in DOORS

Note: There is no GRL Diagram from this example, we will display the results about GRL Diagram by the next experiment.

**Map (#46)**

By changing the DXL script of jUCMNav image ‘map’, we can import another map in DOORS. The second image is import is ‘radio-solution-Map46-Select Band.bmp’.

The DXL script related to this map is:

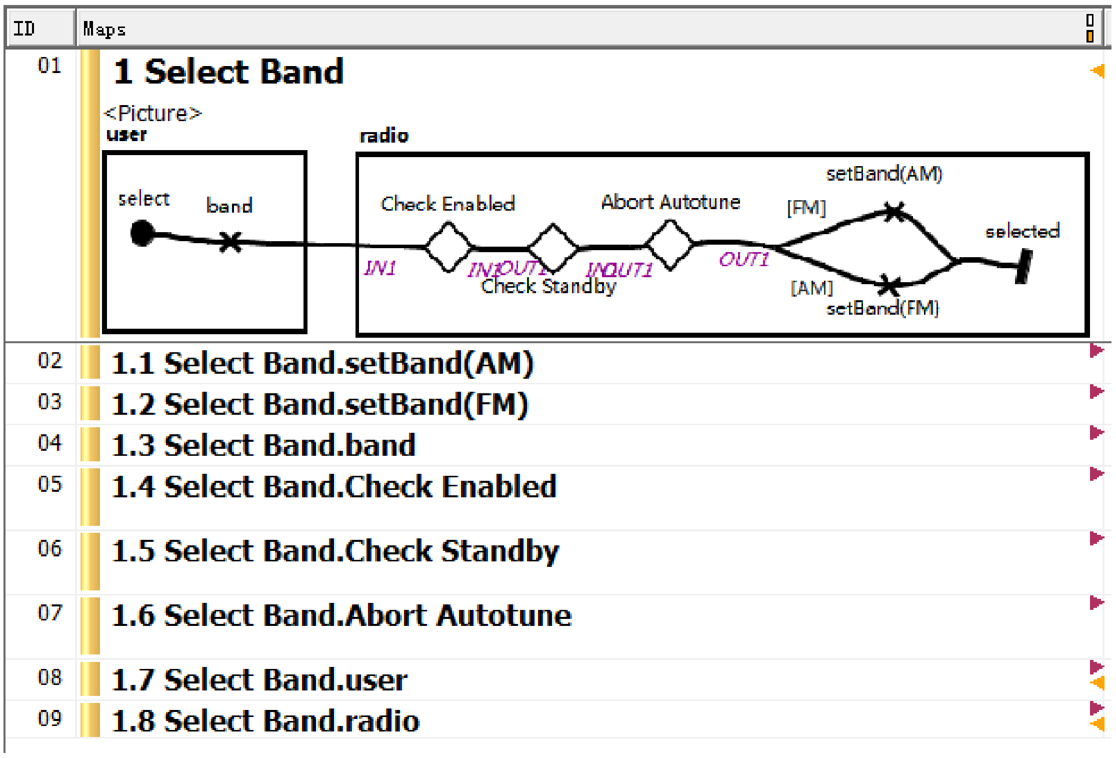
```
map( "46", "Select Band", "", "D:/Study/Winter 2014/Project/URN-Export/0502a/radio-solution-Map46-Select Band.bmp", "Select Band", "Select Band" )
    respRef( "102", "setBand(AM)", "", "50", "101" )
    respRef( "104", "setBand(FM)", "", "50", "103" )
    respRef( "28977", "band", "", "48", "63" )
    stub( "23556", "Check Enabled", "static" )
    stub( "28682", "Check Standby", "static" )
    stub( "33984", "Abort Autotune", "static" )
    compRef( "48", "user", "11", "" )
    compRef( "50", "radio", "13", "" )
```

This script describes the AoURN model is imported in DOORS, which includes two components, 7 concerns, 2 scenarios,28 responsibilities and 1 UCM objects. There is one image need to be imported in RMS which is exported by jUCMNav. The following table describes details of this image.

| | |
|---|---|
| AoURN object | map |
| ID | 46 |
| Name | Select Band |
| Description | Null |
| Map File Name | radio-solution-Map46-Select Band.bmp |
| Map Title | Select Band |
| ConcernID | 4618 |
| ConcernName | Select Band |

**Table 3** details of map imported by DXL script for radio-solution model

The following figure shows the imported UCM Map module, which includes the content of the image file 'radio-solution-Map46-Select Band.bmp'. The new attribute: name of 'Concern' is imported as expected.



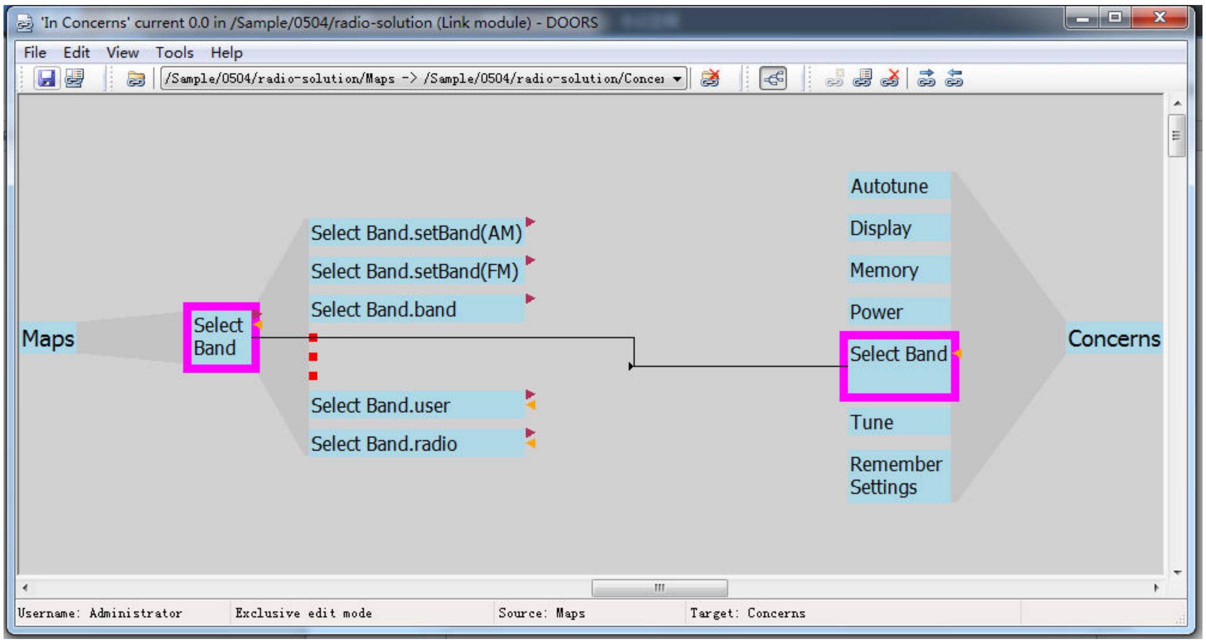
| ID | Maps |
|---|---|
| 01 | **1 Select Band** <Picture> |
| 02 | **1.1 Select Band.setBand(AM)** |
| 03 | **1.2 Select Band.setBand(FM)** |
| 04 | **1.3 Select Band.band** |
| 05 | **1.4 Select Band.Check Enabled** |
| 06 | **1.5 Select Band.Check Standby** |
| 07 | **1.6 Select Band.Abort Autotune** |
| 08 | **1.7 Select Band.user** |
| 09 | **1.8 Select Band.radio** |

**Figure 12** Map module from radio-solution model in DOORS

**New link: In Concern**

By using the MT-DSL from Appendix A, we can track and import 'Concern' in DOORS, we also can trace the link between the module 'Concern' and module 'map/grlDdiagram'. The following figure displays the link 'In Concern' between concern 'Select Band' and map 'radio-solution-Map46-Select Band.bmp' in radio-solution model.

"



Username: Administrator | Exclusive edit mode | Source: Maps | Target: Concerns

**Figure 13** Link 'In Concern' from Map(#46)

### 4.2.3 Re-importing Changes

We need to validate re-importing in DOORS since many DXL functions need to support the modification of models.

#### a. Re-importing Content Changes in Map Files

In this section, we will re-import the models with changes in the content of an imported map in DOORS. There is no GRL Diagram in radio-solution model, so we will choose Map(#46) as test model to validate. Table 4 shows changes in this model to validate this test scenario.

In this example, we edited serval parts of map 'radio-solution-Map6350-Power.bmp' in radio-solution model, then re-imported them in DOORS. As shown in Table 4, the name of map, name of respRef, title of Map and the name of Concern are changed.

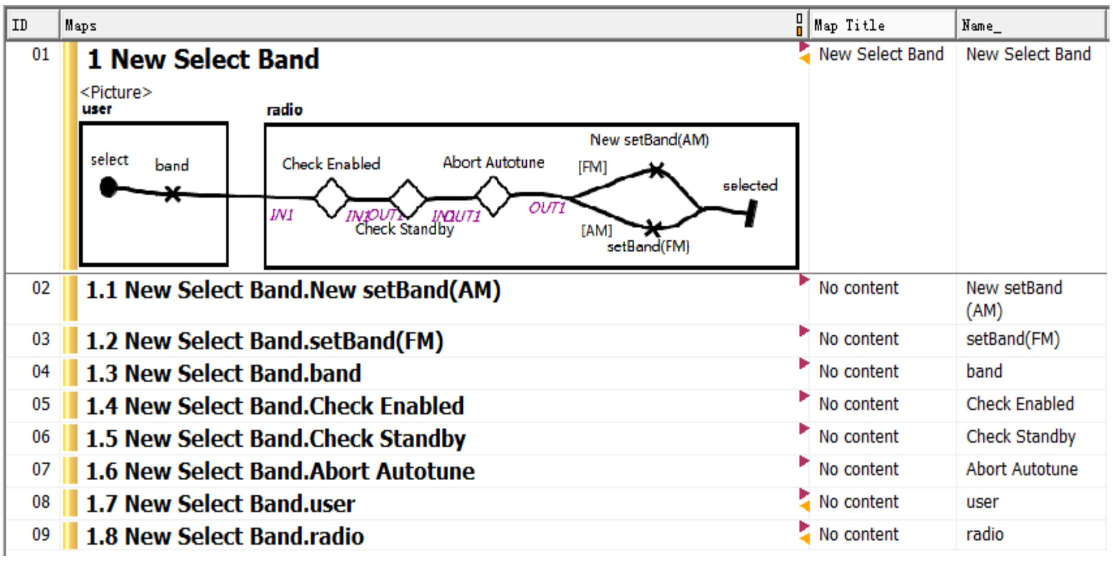
Because there are only Maps as images in this example model, we will edited some attributes of one Map and re-import the content with changes in DOORS.

| Model(ID) | Edited Attribute | Previous value | New value |
|---|---|---|---|

| | | | |
|---|---|---|---|
| Map(#6350) | Name_ | Select Band | New Select Band |
| | Name_<br><br>respRef(102) | setBand(AM) | New setBand(AM) |
| | Map Title | Select Band | New Select Band |

**Table 4** Changes of Map(#46) for re-importing in DOORS

The following figure shows the Map content after re-imported. This result proves that images can be modified and re-imported in DOORS.



| ID | Maps | Map Title | Name_ |
|---|---|---|---|
| 01 | **1 New Select Band**<br><Picture> | New Select Band | New Select Band |
| 02 | **1.1 New Select Band.New setBand(AM)** | No content | New setBand (AM) |
| 03 | **1.2 New Select Band.setBand(FM)** | No content | setBand(FM) |
| 04 | **1.3 New Select Band.band** | No content | band |
| 05 | **1.4 New Select Band.Check Enabled** | No content | Check Enabled |
| 06 | **1.5 New Select Band.Check Standby** | No content | Check Standby |
| 07 | **1.6 New Select Band.Abort Autotune** | No content | Abort Autotune |
| 08 | **1.7 New Select Band.user** | No content | user |
| 09 | **1.8 New Select Band.radio** | No content | radio |

**Figure 14**Modified image content for Map after re-importing

**b. Re-importing Models with modifying Links**

1). In this test, we re-import the model after deleting a link: ‘Reference’ from he DXL script. Table 5 shows the related change in radio-solution model, which is used to validate this test scenario.

We tested the link from respRef in Map,

The test about intentionalElementRef from GRL Diagram will be given in the next section.

| Model(#ID) | Edited Attribute | Previous value | New value |
|---|---|---|---|
| respRef(#102) | Definition ID | 101 | ‘’ |

**Table 5** Change to radio-solution model to validate re-importing changes in DOORS with deleting a link

After we delete the attribute, the DXL script which is related to the change described in Table 5 is shown in Table 6.

| DXL before change | DXL after change |
|---|---|
| responsibility( "101", "New setBand(AM)", "" )<br><br>respRef( "102", "New setBand(AM)", "", "50", **"101"** ) | responsibility( "101", "New setBand(AM)", "" )<br><br>respRef( "102", "New setBand(AM)", "", "50", **""** ) |

**Table 6** Changes to DXL scripts to validate re-importing changes in DOORS with deleting a link

Figure 15 shows the links ‘References’between Maps and Responsibilities in the original model (before changed), which is related the DXL script before changed.

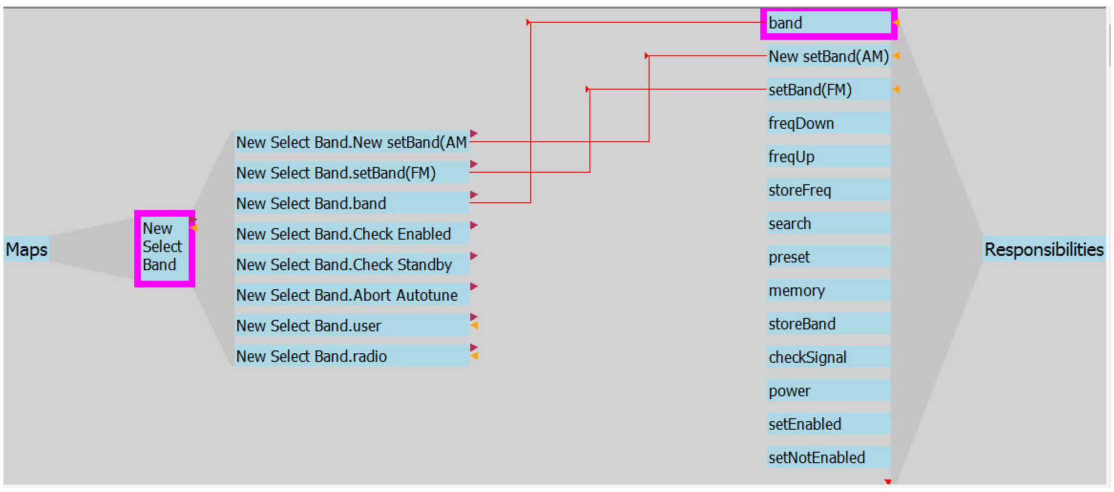


**Figure 15**‘References’links between Maps and Responsibilities in original

radio-solution model

Figure 16 shows the links "References" between Maps and Responsibilities in the new radio-solution model (after change). There is no link from respRef whose ID=102, which is related to the DXL script after change.

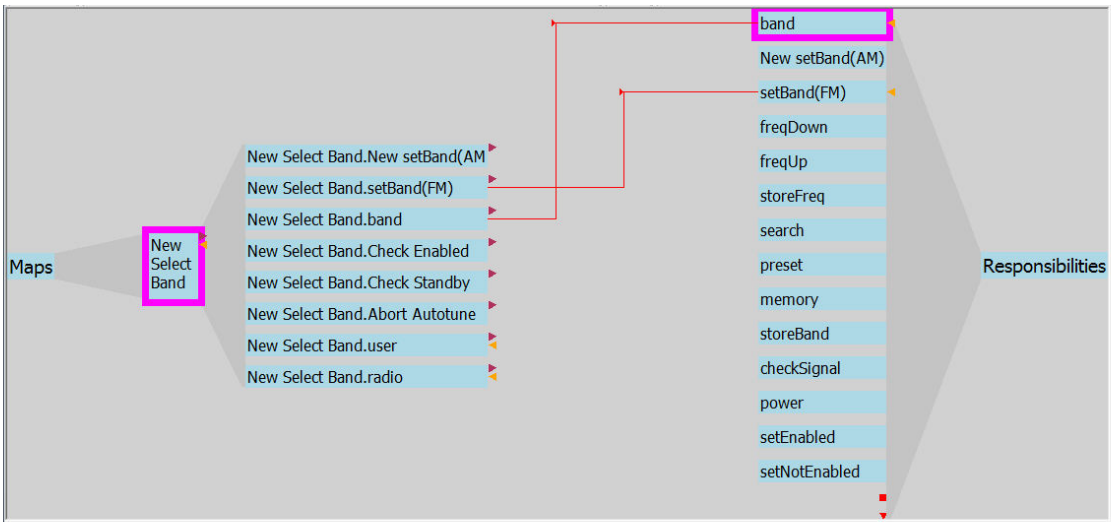


**Figure 16** 'References'links between Maps and Responsibilities in new radio-solution model

2). Similarly, we re-import the model after delete a link: 'In Concern'. Table 7 shows the related changes in the radio-solution model, which is used to validate this test scenario. Table 8 shows the related changes in DXL script.

| Model(#ID) | Edited Attribute | Previous value | New value |
|---|---|---|---|
| map(#46) | ConcernID | 4618 | '' |
| | ConcernName | Select Band | '' |

**Table 7** Change to radio-solution model to validate re-importing changes in DOORS with modifying a link

| DXL before change | DXL after change |
|---|---|
| concern( "4618", "Select Band", "", "", "" ) | concern( "4618", "Select Band", "", "", "" ) |

| | |
|---|---|
| concern( "21730", "Remember Settings", "", "", "" ) map( "46", "Select Band", "", "D:/Study/Winter2014/Project/URN-Export/0504a/radio-solution-Map46-Select Band.bmp", "Select Band", **"4618"**, "**Select Band**" ) | concern( "21730", "Remember Settings", "", "", "" ) map( "46", "Select Band", "", "D:/Study/Winter2014/Project/URN-Export/0504a/radio-solution-Map46-Select Band.bmp", "Select Band", **""**, **""** ) |

**Table 8** Changes to DXL scripts to validate re-importing changes in DOORS with modifying a link

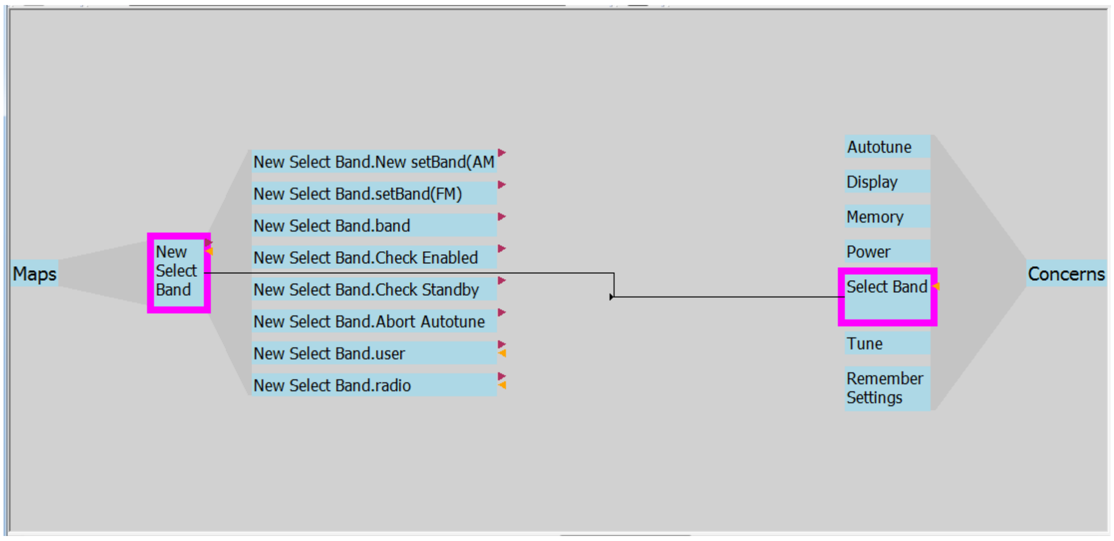


**Figure 17** “In Concern” links between Maps and Concerns in original radio-solution model

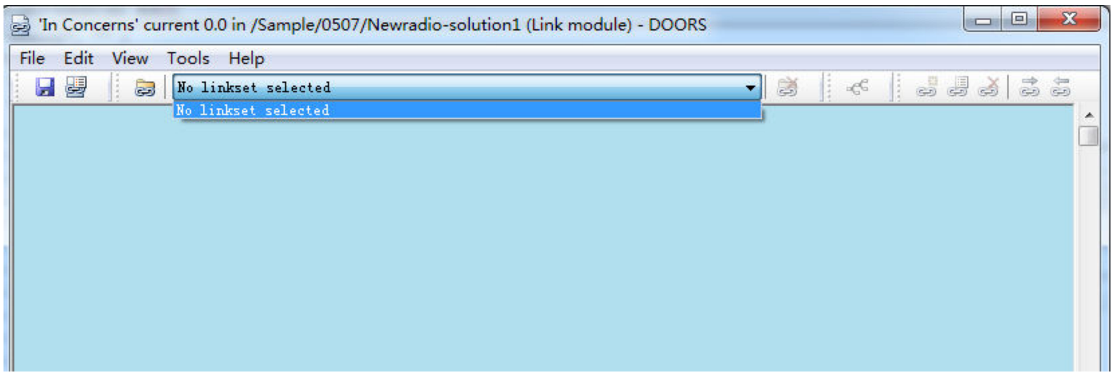

'In Concerns' current 0.0 in /Sample/0507/Newradio-solution1 (Link module) - DOORS

**Figure 18** ‘In Concern’ links does not exist between Maps and Concerns after re-importing

## 4.3CCCMS Models

### 4.3.1 Test scenario

The full version of DXL script is in the folder ‘exported DXL scripts and images’, which contains 2 Actors, 33 Components, 12 Concerns, 81 Responsibilities, 59 intentionalElements, 2 scenarios, 2 GRLDiagrams and 65 Maps.

The following DXL script describes the image part of the script, which is used to invoke the generated library and to import a sample of CCCMS model in DOORS, which includes one Map, one GRL Diagram and all other modules from the full version..

```
grldiagram( "60042", "Concern Interaction Graph", "", "D:/Study/Winter 2014/Project/URN-Export/0502a/CCCMS-GRLGraph60042-Concern Interaction Graph.bmp", "Concern Interaction Graph", "", "" )
    intentionalElementRef( "60044", "Resolve Crisis Concern", "", "", "60043" )
    intentionalElementRef( "60048", "Capture Witness Report Concern", "", "", "60047" )
    intentionalElementRef( "60054", "Recommend Strategies Concern", "", "", "60053" )
    intentionalElementRef( "60098", "New Crisis and Mission Info Concern", "", "", "60097" )
    intentionalElementRef( "87173", "Communicate with Resource at Location Concern", "", "", "87172" )
    intentionalElementRef( "87175", "Request Resources Concern", "", "", "87174" )
    intentionalElementRef( "87279", "Set Resource Status Concern", "", "", "87278" )
    intentionalElementRef( "91574", "Super Observer Status of Mission Concern", "", "", "91573" )
    intentionalElementRef( "91578", "Helicopter Mission Concern", "", "", "91577" )
    intentionalElementRef( "104795", "Communication Infrastructure Concern", "", "", "104895" )

map( "574", "Capture Witness Report", "", "D:/Study/Winter 2014/Project/URN-Export/0502a/CCCMS-Map574-Capture Witness Report.bmp", "Capture Witness Report", "34741", "Communication" )
    respRef( "581", "enterWitnessInfo", "first name, last name, phone number, and address", "575", "31" )
    respRef( "582", "provideCrisisFocusedChecklist", "", "576", "35" )
    respRef( "583", "enterCrisisInfo", "details about the crisis, the time witnessed, etc.", "575", "37" )
    respRef( "587", "provideWitnessAddressPhone", "", "577", "67" )
    respRef( "589", "validateWitnessInfo", "", "576", "113" )
    respRef( "591", "enterLocationAndType", "", "575", "33" )
    respRef( "601", "retrieveVideoFeed", "", "578", "332" )
    respRef( "602", "displayVideoFeed", "", "576", "334" )
    stub( "59702", "ResolveCrisis", "dynamic" )
    compRef( "575", "Coordinator", "47", "" )
    compRef( "576", "System", "49", "" )
    compRef( "577", "Phone Company", "51", "" )
    compRef( "578", "Surveillance System", "328", "" )
```

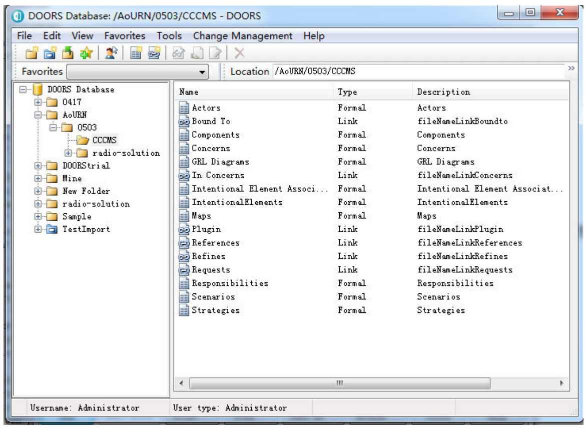
Table 9 shows the two images (exported by jUCMNav) which need to be imported in DOORS.

| AoURN object | map | GRL Diagram |
|---|---|---|
| ID | 574 | 60042 |
| Name | Capture Witness Report | Concern Interaction Graph |
| Description | Null | Null |
| Image File Name | CCCMS-Map574-Capture Witness Report.bmp | CCCMS-GRLGraph60042 -Concern Interaction Graph.bmp |
| Image Title | Select Band | Concern Interaction Graph |
| ConcernID | 34741 | Null |
| ConcernName | Communication | Null |

**Table 9** Images imported by DXL script for radio-solution model

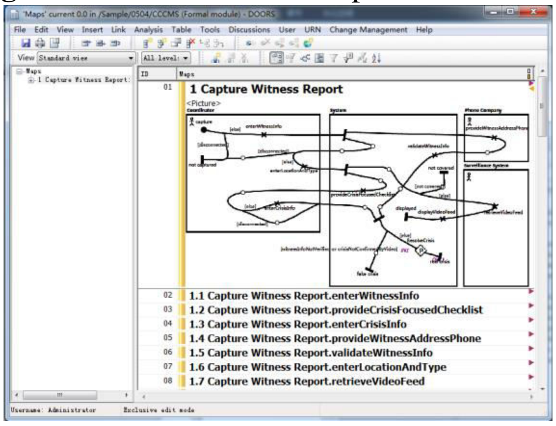
### 4.2.2 Results

The DOORS formal and link modules are the imported model from jUCMNav, they are shown in the following figure. The Figure 20 shows the imported Map module ,which shows the content of image file ‘CCCMS-Map574-Capture Witness Report.bmp’. Similarly, Figure 21 shows the imported GRL Diagram module in DOORS, which contains the content of image file ‘CCCMS-GRLGraph60042-Concern Interaction Graph.bmp’.



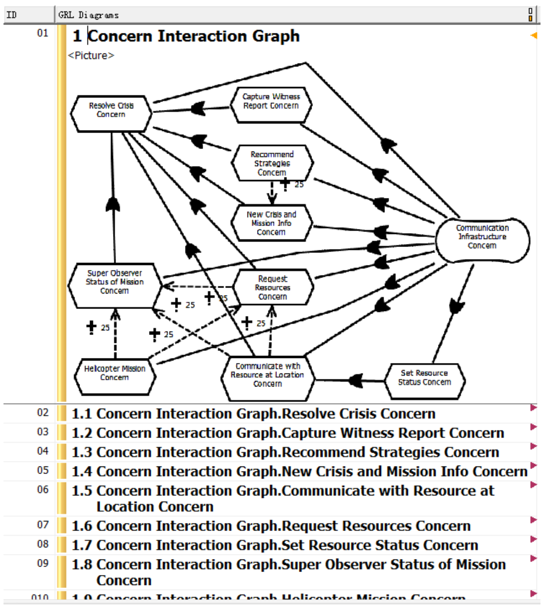
| Name | Type | Description |
|---|---|---|
| Actors | Formal | Actors |
| Bound To | Link | fileNameLinkBoundto |
| Components | Formal | Components |
| Concerns | Formal | Concerns |
| GRL Diagrams | Formal | GRL Diagrams |
| In Concerns | Link | fileNameLinkConcerns |
| Intentional Element Associ... | Formal | Intentional Element Associat... |
| IntentionalElements | Formal | IntentionalElements |
| Maps | Formal | Maps |
| Plugin | Link | fileNameLinkPlugin |
| References | Link | fileNameLinkReferences |
| Refines | Link | fileNameLinkRefines |
| Requests | Link | fileNameLinkRequests |
| Responsibilities | Formal | Responsibilities |
| Scenarios | Formal | Scenarios |
| Strategies | Formal | Strategies |

**Figure 19** CCCMS model imported in DOORS


'Maps' current 0.0 in /Sample/0504/CCCMS (Formal module) - DOORS
File Edit View Insert Link Analysis Table Tools Discussions User URN Change Management Help
View Standard view
All levels
Maps
1 Capture Witness Report
Username: Administrator
Exclusive edit mode

| ID | Maps |
|---|---|
| 01 | 1 Capture Witness Report<br><Picture> |
| 02 | 1.1 Capture Witness Report.enterWitnessInfo |
| 03 | 1.2 Capture Witness Report.provideCrisisFocusedChecklist |
| 04 | 1.3 Capture Witness Report.enterCrisisInfo |
| 05 | 1.4 Capture Witness Report.provideWitnessAddressPhone |
| 06 | 1.5 Capture Witness Report.validateWitnessInfo |
| 07 | 1.6 Capture Witness Report.enterLocationAndType |
| 08 | 1.7 Capture Witness Report.retrieveVideoFeed |

**Figure 20** Map module of CCCMS model imported in DOORS


Resolve Crisis Concern
Capture Witness Report Concern
Recommend Strategies Concern
New Crisis and Mission Info Concern
Communication Infrastructure Concern
Super Observer Status of Mission Concern
Request Resources Concern
Helicopter Mission Concern
Communicate with Resource at Location Concern
Set Resource Status Concern
25

| ID | GRL Diagrams |
|---|---|
| 01 | 1 Concern Interaction Graph<br><Picture> |
| 02 | 1.1 Concern Interaction Graph.Resolve Crisis Concern |
| 03 | 1.2 Concern Interaction Graph.Capture Witness Report Concern |
| 04 | 1.3 Concern Interaction Graph.Recommend Strategies Concern |
| 05 | 1.4 Concern Interaction Graph.New Crisis and Mission Info Concern |
| 06 | 1.5 Concern Interaction Graph.Communicate with Resource at Location Concern |
| 07 | 1.6 Concern Interaction Graph.Request Resources Concern |
| 08 | 1.7 Concern Interaction Graph.Set Resource Status Concern |
| 09 | 1.8 Concern Interaction Graph.Super Observer Status of Mission Concern |
| 010 | 1.9 Concern Interaction Graph.Helicopter Mission Concern |

**Figure 21** GRL diagram module of CCCMS model imported in DOORS

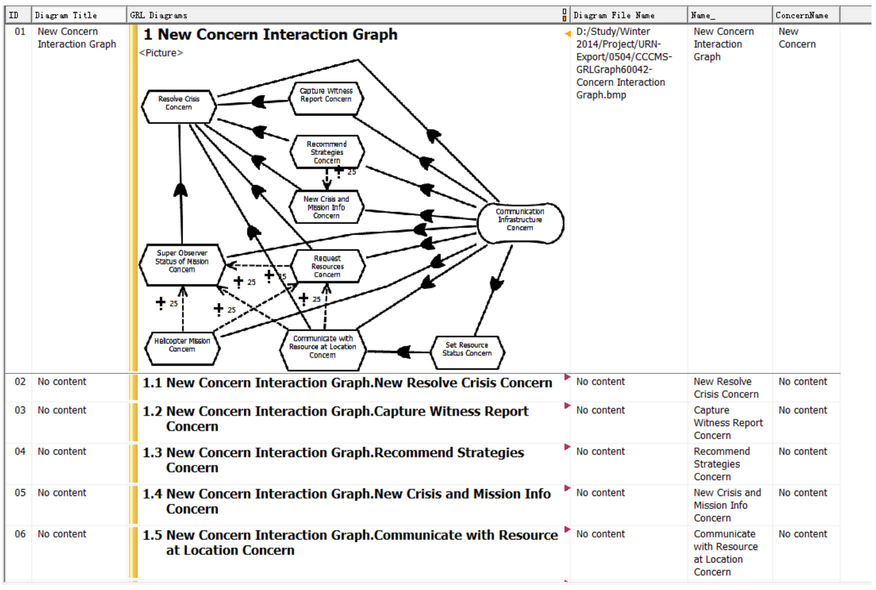
### 4.3.3 Re-import Changes

#### a. Re-importing Content Changes in GRL Diagram

In previous section, we validated re-importing in DOORS with changing attributes such as name, title and concern in Map, we will validate the similar re-importing with changing attributes in GRL Diagram from CCCMS model.

| Model(ID) | Edited Attribute | Previous value | New value |
|---|---|---|---|
| GRLdiagram(#60042) | Name_ | Concern Interaction Graph | New Concern Interaction Graph |
| | Name_ intentionalElementRef(60044) | Resolve Crisis Concern | New Resolve Crisis Concern |
| | Diagram Title | Concern Interaction Graph | New Concern Interaction Graph |

**Table 10** Changes of GRL Diagram(#574) for re-importing in DOORS



Resolve Crisis Concern
Capture Witness Report Concern
Recommend Strategies Concern
25
New Crisis and Mission Info Concern
Communication Infrastructure Concern
Super Observer Status of Mission Concern
Request Resources Concern
25
25
25
25
25
Helicopter Mission Concern
Communicate with Resource at Location Concern
Set Resource Status Concern

| ID | Diagram Title | GRL Diagrams | Diagram File Name | Name_ | ConcernName |
|---|---|---|---|---|---|
| 01 | New Concern Interaction Graph | **1 New Concern Interaction Graph** <Picture> | D:/Study/Winter 2014/Project/URN-Export/0504/CCCMS-GRLGraph60042-Concern Interaction Graph.bmp | New Concern Interaction Graph | New Concern |
| 02 | No content | **1.1 New Concern Interaction Graph.New Resolve Crisis Concern** | No content | New Resolve Crisis Concern | No content |
| 03 | No content | **1.2 New Concern Interaction Graph.Capture Witness Report Concern** | No content | Capture Witness Report Concern | No content |
| 04 | No content | **1.3 New Concern Interaction Graph.Recommend Strategies Concern** | No content | Recommend Strategies Concern | No content |
| 05 | No content | **1.4 New Concern Interaction Graph.New Crisis and Mission Info Concern** | No content | New Crisis and Mission Info Concern | No content |
| 06 | No content | **1.5 New Concern Interaction Graph.Communicate with Resource at Location Concern** | No content | Communicate with Resource at Location Concern | No content |

**Figure 22** Modified image file content for GRL Diagram after re-importing

**b. Re-importing Models with modifying Links**

1). In this test, we re-import the model with a deleted link: reference using the DXL script. Table 11 shows the related change in the CCCMS model, which is used to validate this test scenario.

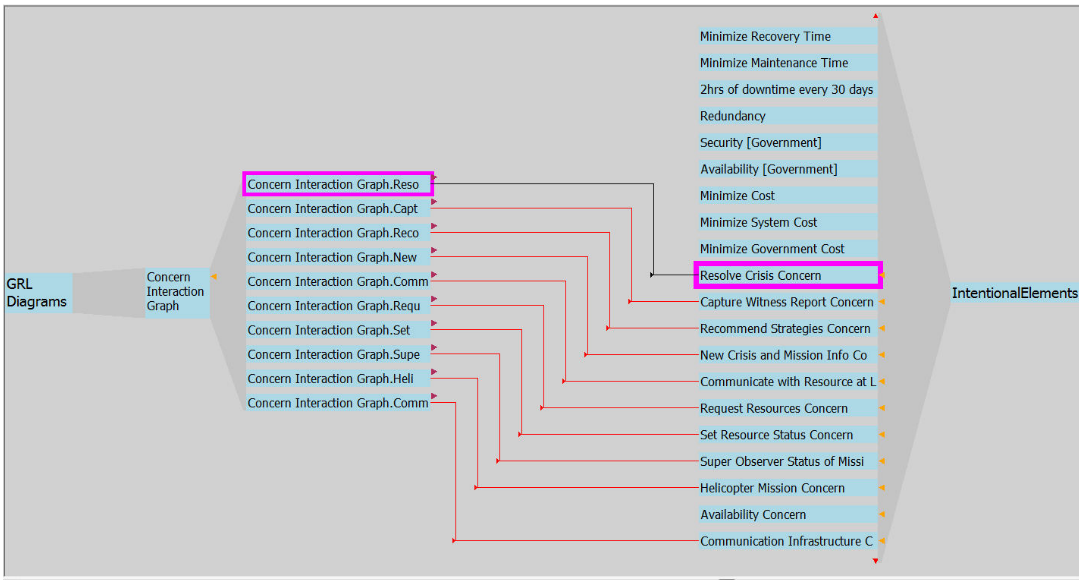
| Model(#ID) | Edited Attribute | Previous value | New value |
|---|---|---|---|
| intentionalElement Ref (#60044) | Definition ID | 60043 | ‘’ |

**Table 11** Change to CCCMS model to validate re-importing changes in DOORS with deleting a link

| DXL before change | DXL after change |
|---|---|
| intentionalElement( "60043", "Resolve Crisis Concern", "", "Task", "And", "None", 0 )<br>intentionalElementRef( "60044", "Resolve Crisis Concern", "", "", **"60043"** ) | intentionalElement( "60043", "Resolve Crisis Concern", "", "Task", "And", "None", 0 )<br>intentionalElementRef( "60044", "Resolve Crisis Concern", "", "", **""** ) |

**Table 12** Change to DXL scripts to validate re-importing changes in DOORS with deleting a link

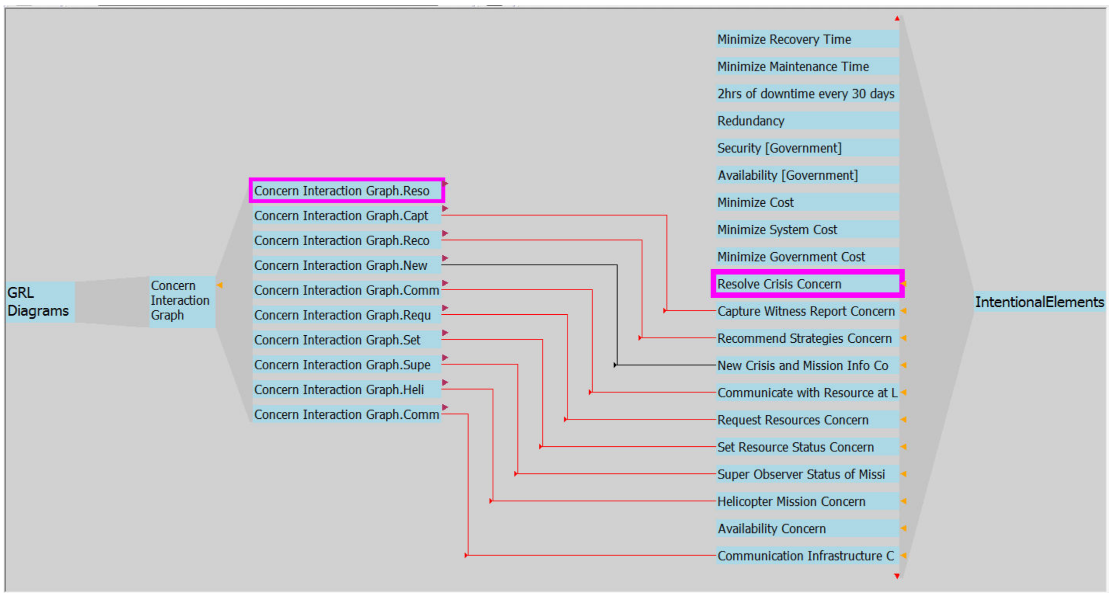
Figure 23 shows the links ‘References’ between GRL Diagram and IntentionalElement in the original model (before change), which is related the DXL script before change.



GRL Diagrams
Concern Interaction Graph
Concern Interaction Graph.Reso
Concern Interaction Graph.Capt
Concern Interaction Graph.Reco
Concern Interaction Graph.New
Concern Interaction Graph.Comm
Concern Interaction Graph.Requ
Concern Interaction Graph.Set
Concern Interaction Graph.Supe
Concern Interaction Graph.Heli
Concern Interaction Graph.Comm
Minimize Recovery Time
Minimize Maintenance Time
2hrs of downtime every 30 days
Redundancy
Security [Government]
Availability [Government]
Minimize Cost
Minimize System Cost
Minimize Government Cost
Resolve Crisis Concern
Capture Witness Report Concern
Recommend Strategies Concern
New Crisis and Mission Info Co
Communicate with Resource at L
Request Resources Concern
Set Resource Status Concern
Super Observer Status of Missi
Helicopter Mission Concern
Availability Concern
Communication Infrastructure C
IntentionalElements

**Figure 23** 'References' links between GRL Diagram and IntentionalElements in original CCCMS model

Figure 24 shows the links "References" between GRL Diagram and IntentionalElements in the new CCCMS model (after change). There is no link from intentionElementRef whose ID=60044, which is related to the DXL script after change.



GRL Diagrams
Concern Interaction Graph
Concern Interaction Graph.Reso
Concern Interaction Graph.Capt
Concern Interaction Graph.Reco
Concern Interaction Graph.New
Concern Interaction Graph.Comm
Concern Interaction Graph.Requ
Concern Interaction Graph.Set
Concern Interaction Graph.Supe
Concern Interaction Graph.Heli
Concern Interaction Graph.Comm
Minimize Recovery Time
Minimize Maintenance Time
2hrs of downtime every 30 days
Redundancy
Security [Government]
Availability [Government]
Minimize Cost
Minimize System Cost
Minimize Government Cost
Resolve Crisis Concern
Capture Witness Report Concern
Recommend Strategies Concern
New Crisis and Mission Info Co
Communicate with Resource at L
Request Resources Concern
Set Resource Status Concern
Super Observer Status of Missi
Helicopter Mission Concern
Availability Concern
Communication Infrastructure C
IntentionalElements

**Figure 24** 'References'links between GRL Diagram and IntentionalElements in CCCMS model after change

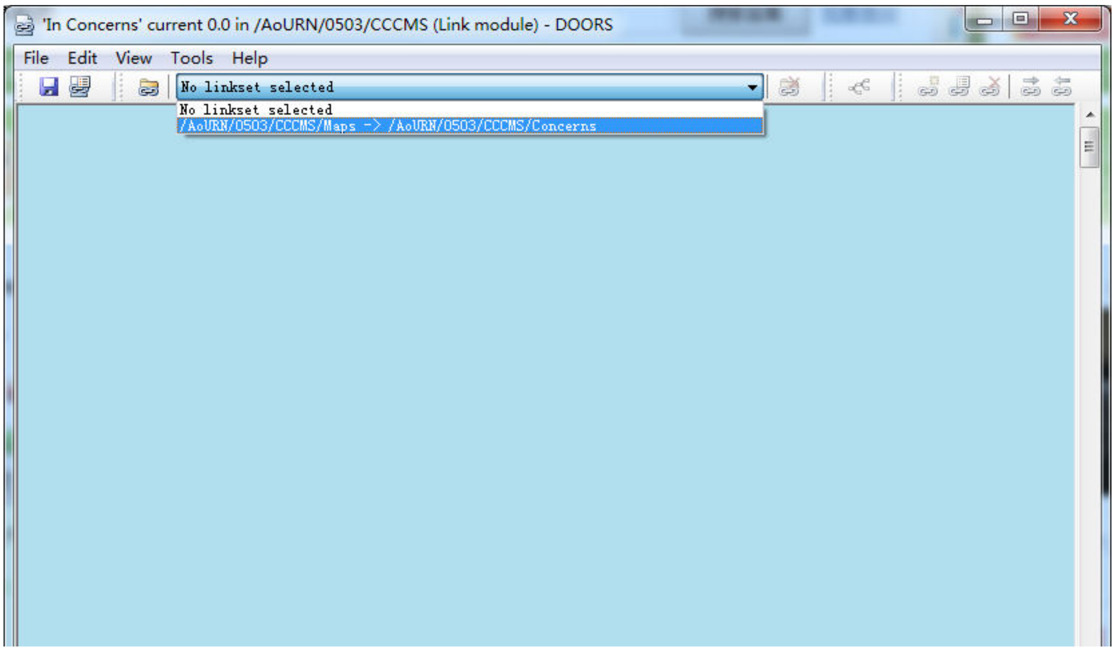
2). In previous section, changed the ConcernID which is related to the Map in radio-solution model and validate the re-importing. For the CCCMS model, we will add a new link between GRL Diagram and Concern. The following table shows another change with link, this time we use link 'In Concern'

| Model(#ID) | Edited Attribute | Previous value | New value |
|---|---|---|---|
| GRL Diagram(#46) | ConcernID | '' | 56337 |
| | ConcernName | '' | Availability |

**Table 13** Change to radio-solution model to validate re-importing changes in DOORS with deleting a link

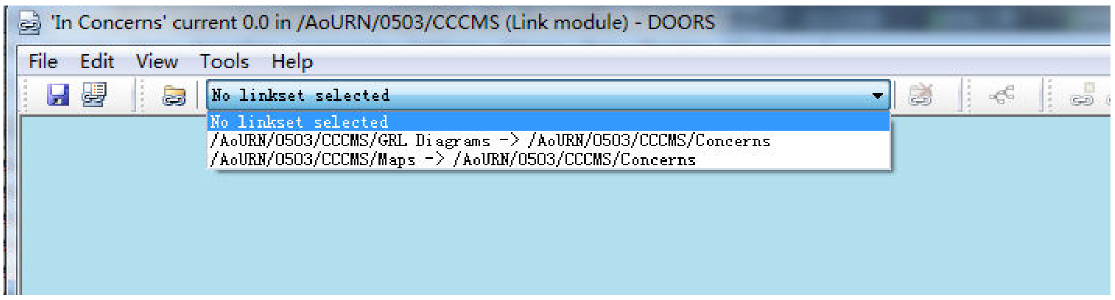
| DXL before change | DXL after change |
|---|---|
| concern( "56337", "Availability", "", "", "" )<br>grldiagram( "60042", "Concern Interaction Graph", "", "D:/Study/Winter 2014/Project/URN-Export/0504/CCCMS -GRLGraph60042-Concern Interaction Graph.bmp", "Concern Interaction Graph", **""**, **""** ) | concern( "56337", "Availability", "", "", "" )<br>grldiagram( "60042", "Concern Interaction Graph", "", "D:/Study/Winter 2014/Project/URN-Export/0504/CCCMS -GRLGraph60042-Concern Interaction Graph.bmp", "Concern Interaction Graph", **"56337", "Availability"** ) |

**Table 14** Changes to DXL scripts to validate re-importing changes in DOORS with deleting a link


'In Concerns' current 0.0 in /AoURN/0503/CCCMS (Link module) - DOORS

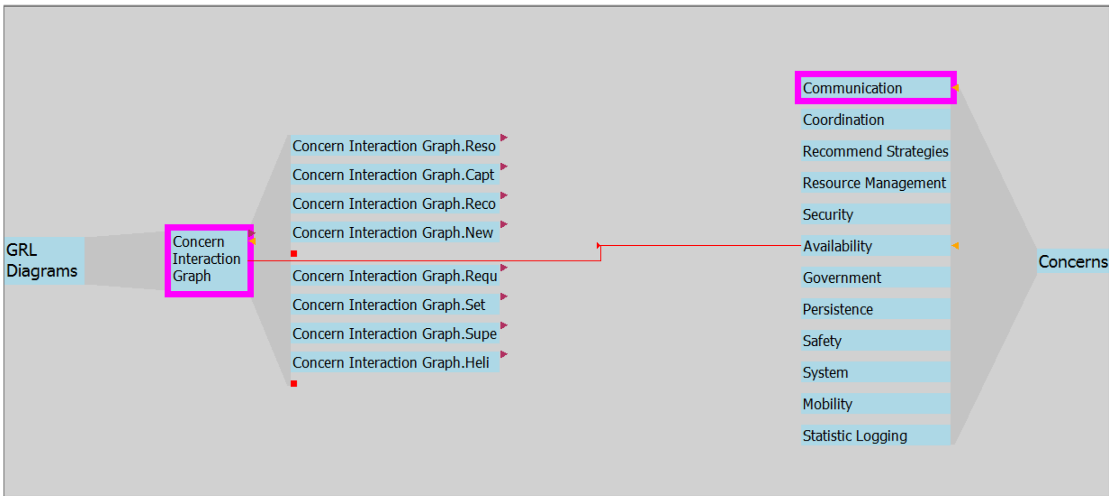
```
No linkset selected
No linkset selected
/AoURN/0503/CCCMS/Maps -> /AoURN/0503/CCCMS/Concerns
```

**Figure 25** 'In Concern' link in CCCMS model before re-importing with changes


'In Concerns' current 0.0 in /AoURN/0503/CCCMS (Link module) - DOORS

```
No linkset selected
No linkset selected
/AoURN/0503/CCCMS/GRL Diagrams -> /AoURN/0503/CCCMS/Concerns
/AoURN/0503/CCCMS/Maps -> /AoURN/0503/CCCMS/Concerns
```

**Figure 26** 'In Concern' link in CCCMS model after re-importing with changes



GRL Diagrams
Concern Interaction Graph
Concern Interaction Graph.Reso
Concern Interaction Graph.Capt
Concern Interaction Graph.Reco
Concern Interaction Graph.New
Concern Interaction Graph.Requ
Concern Interaction Graph.Set
Concern Interaction Graph.Supe
Concern Interaction Graph.Heli
Communication
Coordination
Recommend Strategies
Resource Management
Security
Availability
Government
Persistence
Safety
System
Mobility
Statistic Logging
Concerns

**Figure 27** 'In Concern' links between GRL Diagrams and Concerns in CCCMS model after change

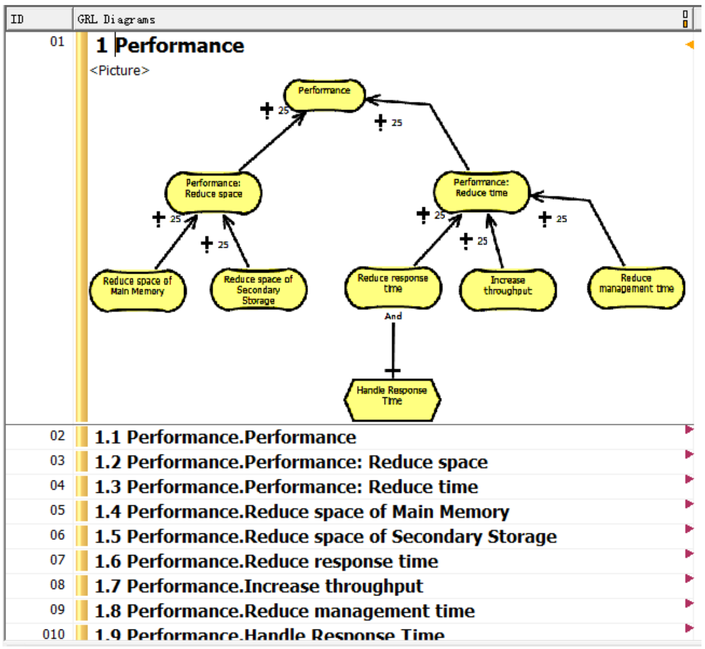
## 4.4 YouKeyKnowsv7 Models

### 4.4.1 Test Scenarios

This model contains 23 GRL Diagrams and 17 Maps, we use the following two image to import in DOORS.

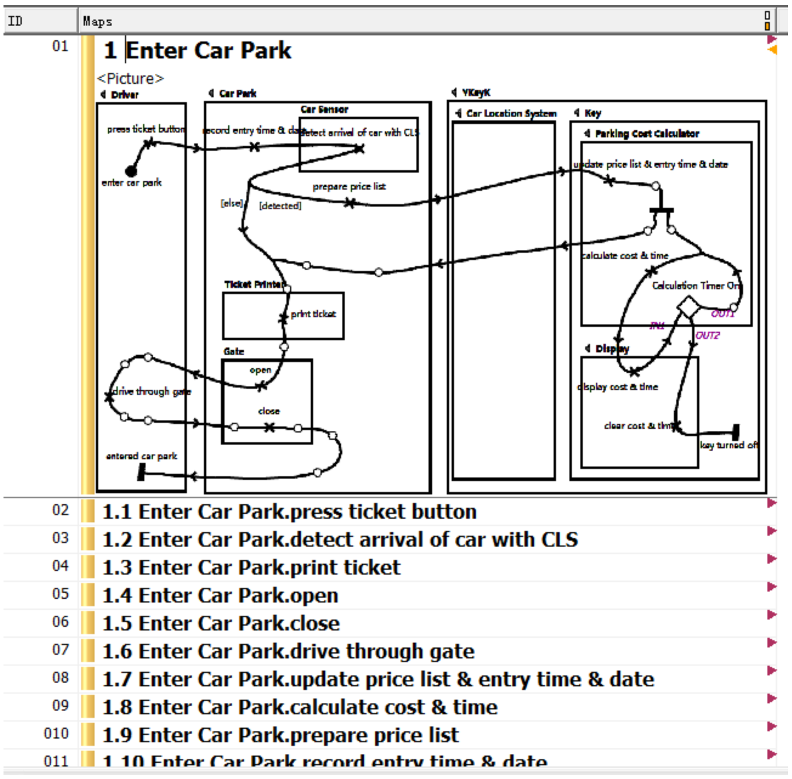
```
grldiagram( "423", "Performance", "", "D:/Study/Winter 2014/Project/URN-Export/0503/YourKeyKnowsv7-GRLGraph423-Performance.bmp
", "Performance", "Performance", "Performance" )
    intentionalElementRef( "14163", "Performance", "", "", "424" )
    intentionalElementRef( "14165", "Performance: Reduce space", "", "", "14164" )
    intentionalElementRef( "14167", "Performance: Reduce time", "", "", "14166" )
    intentionalElementRef( "14169", "Reduce space of Main Memory", "", "",
"14168" )
    intentionalElementRef( "14171", "Reduce space of Secondary Storage", "", "",
"14170" )
    intentionalElementRef( "14173", "Reduce response time", "", "", "14172" )
    intentionalElementRef( "14175", "Increase throughput", "", "", "14174" )
    intentionalElementRef( "14177", "Reduce management time", "", "", "14176" )
    intentionalElementRef( "14223", "Handle Response Time", "", "", "428" )

map( "7976", "Enter Car Park", "", "D:/Study/Winter 2014/Project/URN-Export/0503/YourKeyKnowsv7-Map7976-Enter Car Park.bmp",
"Enter Car Park", "10239", "UC001 Visit Car Park" )
    respRef( "8096", "press ticket button", "", "7978", "8095" )
    respRef( "8098", "detect arrival of car with CLS", "", "8118", "8097" )
    respRef( "8102", "print ticket", "", "8120", "8101" )
    respRef( "8104", "open", "", "8122", "8103" )
    respRef( "8106", "close", "", "8122", "8105" )
    respRef( "8108", "drive through gate", "", "7978", "8107" )
    respRef( "8112", "update price list & entry time & date", "", "8128", "8111" )
    respRef( "8114", "calculate cost & time", "", "8128", "8113" )
    respRef( "8116", "prepare price list", "", "7980", "8115" )
    respRef( "8449", "record entry time & date", "", "7980", "8099" )
    respRef( "8604", "display cost & time", "", "8130", "8603" )
    respRef( "8808", "clear cost & time", "", "8130", "8807" )
    stub( "9984", "Calculation Timer On", "static" )
    compRef( "7978", "Driver", "7977", "" )
    compRef( "7980", "Car Park", "7979", "" )
    compRef( "7984", "Key", "7983", "YKeyK" )
    compRef( "8118", "Car Sensor", "8117", "Car Park" )
    compRef( "8120", "Ticket Printer", "8119", "Car Park" )
    compRef( "8122", "Gate", "8121", "Car Park" )
    compRef( "8124", "Car Location System", "8123", "YKeyK" )
    compRef( "8128", "Parking Cost Calculator", "8127", "Key" )
    compRef( "8130", "Display", "8129", "Key" )
    compRef( "8132", "YKeyK", "8131", "" )
```

The results is shown in following figures:



| ID | GRL Diagrams |
|---|---|
| 01 | 1 Performance <Picture> |
| 02 | 1.1 Performance.Performance |
| 03 | 1.2 Performance.Performance: Reduce space |
| 04 | 1.3 Performance.Performance: Reduce time |
| 05 | 1.4 Performance.Reduce space of Main Memory |
| 06 | 1.5 Performance.Reduce space of Secondary Storage |
| 07 | 1.6 Performance.Reduce response time |
| 08 | 1.7 Performance.Increase throughput |
| 09 | 1.8 Performance.Reduce management time |
| 010 | 1.9 Performance.Handle Response Time |

**Figure 28** GRL Diagram in YouKeyKnowsv7 model



| ID | Maps |
|---|---|
| 01 | 1 Enter Car Park <Picture> |
| 02 | 1.1 Enter Car Park.press ticket button |
| 03 | 1.2 Enter Car Park.detect arrival of car with CLS |
| 04 | 1.3 Enter Car Park.print ticket |
| 05 | 1.4 Enter Car Park.open |
| 06 | 1.5 Enter Car Park.close |
| 07 | 1.6 Enter Car Park.drive through gate |
| 08 | 1.7 Enter Car Park.update price list & entry time & date |
| 09 | 1.8 Enter Car Park.calculate cost & time |
| 010 | 1.9 Enter Car Park.prepare price list |
| 011 | 1.10 Enter Car Park.record entry time & date |

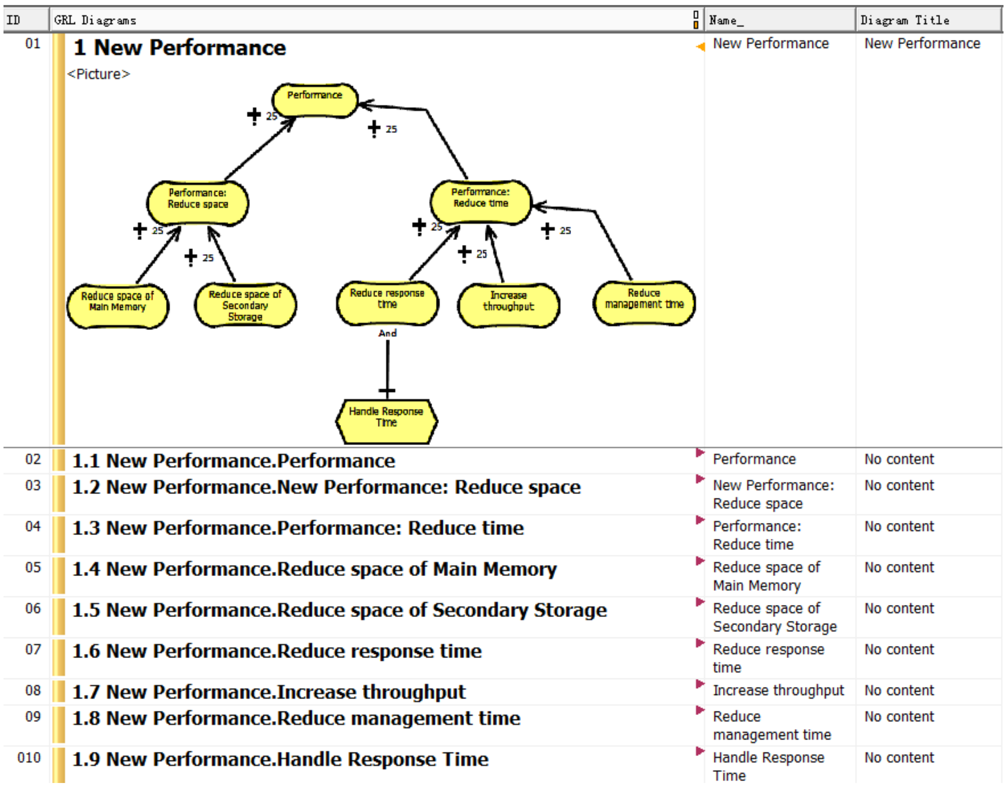
**Figure 29** Map in YouKeyKnowsv7 model

### 4.4.2 Re-importing Changes (Changing content in GRL Diagram)

We will validate the similar re-importing with changing attributes in GRL Diagram from CCCMS model.

| Model(ID) | Edited Attribute | Previous value | New value |
|---|---|---|---|
| GRLdiagram(#423) | Name_ | Performance | New Performance |
| | Name_<br><br>intentionalElement Ref(14165) | Performance:<br><br>Reduce space | New Performance:<br><br>Reduce space |
| | Diagram Title | Performance | New Performance |

**Table 15** Changes of GRL Diagram(#423) for re-importing in DOORS



| ID | GRL Diagrams | Name_ | Diagram Title |
|---|---|---|---|
| 01 | **1 New Performance**<br><Picture> | New Performance | New Performance |
| 02 | **1.1 New Performance.Performance** | Performance | No content |
| 03 | **1.2 New Performance.New Performance: Reduce space** | New Performance: Reduce space | No content |
| 04 | **1.3 New Performance.Performance: Reduce time** | Performance: Reduce time | No content |
| 05 | **1.4 New Performance.Reduce space of Main Memory** | Reduce space of Main Memory | No content |
| 06 | **1.5 New Performance.Reduce space of Secondary Storage** | Reduce space of Secondary Storage | No content |
| 07 | **1.6 New Performance.Reduce response time** | Reduce response time | No content |
| 08 | **1.7 New Performance.Increase throughput** | Increase throughput | No content |
| 09 | **1.8 New Performance.Reduce management time** | Reduce management time | No content |
| 010 | **1.9 New Performance.Handle Response Time** | Handle Response Time | No content |

**Figure 30** Modified image file content for GRL Diagram after re-importing

## 5. Bugs and Issues

1. The type of data includes boolean, however, it does not work. We need to use string instead of boolean in order to generate DXL library normally. For example, the attribute ‘Correlation’of module ‘Contribution’ should be boolean, but the MT-DSL has to define it as ‘string’ type.

2. The location of the image files, such as GRL Diagram or Map is an important issue, we can define this by changing the code of eclipse plugin: ExportDXL.java. For example: write("D:/Study/Winter 2014/Project/URN-Export/0502a/");

3. The Header need to be specified by using function writeHeader () in EportDXL.java. The header need to be defined as same as the location of DXL libraries.
For example: write("#include \"addins/dsl/lib//Utilities.dxl\"\n");

4. Importing GRL Diagram and Map to DOORS, the number is limited (it only allows one GRL Diagram and one Map to be imported into DOORS, otherwise the execution halted because of run-time error)

5. The invalid thread access happened to the example: YouKeyKnowsv7, the DXL file can be exported totally even though this error happened. There should be 23 GRL Diagram and 17 Map, however, we can only get 16 GRL Diagrams and 12 Maps (even though using the original eclipse plugin).

## 6. Structure of Eclipse plugin

According to the new requirement of traced modules and attributes, the eclipse plugin should be corresponding to the MT-DSL description. The structure of the new eclipse plugin shown as follow:

```
writeHeader(urn);
writeActors(urn);
writeComponents(urn);
writeConcerns(urn);
writeResponsibilities(urn);
```

```
            writeIntentionalElements(urn);
            writeElementLinks(urn);
            writeGrlDiagrams(urn, filename);
            writeMaps(urn, filename);
            writeScenarios(urn);
            writeStrategies(urn);
            writeFooter();
```

Using concerns as the example:

The ID, Name, Description, Condition Label and Expression of Concerns can be traced by using the following code. The related java files include: concern.java and condition.java.

```
//Concern (New)
    protected void writeConcerns(URNspec urn) throws IOException {
        for (Iterator iter = urn.getUrndef().getConcerns().iterator();
iter.hasNext();) {
            Concern concern = (Concern) iter.next();
            write("concern( "); //$NON-NLS-1$

            // ID
            write(QUOTES);
            write(concern.getId());
            write(QUOTES_COMMA);

            // Name
            write(QUOTES);
            escapeAndWrite(concern.getName());
            write(QUOTES_COMMA);

            // Description
            write(QUOTES);
            escapeAndWrite(concern.getDescription());
            write(QUOTES_COMMA);

            //Condition Label
            Condition condition = concern.getCondition();
            String label=condition==null ? "" : condition.getLabel();
            write(QUOTES);
            escapeAndWrite(label);
            write(QUOTES_COMMA);

            //Condition Expression
            String expression=condition==null ? "" :
condition.getExpression();
            write(QUOTES);
            escapeAndWrite(expression);
            write(QUOTES);
            write(END_ELEM);

        }
        write("\n"); //$NON-NLS-1$
    }
```

I also added a new association to both Map and GRL Diagram, in order to make them have the link to Concern.

Using GRL Diagram as example:

Note: the existing code such as ID, Name are now shown here.

The ConcernID is used to linked to module concern and the ConcernName is used to display the name of Concern which is more easy to understand.

The related java files are concern.java, `IURNDiagram.java.`

```
 protected void writeGrlDiagrams(URNspec urn, String filename) throws
IOException {
        for (Iterator iter =
urn.getUrndef().getSpecDiagrams().iterator(); iter.hasNext();) {
            IURNDiagram element = (IURNDiagram) iter.next();
            if (element instanceof GRLGraph) {
                GRLGraph grlgraph = (GRLGraph) element;

                // map
                write("grldiagram( "); //$NON-NLS-1$

                //ConcernID
                Concern grlConcern = grlgraph.getConcern();
                String grlConcernID=grlConcern==null ? "" :
grlConcern.getName();
                write(QUOTES);
                write(grlConcernID);
                write(QUOTES_COMMA);

                //ConcernName
                String grlConcern1=grlConcern==null ? "" :
```

```
grlConcern.getName();
                write(QUOTES);
                write(grlConcern1);
                write(QUOTES);
                write(END_ELEM);

                writeActorRef(grlgraph);
                writeGrlNodes(grlgraph);
            }
        }
        write("\n\n"); //$NON-NLS-1$
    }
```

# 7. Conclusion

## 7.1 Contributions

Created two Two MT-DSL models and demonstrated them in the report with short explanation.

By using the new version MT-DSL description and related eclipse plugin, I did experiments and validations of one URN model (HellWorld) and three AoURN models (radio-solution, CCCMS and YourKeyKnowsv7)

## 7.2 Lessons learned

By doing the project this semester, I got good understanding of DSLs and requirements management (Rational DOORS). I also got a better mastering of Java, DXL, URN, AoURN and UML. Plus I did some validation and experiments of model transformation.

## 7.3 Future work

I think this MT-DSL model should be validated more since there have been some bugs found already, such as the type of boolean cannot be used. More than one GRL Diagram or Map can not be imported in DOORS (the run-time error will happen if do to so).

**Appendix A:**

This appendix describes the subset of the AoURN models.

```
model AoURNModel{
	folder AoURNModel{
		module responsibility{
			//default name, id, description
			class responsibility{
			}
		}
		module concern{
			 class concern{
				string "condition label" shows as "Condition Label"
				string "condition expression" shows as "Condition Expression"
			 }
		}

		module ignoreInReport actor{
			class actor{
				string "importanceType" shows as "importance Type"
				string "importanceQuantitative" shows as "importance Quantitative"
			}
		}
		module ignoreInReport intentionalElement{
			fileName "Intentional Elements"
			class intentionalElement{
				string "type"
				string "importanceQuantitative" shows as "importance Quantitative"
				string "decompositionType" shows as "Decomposition Type"
			}
		}
		module map{
			class map{
				diagram "graphFileName"   shows as "Map File Name"
				string "title" shows as   "Map Title"
				string "concern"
				association map1: concerns to "concern"."concern" "concern"
			}
			class respRef{
				string "enclosingComponent" shows as "Enclosing Component"
				string "referenceID" shows as "Definition ID"
				association respID1 : references to "responsibility" "Definition ID"

				association respID2 : boundto to "map"."compRef" "Enclosing Component"
				association respID3 : refines to "map"."map"
			}
```

```
        class noDescription compRef{
            string "referenceComponent" shows as "Definition ID"
            string "parentComponent" shows as "Parent Component"

            association compID1 : references to "component"   "Definition ID"
            association   compID2   :   boundto   to   "map"."compRef"   "Parent
Component"
            association compID3 : refinesto "map"."map"
        }
        class noDescription pluginBidning{
                int "replicationFactor"
                association plugin1: boundto to "map"."stub"
                association plugin2: plugin to "map". "map"
        }

        class noDescription stub{
            string "stubType" shows as "Stub Type"
            association stubID1 : refines to "map"."map"
        }
    }
    module ignoreInReport grlDiagram{
        fileName "GRL Diagrams"
        class grldiagram shows as "grl diagram"{
            diagram "graphFileName"     shows as "Diagram File Name"
            string "title" shows as     "Diagram Title"
            string "concern"
            association grl1: concerns to "concern"."concern" "concern"
        }
        class parentActor{
        }
        class noDescription actorRef{
            string "referenceActor" shows as "Definition ID"

            association actorRef1 : references to "actor"."actor" "Definition ID"
            association actorRef3 : refines to "grlDiagram"."grldiagram"
        }
        class intentionalElementRef{
            string "enclosingActor" shows as "Enclosing Actor"
            string "defID" shows as "Definition ID"
            association          ieAsso1          :          references          to
"intentionalElement"."intentionalElement" "Definition ID"
            association   ieAsso2   :   boundto   to   "grlDiagram"."actorRef"
"Enclosing Actor"
            association ieAsso3 : refines to "grlDiagram"."grldiagram"
        }
    }
    module component{
        class component{
            string "Type" shows as "Type"
        }
    }
    /*
     * Renaming the module intentionalElementAssociations to elementlink
     */
    module elementlink{
        fileName "Intentional Element Associations"
        class elementlink{
            string "Type"
            string "sourceID" shows as "Source ID"
            string "destinationID" shows as "Destination ID"
            association        elemlinkAsso1        :        refines        to
"intentionalElement"."intentionalElement" "source ID"
            association        elemlinkAsso2        :        refines        to
"intentionalElement"."intentionalElement" "destination ID"
        }
        class contribution{
             string "contribution type" shows as "Contribution Type"
            string    "contribution    quantitative"    shows    as    "Contribution
Quantitative"
            string "Correlation"
            string "sourceID" shows as "Source ID"
            string "destinationID" shows as "Destination ID"
            association        contributionAsso1        :        refines        to
"intentionalElement"."intentionalElement" "source ID"
            association        contributionAsso2        :        refines        to
"intentionalElement"."intentionalElement" "destination ID"
        }
    }
    module strategy{
        class strategy{
            string "Author"
        }
    }
    module scenario{
        class scenario{
        }
    }
    // Association type declarations
    associationType concerns "In Concerns"
    associationType plugin "Plugin"
    associationType requests "Requests"
    associationType references "References"
    associationType refines "Refines"
    associationType boundto "Bound To"
}
}
```

**Appendix B:**

This appendix describes the subset of the "HelloWorld" model.

```
model jUCMNav{

    folder jUCMNav{

        module responsibility{
            //default name, id, description
            class responsibility{
            }
        }

        module ignoreInReport actor{
            class actor{

            }
        }

        module ignoreInReport intentionalElement{
            class intentionalElement{

            }
        }
        module map{
            class map{
                diagram "graphFileName"     shows as "Map File Name"
                string "title" shows as     "Map Title"
            }

            class respRef{
                int "Fx"
                int "Fy"
                string "enclosingComponent" shows as "Enclosing Component"
                string "referenceID" shows as "Definition ID"
                association respID1 : references to "responsibility" "Definition ID"
                association respID2 : boundto to "map"."compRef" "Enclosing
Component"
                association respID3 : refines to "map"."map"
            }

            class noDescription compRef{
                int "Fx"
                int "Fy"
                int "Width"
                 int "Height"
                string "referenceComponent" shows as "Definition ID"
                string "parentComponent" shows as "Parent Component"


                association compID2 : boundto to "map"."compRef" "Parent
Component"
                association compID3 : refinesto "map"."map"
            }

            class noDescription stub{
                string "stubType" shows as "Stub Type"
                association stubID1 : refines to "map"."map"
            }

             class startPoint{
                   association spAsso1 : refines to "map"."map"
               }

               class endPoint{
                   association epAsso1 : refines to "map"."map"
               }
        }

        module component{
            class component{
                string "Type" shows as "Type"
            }
        }

        module contribution{
            class contribution{
                string "Type"
                 string "contribution type" shows as "Contribution Type"
                int "contribution quantitative" shows as "Contribution Quantitative"
                string "Correlation"
                string "sourceID" shows as "Source ID"
                string "destinationID" shows as "Destination ID"

                association contributionAsso1 : refines to
"intentionalElement"."intentionalElement" "source ID"
                association contributionAsso2 : refines to
"intentionalElement"."intentionalElement" "destination ID"
            }
        }

        module ignoreInReport grlDiagram{
            fileName "GRL Diagrams"

            class grldiagram shows as "grl diagram"{
                diagram "graphFileName"     shows as "Diagram File Name"
                string "title" shows as     "Diagram Title"
```

```
                    string "concernID" shows as "ConcernID"
                    string "concernName" shows as "ConcernName"

               }


               class noDescription actorRef{
                    int "Fx"
                    int "Fy"
                    int "Width"
                    int "Height"
                    string "referenceActor" shows as "Definition ID"

                    association actorRef1 : references to "actor" "Definition ID"
                    association actorRef3 : refines to "grlDiagram"."grldiagram"
               }

               class intentionalElementRef{
                    int "Fx"
                    int "Fy"
                    string "enclosingActor" shows as "Enclosing Actor"
                    string "defID" shows as "Definition ID"

                    association ieAsso1 : references to "intentionalElement" "Definition
ID"
                    association ieAsso2 : boundto to "grlDiagram"."actorRef"
"Enclosing Actor"
                    association ieAsso3 : refines to "grlDiagram"."grldiagram"
               }
          }

          associationType concerns "In Concerns"
          associationType requests "Requests"
          associationType references "References"
          associationType refines "Refines"
          associationType boundto "Bound To"

     }
}
```

## References:

[1]. Rahman, A. *A Domain-Specific Language for Traceability in Modeling*. M.Sc.A. thesis, School of Electrical Engineering and Computer Science, University of Ottawa, Canada, 2013.

[2]. IBM: *Rational DOORS*. http://www.ibm.com/software/awdtools/doors. Accessed December 2013.

[3] .Mussbacher, G. *Aspect-oriented User Requirements Notation*. PhD thesis, School of Information Technology and Engineering, University of Ottawa, Canada, 2010.

[4]. ITU-T – International Telecommunications Union: *Recommendation Z.151 (10/12) User Requirements Notation (URN) – Language definition*. Geneva, Switzerland, Oct. 2012.

[5]. *jUCMNav 5.4.0*, University of Ottawa, 2013.
http://jucmnav.softwareengineering.ca/jucmnav/ (last accessed December 2013)

[6]. *jUCMNav metamodel*: http://jucmnav.softwareengineering.ca/twiki/bin/view/ProjetSEG/URNMetaModel