Using SysML and MORE to Support Software Maintenance and Integration

[1] Chih-Hung Chang, [1] Chih-Wei Lu, [2] William C. Chu

[1*] *Dept. of Information Management, Hsiuping Institute of Technology, Taiwan*

[2] *Dept. of Computer Science and Information Engineering, Tunghai University, Taiwan*

[1]*{[*]chchang, cwlu}@mail.hit.edu.tw, [2]chu@csie.thu.edu.tw,*

Abstract

Maintaining consistency of software requirement documents with artifacts from other phases of software life cycle is a very important but difficult and time-consuming task. Most requirement documents are written in ambiguous natural language. They are usually less formal and imprecise. OMG released SysML as a standard of the modeling methodologies for software-intensive systems for applications of systems engineering. In this paper, we adopt SysML and MORE, a Model-based Object-oriented approach to Requirement Engineering (MORE) to support and improve the maintenance and consistency of software requirement documents, as well as the consistency with other artifacts through software life cycle previously, to specify software artifacts. Forward from here, the completeness, consistency, traceability and reusability of requirement and its integration can be cost-effectively improved.

Keywords: Software Maintenance, Object-oriented, traceability, SysML

1. MORE and SysML

In the earlier research, we had applied XML [1] to describe software paradigms as an XML-based unified model (XUM) [2]. XUM is a mechanism to integrate and unify sets of models (e.g., UMLs or design patterns) used in well accepted software paradigms (e.g., OOA, OOD, or OOP). And our previous work MORE [3], by applying XUM, effective specifies the feasibility of the modeling from informal text-based documents into OO models. MORE promises the software requirement development from document-centered to model-driven development. Figure 1 shows the objects' relationship among MORE with the requirement representations in text format.

SysML[4] is an OMG modeling specification. SysML supports the specification, analysis, design, and verification and validation of a broad range of complex systems which may include hardware, software, information, processes, personnel, and facilities. SysML is designed to provide simple but powerful constructs for modeling a wide range of systems engineering problems. It is particularly effective in specifying requirements, structure, behavior, and allocations and constraints on system properties to support engineering analysis, and soon evolve to be an important 1standard of the modeling methodologies.

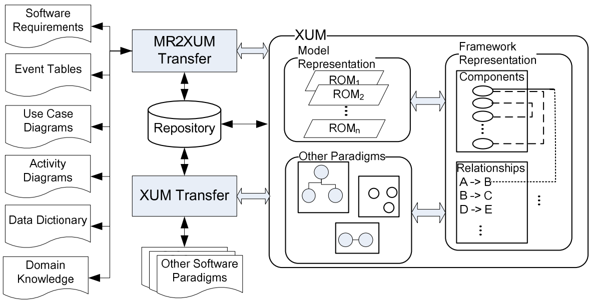

Figure 1: The Construction of XUM and MORE

Traditionally, requirement documents are still mostly written in ambiguous natural languages. In this paper, we will firstly adopt SysML to specify requirement documents. Then, we will associate and integrate SysML-based requirement and other software paradigms according by XUM. In next section, it discusses about the process concisely. After that, a conclusion is given.

2. SysML-based requirement modeling

At the beginning of system development, requirement engineers need to interviews with customers to collect relevant information. In order to collect more complete and precise customer requirements, we have defined a customer requirement elicitation template in this phase. The template includes: project goals, background, system scope, actors, functional requirement, and nonfunctional requirement.

The steps of requirement modeling processes are as the followings:

Step1: Collecting user requirement basing on the customer requirement elicitation template.

Step2: Objectizing and definition requirement object model from the user requirement.

Step3: Modeling use case diagrams and scenarios from the user requirement.

Step4: Refining system requirement from use case diagrams, scenarios, and reusable requirement repository.

Step5: Modeling SysML behavior diagrams and structure diagrams from scenarios.

Step6: Modeling SysML requirement diagrams, and generating system requirement.

Step7: Confirm the system requirement with customer. If the system requirement does not meet the user requirement, go back and repeat steps 1-6.

Step8: Transferring requirement artifacts (include user requirement, SysML paradigms …… etc) into XUM.

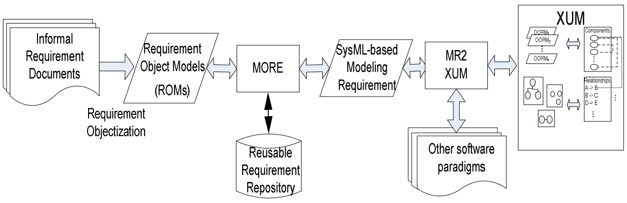
Informal Requirement Documents
Requirement Object Models (ROMs)
Requirement Objectization
MORE
Reusable Requirement Repository
SysML-based Modeling Requirement
MR2 XUM
Other software paradigms
XUM
Components
Relationships
A->B
B->C
D->E

Figure 2: The concept process of MORE

Figure 2 shows the concept model of our approach. With the support of ROMs (requirement object models), the process of the design phase is assisted with reusable requirement objects. So the design documents generated such as the structure diagrams, behavior diagrams, requirement diagrams …etc, will be based on the specification of ROMs and therefore associated with each other in the XML-based Unified Model.

Figure 3 shows an example of SysML requirement diagram, and figure 4 is the corresponding XUM specification.

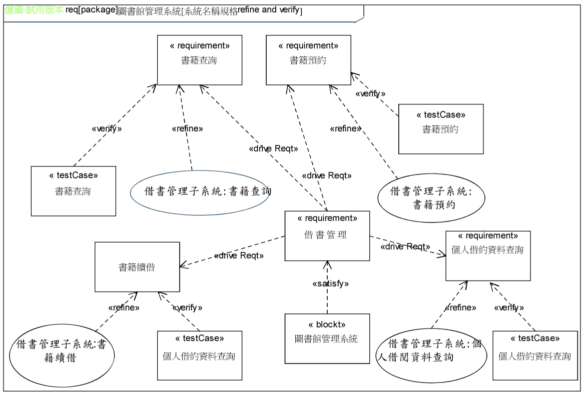

Figure 3: The example of SysML requirement diagram

```
  <UseCase_Daigram>
    <Actor name="student"/>
    <Actor name="system Manager"/>
    <Actor name="books Manager"/>
    <Usecase name="Loan Book">
        <Abstraction_link xlink:label="A_Loan_Book" xlink:title=
                "Use Case of Loan_Book" xlink:from="A_Loan_Book" xlink:to="? "/>
        <Abstraction_link xlink:from="A_Loan_Book" xlink:to=" ?"/>
                                                      … …
</Usecase>
<Usecase name="order Book">
                                                     … …
<Usecase name="manage Book">

<Usecase name="search Book">
                                                     … …
          <Relationship from="student" to="Loan Book" type="association"/>
        <Relationship from=" student " to="Return Book" type="association"/>
    <Relationship from="system Manager" to="Loan Book" type="association"/>
   </UseCase_Daigram>
</Requirement>
```

Figure 4: The corresponding XUM of figure 3.

In XUM, related elements may have abstraction relations, which are connected by abstraction_links. Similarly, integration_links connect views that share the same elements. Based on these links, if any information in each view or any source code gets changed, the affected views can be reflected by following the links.

3. Conclusion

Software requirement is the cardinal importance for developing and maintaining an information system because it is what the system is needed for. Driven by carefully designed SysML models and templates, the requirement documents can be formalized. Thereafter, through MORE and XUM, the related components can be associated and integrated with other software paradigms in other phases to improve the software development process.

In this paper, first we proposed a requirement modeling process with SysML and XUM to guide requirement modeling. Second we proposed a mechanism MORE to integrate system paradigms from requirements to implementation. If system developer or maintainer changes or modifies any of the software paradigms, he/she can easily find out the related elements and paradigms.

For the future work of this research, we will implement a toolset to support SysML and MORE which serves as a mechanism for the integration and maintenance through the whole process of software life cycle.

4. Acknowledgments

This research was supported in part by National Science Council, Taiwan R.O.C., under grants no. NSC97-2218-E-164-001, NSC97-2221-E-164-006, 95-2221-E-029-021-MY3, 96-2628-E-029-002-MY3, 97-2622-E-029-001-CC3

5. References

[1] World Wide Web Consortium, Extensible Markup Language (XML). http://www.w3.org/XML/ [September. 2006]

[2] C.W. Lu, W.C. Chu, C.H. Chang, W.D. Lian, and D.L. Yang, “Integrating Divers Paradigms in Evolution and Maintenance by an XML-based Unified Model,” Journal of Software Maintenance and Evolution, 15(3): 111 – 144, 2003.

[3] C.H. Wang, C.W. Lu, W.C. Chu, and C.H. Chang, “A Model-based Object-oriented Approach to Requirement Engineering(MORE),” the Proceedings of CompSAC 2007.

[4] Object Management Group, OMG Systems Modeling Language (OMG SysML™), V1.0, September. 2007.