Building and Reusing Of Requirements Repository

A thesis submitted as partial fulfillment of the requirements for the degree of Master of Computer Engineering

Submitted by
Ahmed Fakhry Ibrahim
Bachelor of Computer Engineering. (AASTMT)

Under supervision of:

Prof. Yousry El-Gamal
Vice President, Education & Research, Arab Academy for Science & Technology.

Prof. Medhat M. Fakhry
College of Engineering, Computer Engineering Dept., Arab Academy for Science & Technology.

December 2005
Disclaimer

Except where reference is made in the text of the thesis, this thesis contains no material published elsewhere or extracted in whole or in part from a thesis by which I have qualified for or been awarded another degree of diploma. No other person’s work has been used without due acknowledgement in the main text of the thesis.

This thesis has not been submitted for the award of any degree or diploma in any other tertiary institution.
Acknowledgements

I would like to express my gratitude to all those who helped me in my research and in writing of this thesis. In particular, I would like to thank my supervisor, the director of MIS Dept., Assoc. Prof. Medhat M. Fakhry for his aid in conceiving this thesis and for his accurate comments on the contents of the thesis. I thank Prof. Yousry El-Gamal for his encouragement throughout the MSc process.

But most of all, I am indebted to my family for their love, help and infinite patience while awaiting my deliverance from this work. I hereby dedicate this thesis to them.
ABSTRACT

The thesis suggests a new approach to requirements analysis intended to maximize reuse, and thus increase productivity and requirements specifications quality. Such benefits can be gained by reusing requirements from the existing software applications. Alternatively, reusable requirements representations could be identified and utilized in the process of refining software requirements for a single new application.

The thesis uses the application family concept for addressing the technological issues of concern to both requirements engineering and software reuse and uses it as the basis for developing a cohesive requirements reuse process. It shows that this concept could be instantiated into a practical object oriented method for requirements authoring and management (OOMRAM) and embedded in a web based software tool (Requirements Repository). The resulting method-tool pair has been developed to support analysts in developing a domain model, and then using the model in requirements analysis, reuse and refinement.

One of the main problems in requirements refinement is the effective matching of requirements which may be stored in a repository. A major contribution of this work is to deal with this difficulty and to develop an approach to facilitate requirement of classifying, comparing, searching and retrieving requirements stored in a repository.

The core of this work is the development of a requirement representation model to facilitate classification of requirements and their retrieval using an application family concept and a method of calculating similarity between family members. The proposed approach thus helps an analyst change requirements and so avoid the problems of unsatisfied, impractical and unnecessarily weak requirements.
# Table of Contents

1. **Introduction**  
   1.1. Introduction  
   1.2. Background and Motivations  
      1.2.1. Requirements Specification Document  
      1.2.2. Requirements Formality  
      1.2.3. Requirements, Reuse and Refinement  
   1.3. Objectives  
   1.4. Thesis Layout  

2. **Software Requirements Engineering Process**  
   2.1. Introduction  
   2.2. Software Requirements Engineering  
   2.3. Software Requirements Engineering Process  
      2.3.1. Requirements Elicitation  
      2.3.2. Requirements Analysis  
      2.3.3. Requirements specification  
      2.3.4. Requirements Validation and Verification  
      2.3.5. Requirements Modeling Frameworks  
   2.4. Requirements Management  
      2.4.1. Change Management  
      2.4.2. Traceability  
   2.5. Requirements Reuse  
   2.6. Summary  

3. **Application Family Concept As A Guide To Requirements Reuse**  
   3.1. Introduction  
   3.2. Previous Work in Requirements Reuse  
   3.3. Application Family Engineering  
   3.4. Requirements Application Family Model  
      3.4.1. Discriminants  
      3.4.2. Parameters  
   3.5. Single Application Definition  
      3.5.1. Free-Based Selection  
      3.5.2. Discriminant-Based Selection
3.5.3. Artificial Intelligence Based Selection

3.6. Summary

4. The Requirements Repository: Concept And Architecture
4.1. Introduction
4.2. Requirements Representation Model
   4.2.1. Requirements data representation and storage
   4.2.2. Requirements data editing
4.3. The System Architecture
4.4. The Repository Structure
4.5. Summary

5. Building And Maintaining the Requirements Repository: Design Framework
5.1. Introduction
5.2. Building and Maintaining Requirements
5.3. System Operations
   5.3.1. Specify Application Family
   5.3.2. Populate Application Family
   5.3.3. Administer Repository
      5.3.3.1. Modify application family schema
      5.3.3.2. Validate application family contents
      5.3.3.3. Manage users' repository accounts
      5.3.3.4. Control versioning
   5.3.4. Administer Applications Library
      5.3.4.1. Seed Applications Library
      5.3.4.2. Modify Application
      5.3.4.3. Drop Application
5.4. Building and Maintaining Elements' Design
   5.4.1. Business Tier (Application Server)
   5.4.2. Database Tier (Database Server)
   5.4.3. Presentation Tier (Thin Client)
5.5. Summary

6. Using the Requirements Repository: Design Framework
6.1. Introduction
6.2. Using Requirements Specifications
List of Figures

Figure 2-1: Relative Costs of Fixing Requirements Errors 7
Figure 2-2: Requirements Engineering Process Overview 8
Figure 2-3: Requirements Development Process Overview 9
Figure 2-4: Requirements Management Process Overview 9
Figure 2-5: Requirement Engineering Phases 9
Figure 2-6: Requirements Specification and Follow-on Products 11
Figure 3-1: Example application family scope and a family members within that scope that share a common core 20
Figure 3-2: Application family engineering lifecycle 21
Figure 3-3: Tree of parent-child relationships 22
Figure 3-4: Tree structure containing all discriminants forms 23
Figure 3-5: Example of parameter in application family model hierarchy 24
Figure 3-6: Example of application family model 27
Figure 3-7: Example single system specification 28
Figure 4-1: Our Proposed Requirement Object Definition 33
Figure 4-2: Requirement Object Example 35
Figure 4-3: The requirements repository architecture 37
Figure 4-4: Proposed System Use-Cases 38
Figure 4-5: Requirements Repository E-R diagram 39
Figure 5-1: Specify Application Family use- case diagram 42
Figure 5-2: Populate an application family use- case diagram 43
Figure 5-3: Administer repository system use-case diagram 44
Figure 5-4: Administer applications library use- case diagram 44
Figure 5-5a: Specify Application Family - Sequence Diagram 45
Figure 5-5b: Specify Application Family - Activity Diagram 46
Figure 5-6a: Populate Application Family - Sequence Diagram 47
Figure 5-6b: Populate Application Family - Activity Diagram 47
Figure 5-7: Modify Application Family Schema - Sequence Diagram 49
Figure 5-8: Validate Application Family Contents - Sequence Diagram 49
Figure 5-9: Create User Account - Sequence Diagram 50
Figure 5-10: Modify User Account - Sequence Diagram 51
Figure 5-11: Control Versioning – Sequence Diagram
Figure 5-12: Seed Applications Library – Sequence Diagram
Figure 5-13: Modify Application - Sequence Diagram
Figure 5-14: Drop Application - Sequence Diagram
Figure 5-15: Requirements Repository Entities
Figure 5-16: Requirements Repository Structure Diagram
Figure 6-1: Reasoning applications use-case diagram
Figure 6-2: Application Reasoning – Activity Diagram
Figure 6-3: Normal Mode - Sequence Diagram
Figure 6-4: Intelligent Mode - Sequence Diagram
Figure 6-5: Matching Algorithm
Figure 6-6: Navigation structure model for using the requirements repository
Figure 6-7: Electronic record keeping system layout
Figure 6-8: Ratio of cost reduction for each of the family members
List of Tables

Table 5-1: Account table 60
Table 5-2: Application family requirement parameters table 60
Table 5-3: Application family requirement properties table 60
Table 5-4: Application family requirements table 61
Table 5-5: Application families table 61
Table 5-6: Applications table 62
Table 5-7: Application requirements table 62
Table 5-8: Discriminants table 62
Table 5-9: Requirements' versions table 63
## Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMB</td>
<td>Engineering Management Board</td>
</tr>
<tr>
<td>FDIR</td>
<td>Failure Detection, Isolation, and Recovery</td>
</tr>
<tr>
<td>EPG</td>
<td>Engineering Process Group</td>
</tr>
<tr>
<td>ISO</td>
<td>International Organization for Standardization of Geneva, Switzerland</td>
</tr>
<tr>
<td>PAL</td>
<td>Process Asset Library</td>
</tr>
<tr>
<td>CDR</td>
<td>Critical Design Review</td>
</tr>
<tr>
<td>CIO</td>
<td>Chief Information Office</td>
</tr>
<tr>
<td>CMM©</td>
<td>Capability Maturity Model©</td>
</tr>
<tr>
<td>CMMI©</td>
<td>Capability Maturity Model© Integration</td>
</tr>
<tr>
<td>CMMI©-SE/SW</td>
<td>Capability Maturity Model© Integration for Systems Engineering and Software Engineering</td>
</tr>
<tr>
<td>CMU</td>
<td>Carnegie Mellon University</td>
</tr>
<tr>
<td>COTS</td>
<td>Commercial-Off-The-Shelf</td>
</tr>
<tr>
<td>CSCI</td>
<td>Computer Software Configuration Item</td>
</tr>
<tr>
<td>EIA</td>
<td>Electronic Industries Alliance; subsidiary of Government Electronics and Information Technology Association of Arlington, VA</td>
</tr>
<tr>
<td>FAR</td>
<td>Federal Acquisition Regulation</td>
</tr>
<tr>
<td>GOTS</td>
<td>Government-Off-The-Shelf</td>
</tr>
<tr>
<td>GPMC</td>
<td>Governing Program Management Council</td>
</tr>
<tr>
<td>HWCI</td>
<td>Hardware Configuration Item</td>
</tr>
<tr>
<td>I/O</td>
<td>Input/Output</td>
</tr>
<tr>
<td>IEEE</td>
<td>Institute of Electrical and Electronics Engineers, Standards Association of Piscataway, NJ</td>
</tr>
<tr>
<td>ITA</td>
<td>Independent Technical Authority</td>
</tr>
<tr>
<td>ITMRA</td>
<td>Information Technology Management Reform Act</td>
</tr>
<tr>
<td>IV&amp;V</td>
<td>Independent Verification and Validation</td>
</tr>
<tr>
<td>MOTS</td>
<td>Modified-Off-The-Shelf</td>
</tr>
<tr>
<td>NESC</td>
<td>NASA Engineering and Safety Center</td>
</tr>
<tr>
<td>NPD</td>
<td>NASA Policy Directive</td>
</tr>
<tr>
<td>NPR</td>
<td>NASA Procedural Requirements</td>
</tr>
<tr>
<td>PDR</td>
<td>Preliminary Design Review</td>
</tr>
<tr>
<td>SCAMPI</td>
<td>Standard CMMI© Appraisal Method for Process Improvement</td>
</tr>
<tr>
<td>SEI</td>
<td>Software Engineering Institute</td>
</tr>
<tr>
<td>SW</td>
<td>Software</td>
</tr>
<tr>
<td>SWE</td>
<td>Software Engineering</td>
</tr>
<tr>
<td>Acronyms</td>
<td>Explanation</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>CBR</td>
<td>Case Based Reasoning</td>
</tr>
<tr>
<td>DB</td>
<td>Database</td>
</tr>
<tr>
<td>DBMS</td>
<td>Database Management System</td>
</tr>
<tr>
<td>E-R</td>
<td>Entity-Relationship (Modeling Notation)</td>
</tr>
<tr>
<td>eRecordKeeping</td>
<td>Electronic Record Keeping Applications</td>
</tr>
<tr>
<td>ES</td>
<td>Expert System</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>KR</td>
<td>Knowledge Representation</td>
</tr>
<tr>
<td>MA</td>
<td>Multiple Adaptor Discriminant</td>
</tr>
<tr>
<td>MRAM</td>
<td>Method for Requirements Authoring and Management</td>
</tr>
<tr>
<td>OO</td>
<td>Object Oriented</td>
</tr>
<tr>
<td>OOMRAM</td>
<td>Object Oriented MRAM</td>
</tr>
<tr>
<td>OP</td>
<td>Optional Discriminant</td>
</tr>
<tr>
<td>RARE</td>
<td>Reuse-Assisted Requirements Engineering</td>
</tr>
<tr>
<td>RDBMS</td>
<td>Relational Database Management System</td>
</tr>
<tr>
<td>RE</td>
<td>Requirements Engineering</td>
</tr>
<tr>
<td>RRR</td>
<td>Requirements, Reuse and Refinement</td>
</tr>
<tr>
<td>RKR</td>
<td>Requirements Knowledge Repository</td>
</tr>
<tr>
<td>SA</td>
<td>Single Adaptor Discriminant</td>
</tr>
<tr>
<td>SA&amp;D</td>
<td>Systems Analysis and Design</td>
</tr>
<tr>
<td>SDLC</td>
<td>Software Development Life Cycle</td>
</tr>
<tr>
<td>SE</td>
<td>Software Engineering</td>
</tr>
<tr>
<td>UML</td>
<td>Unified Modeling Language</td>
</tr>
</tbody>
</table>