

R-TOOL: A SUPPORTING TOOL FOR A QUALITY ORIENTED REUSE STRATEGY

Maryoly Ortega, Anna Grimán, María Pérez and Luis E. Mendoza

Departamento de Procesos y Sistemas, Universidad Simón Bolívar, Valle de Sartenejas, Caracas, Venezuela

Keywords: Ontology, Reuse Strategy, Software Quality, V&V, Tool.

Abstract: The quality of reusable elements must be rigorously monitored and guaranteed before they can be reused, this is known as Certification. High levels of certification of these elements generate trust and stimulate reuse. In this paper we describe the development of a tool based on quality oriented reuse strategies. To this end, we take as starting point an ontology that rigorously correlates the essential concepts of systematic reuse to quality. This ontology reinforces the proposed strategy, which in turn is supported by the tool. The methodology used is based upon the Methodological Systemic Framework for Information Systems Research. For the development of the tool, we used the iterative incremental development process Rational Unified Process (RUP.) We took into account the inception and elaboration phases, and developed an iteration of the construction phase. As a result of the development process we built a tool which supports the main activities of the proposed strategy. These activities are Certify Domain Models, Requirement Specifications, Architectural Designs and Code, through checklists, allowing to store, classify, search and recovery the reusable elements and its properties.

1 INTRODUCTION

We can reuse not only software components but also artifacts that were developed along the development process (Reifer, 1997).

The certification of reusable elements allows developers to assure high levels of quality before the elements are reused. Reuse promotes improvements when developing software, since it increases productivity and quality within the development process (Sommerville, 2004).

The objective of this paper is to build a tool which supports a quality oriented reuse strategy.

The strategy aims to systematically introduce reuse since the beginning of the software development process, conceiving the reuse not only of Software Components but also of other artifacts such as Domain Models, Requirement Specifications, and Architectural Design.

Next, in section two we present a summary of the background upon which the research was developed. In section 3 we will describe the research methodology. Section 4 describes the strategy to support quality oriented reuse. Section 5 presents the development of the supporting tool. Finally, section 6 presents some concluding remarks and future work.

2 BACKGROUND

According to Frakes and Kang (2005), *software reuse* is the use of existing software or software knowledge to construct new software. *Systematic reuse* is introduced early in the life-cycle of software development. It is a formal and well-documented process, it is domain-specific and based on a re-creative process (Sodhi and Sodhi, 1999). *Reusable assets* can be either reusable software or software knowledge. According to (Reifer, 1997) there are six properties for software products to become reusable assets: Domain Specific, Generalized Form, Fixed Functionality, Known Performance, Well-bounded Interfaces, Demonstrable Quality.

The Asset Management is the process used to manage the reusable elements making them available to users in a satisfactory way. This process considers four activities: Library Management, Library Population, Library Operation, and Library Maintenance (Reifer 1997).

Library Population includes the certification process that guarantees an element possesses a particular property (Li et al., 2002). The V&V activities use techniques which are classified into: static (revisions, inspections, walkthroughs, audits), dynamic (white-box testing and black-box testing),

inquisitive (questionnaires, checklists and scenarios), and measuring activities (architecture description languages, and metrics).

Ghiotto et al. (2006) proposed ontology where these reuse concepts and quality concepts are linked. This ontology was the base for the proposed tool in this paper.

3 METHODOLOGICAL FRAMEWORK

This research used the Methodological Framework for Research of Information Systems (Pérez et al., 2004).

The adaptation of the Methodological Framework for this work consists of ten steps: 1) Documentary and bibliographical research; 2) Background Analysis; 3) Formulation of the Objectives and Scope of the Research; 4) Adaptation of the Methodological Framework; 5) Proposal of a strategy to support quality reuse; 6) Analysis of Context; 7) Application of the DESMET Methodology (Kitchenham, 1996), the method most appropriate to evaluate the strategy was the feature analysis-survey; 8) Evaluation of the proposed strategy; 9) Analysis of the results; 10) Conclusions and Recommendations.

4 PROPOSED STRATEGY

The proposed strategy to be supported by the tool was based in three paradigms (Ramachandran, 2005; Reifer, 1997): 1) it specifies the domain, 2) it is supported by the process and 3) it has organizational and management support. From a quality point of view the strategy will follow the GQM (Basili, 1992) paradigm. The objective of the measurement will be identified for each type of reusable element using their characteristics. Metrics will be used to determine the level of certification of the elements.

4.1 Objectives of the Strategy

The objectives of the reuse strategy are: 1) To introduce reuse within the software development process: 1.1) To implement reuse at an early stage of the life cycle; 1.2) To reuse other elements in addition to the code; 2) To reduce the timeframes associated with the process of software development; 3). To reach the third level of the of Software Reuse Maturity Model-RMM (Sodhi y Sodhi, 1999). The repository must be populated with certified and validated elements. A set of initial

metrics are implemented to reflect the benefits of reuse in terms of production of high quality and better profitability.; 4) To increase the levels of quality of each reusable element before its reuse; and 5) To promote reuse within the development group.

4.2 Components of the Strategy

We identified the following components in the strategy: a) Domain Analysis in the development process; b) Domain Analyst Role; c) Reuse within all the phases of the software development process; d) Use of standards and e) Use of a repository of reusable elements.

5 DEVELOPMENT OF R-TOOL

5.1 Inception Phase

The purpose of the Inception phase is to achieve an agreement with the stakeholders, regarding the objectives of the project (Rational, 2000; Kruchten, 2003).

5.1.1 Vision

This document provides a complete vision of the software system under development and supports the contract between the funding authority (Research Lab for Information Systems, LISI) and the development entity. The problem is to store large amounts of software elements and/or information manually which exhibit great potential for reuse. This problem affects analysts, architects and developers. One solution is to automate the storage, classification, search and recovery of the software elements.

5.1.2 Functional Requirements Specification

This document is used to express the behavior of R-tool. The functional requirement are: Log system users, Register users, Authenticate users, Certify reusable elements, Introduce reusable elements to the repository, Eliminate the reusable elements, Browse a catalogue of reusable elements, Locate reusable elements, Recover reusable elements, Modify checklists, Produce time statistics, Manage reuse statistics, Send an e-mail to the author of the element when it is reused, Manage Patterns, Specify Goal Question Metrics, Certify Use Case Diagrams and Activity Diagrams, Certify through checklists, Answer to checklists, Management of vocabulary.

5.1.3 Supplementary Requirements Specification

We take as a reference the quality model ISO/IEC 9126 to present a list of detailed complementary requirements for R-tool: 1) The system must be a web application; 2) The system must be sustainable and evolving; 3) The system must be user-friendly; 4) The system must be efficient; 5) The system must be tolerant to failures; 6) The system must reflect the reuse philosophy.

5.1.4 Initial Use Case Model

The use case model of the tool for support of the quality reuse strategies shows a total of 74 use cases. Figure 1 shows the global vision of the actors and sub-systems of the tool.

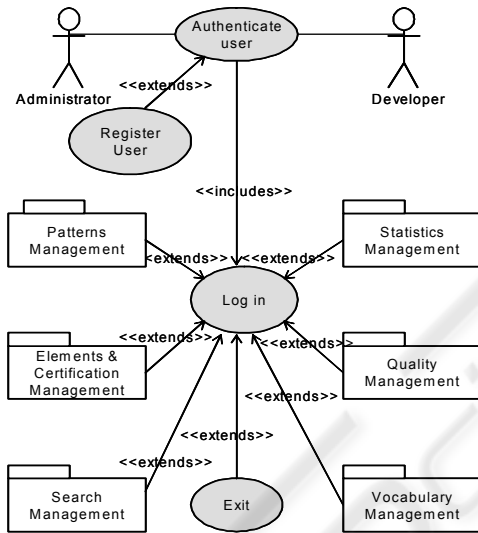


Figure 1: Global view of the actors and sub-systems of R-tool.

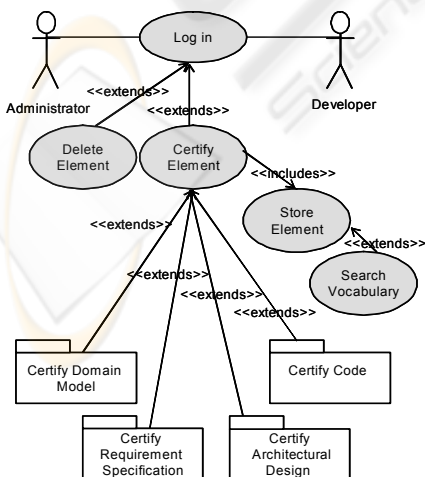


Figure 2: Elements and Certification Management Subsystem.

The use cases which are critical for the first iteration of the Construction phase of the (RUP) methodology are related to the Elements and Certification Subsystem (see Figure 2).

Figure 3 shows a diagram of the use case for the Requirement Specification Certification Subsystem.

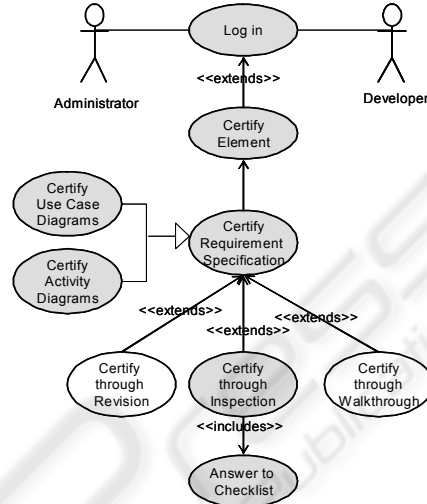
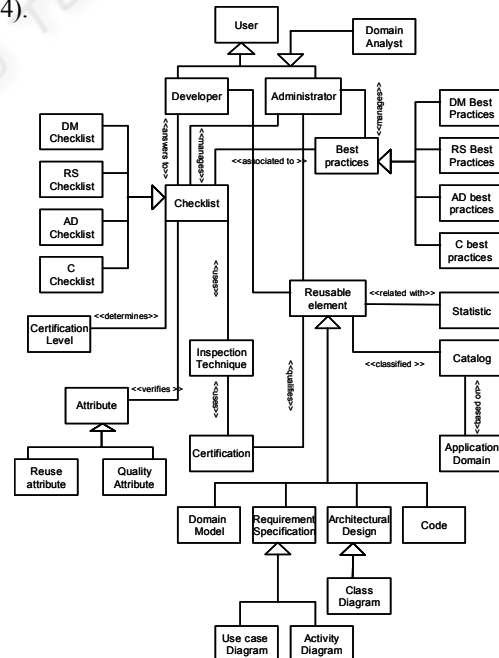


Figure 3: Requirement Specification Certification Subsystem.

5.2 Elaboration Phase

The purpose of the Elaboration phase is to determine a baseline for the system. We present the Conceptual Model of the Logical View for the tool (see Figure 4).



Legend: DM-Domain Model, RS-Requirement Specification, AD-Architectural Design, C-Code

Figure 4: Conceptual Model of the tool.

5.3 Construction Phase

The reach of the construction phase for this version of the tool is limited to only one iteration, in which we implement the use cases identified as critics: 1) Certifying four types of reusable elements: Domain Model, Requirement Specification, Architectural Design and Code Modules. There are three types of diagrams that can be certified: Use Case, Activity, and Class Diagrams; 2) To store reusable elements in a repository; 3) Eliminate reusable elements from the repository; 4) Search for reusable elements through four types of search: by name, by keyword, by project and by catalogue consult; Modify checklists used as certification mechanisms.

The first version of the tool presents a checklist to certify Requirements Specifications, that includes some questions, e.g. Has all business use cases been classified according to its relevance (primary, secondary, optional)?, Do the requirements define all the information that will be displayed to user?, etc.

The properties stored for each reusable element are: File name, Element name, File type, Author e-mail, Description, Time employed in elaborating the element, Language, Project where the element was developed, Keywords, Certification level, Performance (code), Use restrictions, Environmental restrictions, etc.

6 CONCLUSIONS

Checking for the desired quality characteristics for each element the tool allows us to guarantee that the reusable elements have an adequate level of quality. In addition, the tool promotes reuse and allows users to certify storage, search and recover reusable elements. The tool will keep track of their properties and history in any organization.

Future work considers the evolution of the tool as well as the development of methodologies and quality methods which favor quality reuse during the software's development process.

ACKNOWLEDGEMENTS

We acknowledge Pamela Ghiotto for her collaboration in this research. This research was supported by Universidad Simón Bolívar: DID Project # S1-IN-CAI-012-06 and by the Science and Technology Ministry of R. B. Venezuela: Fonacit Project # S1-2005000165.

REFERENCES

- Basili, V., 1992. *Software modeling and measurement: the goal/question/metric paradigm*. Technical Report CS-TR-2956. University of Maryland.
- Ghiotto, P., Ortega, M., Grimán, A., Mendoza, L. and Pérez, M., 2006. Ontology Proposal for Quality Oriented Reuse. In *Proceedings of IEEE IRI-2006*, Hawaii, USA, September:533-538.
- Frakes, W. B., Kang, K., 2005. Software Reuse Research: Status and Future, *IEEE Transactions on Software Engineering*, Vol.31, No. 7: 529-536.
- Kitchenham B., 1996. *DESMET: a method for evaluating Software Engineering methods and tools*. technical report TR96-09, University of Keele.
- Kruchten, P., 2003. *The Rational Unified Process: An Introduction*. Third Edition. Addison-Wesley Professional.
- Li P., Shaw M., Stolarick K., Wallnau K., 2002. The potential for synergy between certification and insurance, in *Proceedings of the First International Workshop on Software Reuse Economics, held in conjunction with the Seventh International Conference on Software Reuse*, Austin, Texas.
- Pérez, M., Griman, A., Mendoza, L. and Rojas T., 2004. A Systemic Methodological Framework for IS Research, In *Proceedings of AMCIS-10*, New York, New Cork.
- Ramachandran, M., 2005. Software Reuse Guidelines. *ACM SIGSOFT Software Engineering Notes*, USA, ACM. Vol. 30, No. 3: 1-8.
- Rational Unified Process with content for the Microsoft Web Solution Platform, 2000.
- Reifer, D. J., 1997. *Practical Software Reuse, Strategies for introducing reuse concepts in your organization*, John Wiley & Sons, Inc.
- Sodhi, J. and Sodhi P., 1999. *Software Reuse, Domain Analysis and Design Process*. Computing McGraw Hill.
- Sommerville, I., 2004. *Software Engineering*, 7^o edition. Pearson Education.