
Luciano Édipo Pereira da Silva\textsuperscript{1} *, Débora Maria Barroso Paiva\textsuperscript{1} Ellen Francine Barbosa\textsuperscript{2} Rosana Teresinha Vaccare Braga \textsuperscript{2} Maria Istela Cagnin \textsuperscript{1}

\textsuperscript{1}Facom – Federal University of Mato Grosso do Sul – Campo Grande (MS) - Brazil
\textsuperscript{2}ICMC – University of São Paulo – São Carlos (SP) - Brazil

E-mail: \textsuperscript{1}\{lucianoedipo, dmbpaiva, istela\}@gmail.com \textsuperscript{2}\{francine, rtvb\}@icmc.usp.br

Abstract

Reuse is an important mechanism to increase productivity and to reduce time and costs during software development. Although source code is the most commonly reusable asset, other types of assets can also be reused, such as requirements, business processes, analysis and design models, etc. In this context, it is important that the knowledge about reusable assets and its management are available to potential stakeholders. This work presents the development of an ontology of reusable assets specification and management, named ONTO-ResAsset. This ontology is evaluated under two points-of-view: domain experts and non-experts.

Keywords– Ontology, Software Assets, Software Asset Management, Software Reuse

1. Introduction

Software reuse, when conducted in a planned and systematic way, leads to better product quality, since the reusable assets have been previously used and tested. Additionally, it promotes agility to software projects execution and, consequently, cost reduction \cite{15}. The reuse of software assets is not limited only to the implementation phase. Rather, it can also be applied in previous phases related to requirements elicitation, analysis and design \cite{11}, as well as maintenance and supporting activities (e.g. project management documentation, verification, validation, and testing). Thus, it is important to do the proper management of assets reuse \cite{8}, which is generally aided by reuse repositories \cite{1}.

The knowledge about assets and their management is scattered and spread throughout several types of documents (e.g. papers, books, standards, patterns, guides, etc.), under different formats, and in several abstraction levels, besides being available in different ways. For example, the knowledge about reusable assets specification can be obtained specifically in books or technical reports \cite{12, 4}; the knowledge about reusable assets management and reuse repositories is spread in papers, books, and standards \cite{3, 8, 1}.

Under this perspective, it is relevant to notice that the knowledge about assets specification and assets reuse management should be unified, organized, and shared among those interested in it. This can be achieved through ontologies \cite{7}, which consist in an adequate technique to conceptualize knowledge.

The main goal of this paper is to present the development of a semi-formal ontology for the specification and management of reusable assets, named ONTO-ResAsset.

2. Related work

Works found on the literature about ontologies for the domain of interest of this paper define specific assets that can be reused during some activities of software development cycle, such as the SQA (Software Quality Assurance) ontology \cite{2}, the ontology of reusable test cases \cite{10} and the UI\textsuperscript{2}Ont ontology belongs to the user interfaces domain \cite{13}.

However, we did not find works that define ontologies to represent how a reusable asset can be specified, considering those that can be reused during the fundamental activities of software construction, and how a reusable asset can be managed. This knowledge is very important to all parts interested in software reuse and has motivated this work.

3. ONTO-ResAsset Development

The ONTO-ResAsset \cite{14} was built based on the \textit{Methontology} methodology \cite{6}, which defines a sequence
of steps that allow the gradual evolution of ontologies and the incorporation, at any moment, of new concepts as these steps are performed. To document the terms of the ONTO-ResAsset domain, it was used the glossary of terms and textual description in natural language. For the representation of the proposed ontology, a conceptual model was used, using UML (Unified Modeling Language) class diagrams.

Several information sources were used to identify the concepts of the ONTO-ResAsset [14, 3, 8, 12, 9]. The execution of the Methontology steps, used for the definition of the ONTO-ResAsset, is described below.

3.1. Step: Planning

The purpose of the ONTO-ResAsset ontology is to represent the domain of reusable assets in the context of software development, from the perspective of fundamental activities of the software construction (i.e, Requirements, Analysis, Design, Implementation, and Test), and considering the specification and the management of the reusable assets.

Regarding the level of formality, the ONTO-ResAsset is classified as a semi-formal ontology, according to Uschold [17], as it is expressed in a formally defined language - UML class diagram. Additionally, structured natural language is used in its representation.

To support the definition of the ONTO-ResAsset, the identification of domain terms was performed through four iterations of two steps of the Methontology (specification and knowledge capture), described in the next section. In the first iteration, the fundamental activities that are part of software development are identified and defined. In the second iteration the artifacts that can be reused during these activities were identified. In the third iteration the representation of artifacts was mapped as reusable assets. Finally, in the fourth iteration the required processes for managing reusable assets were identified.

3.2. Steps: Specification and Knowledge Capture

The knowledge capture step was performed transversely to the other steps of Methontology. Since the goal of this methodology is the creation of ontologies in a evolutionary manner, the knowledge capture step is carried out all the time.

In this work, the activities commonly performed for building the software were considered as fundamental activities of the software development cycle. These activities were identified from the analysis of construction activities of the software processes ontology [5] and of the development process of ISO/IEC 12207 [8]. The result of this step is a group of terms (G1) that describe fundamental activities, as for example Requirements, Analysis, Design, Implementation, and Test.

After defining the fundamental activities, we sought information about the artifacts commonly found in the literature and reused in at least one of these activities. The information sources to identify the concepts for that group (G2) were: the systematic review of Konda and Mandava [9], the domain analysis performed by Silva [14] from six reuse repositories, as well as the artifacts used during the software development according to Sommerville [16].

After the identification and selection of artifacts used by fundamental activities of software construction, it was noted in the literature the concepts that are related to reusable assets specification. Therefore, it was used as information source for that group (G3) the reusable assets specification of OMG [12]. In this paper, we considered the first two hierarchy levels of this specification for mapping the terms of the reusable assets specification, because they provide the necessary and appropriate knowledge about reusable assets within the ONTO-ResAsset scope, defined in the planning step.

The last group of terms (G4) of the ONTO-ResAsset domain refers to the management of reusable assets. For this, the following references have been used as information sources: Reuse Asset Management Process of ISO/IEC 12207 [8], and the set of functionalities of reuse repositories defined by Burégio [3], which contain functionalities commonly found in reuse repositories.

In the specification step, we defined 28 competence questions of the ONTO-ResAsset (for example: What activities are part of software construction? How is it possible to identify the problem that an asset can solve? What are the mechanisms that can be used to support the reuse asset management?), whose answers should be provided by the ontology. Moreover, in the specification step, a glossary of terms was created for each group of terms aforementioned, that due to space constraints are not presented here. Each glossary presents the name of each term, the information sources used and a textual description of each term. In the glossary of terms of group G1 we defined 6 terms (RequirementActivity, AnalysisActivity, DesignActivity, ImplementationActivity, etc). In the glossary of terms of group G2 we identified 11 terms (RequirementDocument, Model, SourceCode, Component, etc). In this group, we also identified the artifacts that are created or used in each activity of the previous group. In the glossary of terms of group G3 we defined 19 terms (Artifact, Asset, Classification, Context, etc) and in the glossary of terms of group G4 we found 18 terms (ManagementPlan, Asset, Classification, Context, etc).

3.3. Step: Conceptualization and Implementation

In the conceptualization step, the concepts corresponding to the terms obtained in the previous section and the re-
relationships and rules of association existing between them were identified.

It is worth noting that 24 new concepts were abstracted in this step. They resulted from the detailing of some relationships, but were not present in any information source used. In group G1 just a new term was defined (Person, which may take one or more roles during the fundamental activities of software construction). In group G3 we defined 16 new terms (AssetContext, SolutionArtifact, Problem, RequirementSolution, AnalysisSolution, etc). Finally, in group G4 we identified 7 new terms (Version, Feedback, Mechanism, NotificationMechanism, etc).

The concepts abstracted were needed to guide the conceptualization, aiming to answer the competence questions of the ONTO-ResAsset established in the specification step.

The relationships between the concepts were extracted from textual descriptions of domain terms, documented in the glossaries of terms (defined in the specification and knowledge capture steps). For each relationship, it has been defined a name, related to an action produced by the relation, for example, \textit{sendsWarningAbout} (<NotificationMechanism> \textit{sendsWarningAbout} <Asset>), and the multiplicity of the relationship between the concepts (for example, a <NotificationMechanism> \textit{sendsWarningAbout} one or more <Asset>).

In the Conceptualization step, the meaning of each relationship was defined in a textual format. Besides, in this step a table of relationships was also created, and it presents all concepts involved in each relationship, the multiplicity and the relationship name. In some cases, rules for the existence of certain relationships were also defined and presented. Furthermore, each element of the conceptual model was described in natural language, helping to get a semi-formal definition of the domain addressed by the ontology.

In the implementation step, all the concepts, relationships and multiplicities were represented in a conceptual model, using the UML class diagram, where each class is a domain concept and the associations (including aggregation and composition) and inheritance relationships are the relationships established between the concepts.

In Figure 1 the high-level conceptual model of the ONTO-ResAsset is shown. Each group of concepts is represented by a package. Due to space restrictions, the conceptual model of each group could not be shown. The complete conceptual model of the ONTO-ResAsset can be obtained elsewhere [14].

4. ONTO-ResAsset Evaluation

ONTO-ResAsset was evaluated by experts and non-experts in software reuse. The evaluation aimed at qualitatively analysing the ability of ONTO-ResAsset to provide knowledge about the domain of reusable assets specification and management, in an attempt to support the dissemination of this domain.

The evaluation of ONTO-ResAsset was performed in two steps, considering different perspectives. In Step 1, the ontology was evaluated by experts in the field of software reuse. In Step 2, the evaluation was conducted by two groups of evaluators, representing non-experts in the field, particularly interested in the reuse of software assets.

In Step 1 the ONTO-ResAsset was sent to a group of Software Engineering researchers with interest in software reuse. The goal was to get an impression from experts regarding: (i) the consistency and the proper coverage of the ontology according to its scope; and (ii) the possible errors/inconsistencies with respect to the definition and representation of the concepts and relationships among them.

As a result of Step 1, ten suggestions for improvements in ONTO-ResAsset were identified. Of these, three refer to concepts (improvements in the textual description, specialization of concepts, removal of a concept, and so on) and seven are related to relationships between concepts (removal of redundant relationships, changing from aggregation and composition relationships to association relationships, and so on). All suggestions for improvements were addressed and implemented in ONTO-ResAsset.

Next, in Step 2, the ONTO-ResAsset was sent to a group of domain non-experts. The goal was to evaluate if the ontology was able to transfer and disseminate knowledge regarding the specification and management of reusable assets. The evaluation questionnaire of Step 2 was completely based on the competence questions established during the specification of the ONTO-ResAsset

Step 2 was performed by two groups of evaluators. Group 1 was composed of eleven non-experts, who are part of the development team for a software company. Group 2 was composed of nine students of a software development course at Facom (Graduate Program), without any expertise knowledge.
in software reuse.

Each questionnaire answered by the evaluators in Groups 1 and 2 was analysed in order to verify the correctness of the answers and therefore assess the understanding of the domain expressed in the ontology by the evaluators. Each answer was scored as correct, partially correct, incorrect or unanswered. The answer was classified as partially correct when it had only some parts of the expected concepts or when it did not have all the correct items as expected.

The overall percentage of correct answers of the questions answered by Groups 1 and 2 was 68% and 73%, respectively; the percentage of partial success was 16% and 10%, respectively. Also, we noticed that 14% of the answers of Group 1 and 13% of Group 2 were completely wrong, and 2% and 4% of the questions were not answered by groups 1 and 2, respectively. From the overall analysis of the answers, it was possible to infer that the evaluators were able to understand the domain since they answered the competence questions that guided the construction of ONTO-ResAsset with a satisfactory success rate (on average 70.5%). It was also observed that the questions with lower scores in both groups refer to the concepts of reuse asset management (55% of correct answers by Group 1 and 61% by Group 2). This may indicate that the concepts of reuse asset management and their relationships require a more detailed textual description.

5. Conclusion and Future Work

The main contribution of this work refers to the establishment of ONTO-ResAsset – an ontology that represents and documents, in a single place and format, knowledge about the specification and management of assets that can be reused during the fundamental activities of software construction. It is important to highlight that the proposed ontology contributes to the research in the area of software reuse, gathering the existing knowledge available in several relevant sources of information in the literature.

From the evaluation performed, we also noticed that ONTO-ResAsset is able to adequately represent the knowledge about specification and management of assets, being possible to integrate the proposed ontology with other ontologies in the area of software reuse.

As future works, we highlight the need of: (i) implementing ONTO-ResAsset with the support of an ontology editor and knowledge acquisition system (such as Protégé); (ii) evolving ONTO-ResAsset to represent the taxonomic groups in more details; and (iii) evaluating ONTO-ResAsset with a greater number of non-experts in order to obtain more concrete and significant results.

References