A Unified Ontology-Based Process Model for Software Maintenance and Comprehension

Juergen Rilling[1], Yonggang Zhang[1], Wen Jun Meng[1], René Witte[1], Volker Haarslev[1], and Philippe Charland[2]

[1] Department of Computer Science
and Software Engineering
Concordia University, Montreal, Canada
{rilling,yongg_zh,w_meng,rwitte,haarslev}@cse.concordia.ca
[2] System of Systems Section
Defence R&D Canada Valcartier,
Val-Bélair, Canada
philippe.charland@drdc-rddc.gc.ca

Abstract. In this paper, we present a formal process model to support the comprehension and maintenance of software systems. The model provides a formal ontological representation that supports the use of reasoning services across different knowledge resources. In the presented approach, we employ our Description Logic knowledge base to support the maintenance process management, as well as detailed analyses among resources, e.g., the traceability between various software artifacts. The resulting unified process model provides users with active guidance in selecting and utilizing these resources that are context-sensitive to a particular comprehension task. We illustrate both, the technical foundation based on our existing SOUND environment, as well as the general objectives and goals of our process model.

Keywords: Software maintenance, process modeling, ontological reasoning, software comprehension, traceability, text mining.

1 Introduction and Motivation

Software maintenance is a multi-dimensional problem space that creates an ongoing challenge for both the research community and tool developers. These maintenance challenges are caused in particular by the variations and interrelationships among software artifacts, knowledge resources, and maintenance tasks [3,20,22]. Existing solutions [10,20] that address aspects of these challenges are commonly not integrated with each other, due to a lack of integration standards or difficulties to share services and/or knowledge among them. The situation is further complicated by the non-existence of formal process models to create a representation that describes the interactions and relationships among these artifacts and resources.

There has been little work in examining how these resources work together for end users [13,20] and how they can collaboratively support a specific program maintenance task. Maintainers are often left with no guidance on how to complete a particular task

T. Kühne (Ed.): MoDELS 2006 Workshops, LNCS 4364, pp. 56 – 65, 2007.

within a given context. Our research addresses this lack of context sensitivity by introducing a formal process model that stresses an active approach to guide software maintainers during maintenance tasks. The process model, its basic elements and their major inter-relations are all formally modeled by an ontology based on Description Logic (DL) [2]. The process behavior is modeled by an interactive process metaphor. Our approach differs from existing work on comprehension models [3], tool integration [11, 20] and task-specific process models [8,19,20] in several aspects:

1. A formal software maintenance process model based on an ontological representation to integrate different knowledge resources and artifacts.
2. An open environment to support the introduction of new concepts and their relationships, as well as enriching the existing ontology with newly gained knowledge or resources.
3. The ability to reason about information in the ontological representation to allow for an active and context-sensitive guidance during the maintenance process.
4. Analysis of relationships among resources, e.g., the traceability between artifacts.

The process model itself is motivated by approaches used in other application domains, like Internet search engines (e.g., Google[1]) or online shopping sites (e.g., Amazon[2]). Common to these applications is that they utilize different information resources to provide an active, typically context-sensitive user feedback that identifies resources and information relevant to a user's specific needs. The challenge in applying similar models in software maintenance goes beyond the synthesis of information and knowledge resources. There is a need to provide a formal meta-model to enable reasoning about the potential steps and resources involved in a maintenance process.

For example, a maintainer, while performing a comprehension task, often utilizes and interacts with various tools (parsers, debuggers, source code analyzers, etc.). These tool interactions are a result of both, the interrelationships among artifacts required/delivered by these tools and the specific techniques needed to complete a particular task. Identifying these often transitive relationships among information resources becomes a major challenge. Within our approach, we support automated reasoning across these different information resources (e.g., domain knowledge, documents, user expertise, software, etc.) to resolve transitive relationships. Furthermore, our model can be applied to analyze and re-establish traceability links among the various resources in the knowledge base [1].

From a more pragmatic viewpoint, process models have to be able to adapt to ever changing environments and information resources to be used as part of the process itself. In our approach, we address this problem by providing a uniform ontological representation that can be both extended and enriched to represent any newly gained knowledge or change in the information resource(s). This knowledge will also become an integrated part of the process that can be further utilized and reasoned on.

The remainder of the article is organized as follows. The relevant research background is introduced in Section 2. Section 3 describes in detail the context-driven program comprehension process model, followed in Section 4 by its implementation and validation. Discussions and future work are presented in Section 5.

[1] www.google.com
[2] www.amazon.com