Augmented Software Cognition

Paul Kogut, Carlos Amaro, Ross Klingensmith
Lockheed Martin Integrated Systems & Solutions, King of Prussia, PA
{paul.a.kogut,carlos.c.amaro,ross.f.klingensmith}@lmco.com

Abstract. This short position paper describes software engineering problems encountered on large scale systems and provides a vision for future semantic-based solutions that are being investigated in the Lockheed Martin Software Technology Initiative.

1 The Software Cognition Problem

In October 2005 Lockheed Martin Integrated Systems and Solutions (IS&S) earned the Software Engineering Institute (SEI) Capability Maturity Model Integration® (CMMI) Maturity Level 5. CMMI Level 5 defines industry best practices for integrated systems and software engineering. Based on the CMMI philosophy of continuous process improvement, LM IS&S recently formed a Software Technology Initiative (STI) that includes academic partners. STI will explore a variety of research directions that will meet the productivity and quality challenges of large scale software engineering in the future. One STI research area is software cognition.

Software engineers suffer from information overload just like many other knowledge workers (e.g., intelligence analysts, medical researchers…). Large software projects have information scattered throughout requirements and design documents, UML models, product manuals and code. It is difficult to understand the system as a whole. Software engineers spend too much time finding the right information. This complex interdependent information contains domain/application specific jargon which a software engineer often has to learn for each project they get assigned to. Misinterpretations of this jargon due to semantic ambiguity and inconsistencies lead to design errors and code defects. The application specific jargon found in software artifacts also inhibits reuse and incurs a large cost for transferring knowledge between:

- systems engineers and software engineers
- current project team members and new team members
- developers, operations personnel and software maintainers

Only a limited amount of knowledge about a system is represented in modeling languages such as UML and SysML[1]. Most knowledge remains in free text (requirements statements, design documents, comments in code and UML models, email…). As a result of this large amount of text, engineers spend a lot of time looking for what they need to know. Also, consistency and completeness checking (e.g., design reviews and peer reviews) is subjective, time consuming and subject to the cognitive limitations of humans.

2 Solving the Software Cognition Problem

We believe that a combination of Semantic Web ontologies, automated reasoning and human language technology is part of the answer to the software cognition problem described in the previous section. The goal is to improve human understanding of complex systems by delegating some of the thinking to the machine. In the first phase we plan to focus on reducing the effort to find relevant info in software artifacts. In the second phase we plan to focus on semantics-based development where ontologies are used early in the life-cycle to prevent errors and reduce knowledge transfer inefficiencies.

The approach in phase 1 is to investigate the application of Question Answering (QA) technology [2] to pre-existing software artifacts. Traditional information retrieval approaches (e.g., Google) return lists of possibly relevant documents that the user still has to read through. QA systems provide short direct answers to questions ranging from simple factoids (e.g., *What is the performance budget for target query processing?*) to more essay-like questions (e.g., *What functionality does the system provide for sensor fusion?*). QA systems apply various combinations of natural language processing, information retrieval and automated reasoning techniques. Semantic Web based QA systems such as Aqualog [3] have the added advantage of querying across unstructured text and structured sources (e.g., UML models).

Figure 1 shows a conceptual architecture for an Augmented Software Cognition (ASC) system. The basic idea is to input various forms of textual software artifacts and generate ontologies and knowledge bases that can be used to answer questions from the software engineer and eventually reason about consistency [4]. The ASC system would provide a layer on top of existing Model Driven Architecture (MDA) tools that support UML and SysML.

The ASC project is in its early stages. Initial efforts are focused on understanding the kinds of questions that would benefit software engineers. The test data consists of requirements and design documents from an existing product line architecture in the aerospace domain. The approach is to implement a simple prototype with a mature QA system such as Power Answer [5]. We also plan to investigate dialog–based QA technology such as HITIQA [6].

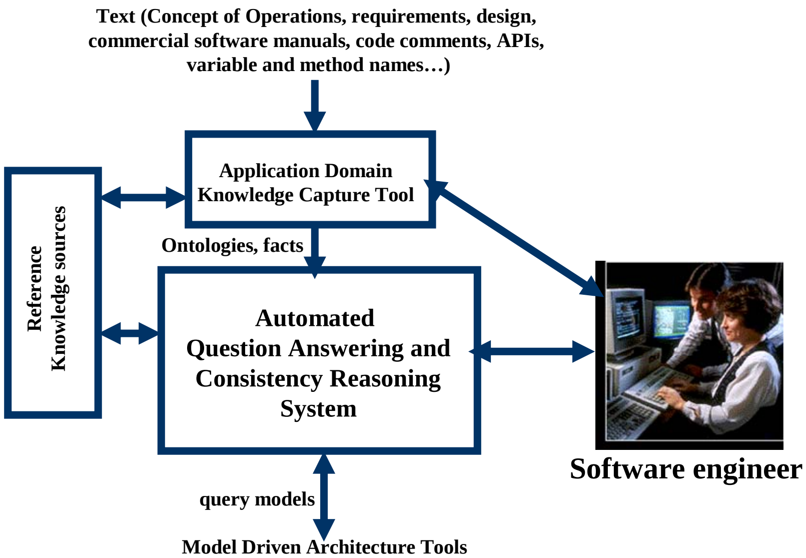

Figure 1: Augmented Software Cognition Architecture

The approach in phase 2 is to investigate semantic-based development. Ontology development would begin in the earliest phases of system definition. Proactive consistency checking tools will prevent engineers from making semantic mistakes from requirements all the way to code and test cases. The tools would force you to use terminology from the existing project ontologies or extend the ontologies as the system is elaborated. These tools would be designed to be interactive and user friendly by insulating the engineer from OWL details. The goal is to achieve semantic traceability from requirements and design assumptions down to method and parameter names. A semantic infrastructure across all software artifacts would support advanced forms of question answering and quality assurance and shift some of the cognitive burden from the engineer to the machine.

Acknowledgments. The material in this paper is based on work supported by the Lockheed Martin Software Technology Initiative.

References

1. SysML http://www.omgsysml.org
2. Strzalkowski, T., Harabagiu, S., editors, Advances in Open Domain Question Answering, Springer, 2006
3. Aqualog http://kmi.open.ac.uk/technologies/aqualog/
4. Baclawski, K., Kokar, M., Waldinger, R., Kogut P., "Consistency Checking of Semantic Web Ontologies," *1st International Semantic Web Conference (ISWC 2002)*, Sardinia, Italy, June 2002.
5. LCC Power Answer http://www.languagecomputer.com/solutions/index.html
6. Strzalkowski, T., Small, S., Hardy, H., Yamron, B., Liu, T., Kantor, P., Ng, K., Wacholder, N., "HITIQA: A Question Answering Analytical Tool" 1st Int. Intelligence Analysis Conference, May 2005