Chapter 3.9
Design Diagrams as Ontological Sources:
Ontology Extraction and Utilization for Software Asset Reuse

Kalapriya Kannan
IBM India Research Labs, India

Biplav Srivastava
IBM India Research Labs, India

ABSTRACT

Ontology is a basic building block for the semantic web. An active line of research in semantic web is focused on how to build and evolve ontologies using the information from different ontological sources inherent in the domain. A large part of the IT industry uses software engineering methodologies to build software solutions that solve real-world problems. For them, instead of creating solutions from scratch, reusing previously built software as much as possible is a business-imperative today. As part of their projects, they use design diagrams to capture various facets of the software development process. We discuss how semantic web technologies can help solution-building organizations achieve software reuse by first learning ontologies from design diagrams of existing solutions and then using them to create design diagrams for new solutions. Our technique, called OntExtract, extracts domain ontology information (entities and their relationship(s)) from class diagrams and further refines the extracted information using diagrams that express dynamic interactions among entities such as sequence diagram. A proof of concept implementations is also developed as a Plug-in over a commercial development environment IBM's Rational Software Architect.

INTRODUCTION

A Scientific American article describes evolution of Web that consisted largely of documents for humans to read and that included data and information for computers to manipulate. In order, to help the people and machines to communicate concisely, this huge repository of information (Web) have to be re-engineered to define shared and common domain theories that are machine processable. This is the precise aim of Semantic Web to build metadata – rich Web where presently human-readable content will have machine-understandable semantics. Semantic Web achieves the increased demand of shared semantic and a web of data and information derived from it through the adaptation of common conceptualizations referred to as Ontologies. Ontologies are fundamental building blocks of Semantic Web and therefore cheap and fast construction of domain-specific ontologies is crucial for the success and the proliferation of the Semantic Web. We explore how a semantic web may impact software engineering where common conceptualizations (e.g., software, work products, experience) from previous projects need to be reused in new projects.

The chapter will contribute in the following ways: (a) It will introduce an ontology learning technique from design diagrams (specifically UML) which is in line with the research direction of building ontologies from different sources to enable a rich semantic web; (b) It will show a concrete semantic web application how the semantic web technology of ontology can promote software reuse; (c) It will put recent research on transformations from UML to OWL and vice-versa in perspective; and (d) it will motivate more research effort in semi-automatically building integrated information models.

Ontology Learning and Sources

Ontology learning facilitates the construction of ontologies by ontology engineering. Ontology learning includes a number of complementary disciplines that feed on different types of unstructured, semi-structured and fully structured data in order to support a (semi -) automatic, cooperative ontology engineering process. Ontology learning is quiet a tedious task and often requires manual intervention. Ontology learning algorithms have concentrated in finding efficient methods of automatically extracting ontology information. Today, research apart from automatic ontology learning has also focused on finding sources (RDF, html, unstructured text etc) of ontological information. In our work we concentrate on identifying one such source i.e., design diagrams and provide an (semi-) automatic technique to extract (domain) ontological information. In the process of software solution development, large amount of ontological information is implicitly modeled in design diagrams. Our work aims at extracting this implicit ontology from the design diagrams. It largely helps reuse of the concepts in other software solutions.

In Figure 1, a simplified project cycle is shown. The project requirements are collected; the solution is developed and tested, and finally released to the customer. At the end of the project, a report is generated to capture the learning. However, it is rare that project-end deliverables or reports are explicitly used to improve the solution development process for new projects.

Our key observation is that the similarity among the projects alludes to the existence of a veritable domain of discourse whose ontology, if created, would make the similarity across the projects explicit. However, manually creating such domain ontologies is infeasible due to the training needed for the software professionals and the efforts required. Although many ontology engineering tools have matured over the last decade, manual ontology acquisition remains a tedious, cumbersome task that can easily result in domain knowledge acquisition bottleneck. We note that design diagrams are integral part of software project deliverables as they document crucial facets of the solution dealing with software artifacts, their relationships and runtime behavior. Much, but not all, information in theses diagrams are ontological information. If one were to extract and explicitly represent them, they would enable