Design Knowledge Creation and Analysis

Paul S Grisham and Dewayne E. Perry

Empirical Software Engineering Laboratory
Electrical and Computer Engineering
The University of Texas at Austin

grisham@mail.utexas.edu, perry@ece.utexas.edu

We propose to use content analysis of the professional software designer study to supplement our project on the use of design knowledge in the maintenance and evolution of software architectures for software-based systems. The goal of our work is to develop a better understanding of the documentation needs of system maintainers and to build a model of how they interact with documented knowledge about legacy designs. Software architecture provides an early view into the structure of the system, offers management support for implementation planning and cost estimation, and creates an intermediate model of the relationship between entities in the problem domain and entities in the solution domain. The context of design decisions and the intentions of designers – especially at the architectural level of abstraction – may be critical to system maintenance and evolution. Although languages and tool support for capturing and visualizing design knowledge have been proposed before, research has been ineffective in relating the capabilities of those systems to the informational needs of system maintainers, that is, determining what design knowledge from the initial design activities answer which questions asked by system maintainers of the legacy design.

We have developed a preliminary model of the design activities of system architects, system maintainers, and the differences between them based on prior cognitive task analysis in the research literature, interviews with senior designers, and our own prototype study. Our first use of the professional designers data is to perform a partial validation of this model. Although we will only be able to analyze the data with respect to initial design, many aspects of our model are derived from other engineering design domains, and observing the behaviors in software designers will strengthen our model's validity in the software design domain.

Our second use of the data is to determine the sources and types of knowledge generated during initial design activities. Whenever an artifact is produced as the result of a design activity, some knowledge is created to some purpose. The content and structure are both fitted to a specific purpose and a specific consumer of that knowledge, whether that purpose is made explicit or left implicit. Typically, these purposes are intended for use in the present or near-future context of the project—i.e., upstream activities—and little concern is given to understanding the later concerns and activities—i.e., downstream—such as maintenance, evolution, and reuse.

We will be using content analysis techniques to identify the creation of new knowledge as a consequence of design decisions and selection by the initial designers, or the transformation of tacit knowledge into explicit knowledge through arbitration and reasoning. We also seek to identify the types and sources of that knowledge, such as decisions based on prescriptive or mitigating knowledge, business context, or managerial dictate. If possible, the confidence and authoritativeness of this new knowledge will be evaluated. As we come to understand more about the intellectual work of software maintainers and their knowledge needs, we may discover that the information they require is not merely undocumented by the initial designers, but may also be tacitly unknown by them as well. In this case, the problem of knowledge transfer between initial designers cannot be satisfied through purely technological means.