CS Tech Rep # 2002-18
to 3/15/02

An Experiment in Applying Linguistic Insight to Improve Requirements

Kimberly S. Hanks
Department of Computer Science
University of Virginia
151 Engineer's Way
Charlottesville, VA 22904-4740
+1 44 434 982 2993
ksh4q@cs.virginia.edu

John C. Knight
Department of Computer Science
University of Virginia
151 Engineer's Way
Charlottesville, VA 22904-4740
+1 44 434 982 2216
knight@cs.virginia.edu

ABSTRACT

We discuss the role of domain knowledge in requirements validity, and review cognitive linguistic theory as applied to domain knowledge communication. We then review the motivation for, and structure of, an artifact, the domain map, designed to reduce the potential for breakdown in domain knowledge propagation by systematically compensating for normal variability in the ways that humans individually apply communicative heuristics. We further describe the design and execution of an experiment constructed to investigate the value of domain maps for improving requirements by increasing the potential for domain knowledge comprehension. Results of the experiment indicate that domain maps can be an effective tool in the propagation of domain knowledge with more fidelity.

1. INTRODUCTION

Errors introduced into software during the early stages of the lifecycle pose a doubly significant challenge to the development of high quality systems. First, errors introduced early are correspondingly more difficult and expensive to correct [20, 5]. In the worst case, they are left uncorrected and for some systems can result in significant loss. Second, there are more *of* them [21, 5]: the requirements stage in particular is implicated as the locus of introduction of more defects than any other. Lutz found that the majority of safety-critical defects in the systems she studied derived from poor requirements [18], and the Air Force's Rome Laboratory found that the majority of *all* defects they observed derived from poor requirements [26]. Brooks has stated that "[t]he hardest single part of building a software system is deciding precisely what to build" [2].

To a first approximation, then, the greatest improvement to software quality and to the efficiency of the development process is to be gained if a substantial advance in increasing requirements validity can be achieved. The field has advanced techniques for verification; under appropriate circumstances, we are quite facile at proving that a program meets its specification. However, the real problem appears to remain in producing a valid specification to begin with.

Domain knowledge plays a crucial role in this problem, since a goal of requirements is domain knowledge transfer. Curtis, Krasner, and Iscoe implicated the "...thin spread of application domain knowledge" as a main limiting factor to software productivity and quality in the design of large systems [8]. This makes requirements a communication problem, as echoed by Hayhurst and Holloway [11], with domain knowledge the thing being communicated and domain experts and developers doing the communicating. Without accurate conceptions of the real-world semantics relevant to a system, developers are forced to rely on misunderstandings and invalid assumptions about the entities they model, often without realizing it. Propagation of these misunderstandings and invalid assumptions can lead to annoying, expensive, or even catastrophic failures. Curtis, Krasner, and Iscoe report that "[c]ustomer representatives and system engineers complained that implementations had to be changed because development teams had misconceptions of the application domain" [8], and Lutz states "[i]t is not the internal complexity of a module but the complexity of the module's connection to its environment that yields the persistent, safety-related errors seen" [18]. To understand this connection, developers must acquire an understanding of relevant domain knowledge.

We previously examined how this breakdown in communication of domain knowledge might occur, and proposed a theory to explain its mechanism based on results from cognitive linguistics [9, 10]. Using insights gained in the process, we suggested the shape of an approach to coping with the challenges inherent in communicating domain knowledge; this approach embodies strategies specifically