

A Cooperative System for Knowledge Acquisition from Multiple Experts

Marita Duecker & Dirk Zimmermann; C-LAB
Stefanie Thies, University of Paderborn
Fuerstenallee 11, 33102 Paderborn, Germany
{mary, dirkz}@c-lab.de, thies@hni.upb.de
<http://www.c-lab.de>, <http://www.hni.upb.de>

1 Motivation

In various disciplines knowledge-based systems have come into operation. In different applications the expertise of e.g. chemists, mathematicians, physicians or geologists has been compiled to create application-specific knowledge-based systems. However, the actual process of knowledge acquisition remains a challenge, because expertise is hard to articulate and formalize. This is especially complicated when multiple (domain) experts are involved in this task, because multiple expertise has to be transformed into one unique overall representation. Despite the complexity of knowledge acquisition of multiple experts there are several advantages of involving multiple experts in constructing a knowledge base: 1. in case of consensus a better understanding of the knowledge can be expected, 2. the quality of the knowledge base concerning validity, consistency, completeness, accuracy and relevancy is improved, 3. productivity is increased, 4. incorrect solutions are identified more easily, 5. broader domains are addressed and 6. more complex problems can be dealt with. Unfortunately, a lot of problems arise concerning experts having different mental models, time differences and geographical dispersion, difficulties in scheduling meetings, dominating experts or other groupthink phenomena.

Our objective is to present a framework which supports the benefits and overcomes the challenges which go along with knowledge acquisition from multiple experts.

This is done by letting multiple experts formulate and enhance a knowledge base on their own as long as no conflicting knowledge occurs. In case of occurring conflicts the involved experts are contacted and supported in negotiating and agreeing on an optimal solution.

2 System Description

We would like to demonstrate a development environment that allows experts to compile their knowledge collaboratively. The core of our development environment consists of an integration platform which combines and gives access to modules for formulating, evaluating, analyzing, visualizing, and negotiating knowledge.

Our system provides a visual language for formulating knowledge in form of if-then rules. Experts are enabled to assemble rules using pre-defined components of our visual language. In order to avoid contradictory or redundant knowledge base contents, knowledge to be added or modified is evaluated using the consistency check module. Every time an expert tries to modify the knowledge base via the integration platform a consistency check is performed automatically. In case of inconsistencies, the expert gets an overview of all inconsistencies including the relations between causing and affected rules. As experts should not delete or modify rules of other experts in favor of their own, coordination facilities help to contact all authors of affected rules and therefore help to initiate a negotiation. Detailed visualizations of affected rules and their simulated behavior for different input values support experts in analyzing the cause of inconsistencies. Cooperation facilities, synchronous as well as asynchronous, allow experts to communicate and exchange information. Thus, experts are able to reflect on each other's opinion, discuss viewpoints, and achieve an agreement on the conflicting rules.

Negotiation or more general coordination, communication and cooperation turned out to be dominating factors concerning knowledge work.

We will demonstrate our multi-user system using several scenarios within the application of design-oriented critiquing systems. Scenarios concern generating and publishing rules, analyzing conflicting situations, initiating, preparing and performing a negotiation.

3 Implementation

The prototype of our integration platform is implemented in Java™. Our client-server architecture uses sockets for the communication between server and clients. For storing the contents of the environment objects the server uses ObjectStore PSE for Java. The user interface makes use of Java™ Foundation Classes (JFC) /Swing 1.1.1 in order to provide a customizable look and feel without relying on the native windowing system. Cooperation facilities are realized using JavaMail™ 1.1 and Java™ Shared Data Toolkit 2.0.