Chapter XII
A Case-Classification-Conclusion
3Cs Approach to
Knowledge Acquisition:
Applying a Classification Logic Wiki to
the Problem Solving Process

Debbie Richards
Macquarie University, Australia

Megan Vazey
Macquarie University, Australia

ABSTRACT

In this chapter, we postulate that the problem solving process in many domains involves identifying the class of problem on hand, identifying an appropriate solution, and recognising opportunities for its reuse. We suggest a solution that builds up knowledge of a given domain by recording observations, diagnoses and actions in a “3Cs form” of Cases, Classifications, and Conclusions. Our solution allows knowledge workers in any domain where heuristics are relied on to form classifications, and then apply generalised conclusions on the basis of the given classification, to collaboratively refine and expand a topic by consistently asking users to confirm, add to, or refine the presented knowledge in the context of the current case being classified. Our solution is presented in the context of the corporate call centre and is a significant extension of the Multiple Classification Ripple Down Rules algorithm. We present a 3Cs Logic Wiki that takes the best features of current collaborative knowledge exchange mechanisms, and captures a logic structure on top of that which provides for rapid indexing of acquired knowledge.
INTRODUCTION

The move from product-based to service-based industries in developed countries is clearly seen in the central role now played by an organisation’s support centre. In many cases the success of the organisation will depend not on the superiority of their product but on how well they handle support for that product. This is particularly true for the IT industry due to the complex nature of IT products.

We are presently involved in a project to improve the success of a sizeable multinational support-centre operating in the IT industry. In this context, success can be measured by the effectiveness and efficiency by which customer problems are handled, for example: reduced problem incidence, increased customer self-service, increased automation of problem diagnosis and solution matching, increased accuracy of solution matching as measured by reduced case revisits, increased solution re-use, reduced duplication of solutions, rapid fault and enquiry resolution times, increased customer satisfaction, increased in-line self-learning by support centre staff, increased staff satisfaction, and reduced staff turnover. As we can see, the solution is needed by all stakeholders, which includes customers, knowledge-workers, management and the organisation, and thus the solution must meet a wide range of goals including fitting in with the organisational culture and daily workflow.

This chapter is organised as follows: first we identify the problems that IT organisations face in their management of software and hardware products followed by a review of some of the solutions that have been offered. We then introduce the Multiple Classification Ripple Down Rules technology that we have adapted. Next, we describe our methodology and approach including the extensions needed to support the specifics of the support centre domain. In the final section we provide our conclusions and the current state of the project.

The Problem

Knowledge Management for Software and Hardware

The inherent difficulties in software, identified by Brooks (1987) as complexity, conformity, changeability, and invisibility, have ramifications not only for software engineering but for the management of knowledge related to that software. Since Brooks’ landmark paper, the need to both change and conform to complex environments has increased beyond all expectations. For example, in earlier times, acceptance, integration and stress testing were performed with users, hardware, platforms, applications, inputs, and throughput that could be identified before the project started. For many systems that is no longer realistic. Old strategies such as user training to compensate for product shortcomings, designed to pass on the bridging knowledge, are no longer viable in cross-vendor and e-commerce applications.

Knowledge management is not just a problem for software. Managing knowledge about hardware has become more difficult since the (relatively) simple mainframe of the 1980s has been replaced in the 1990s with smaller, less expensive open system and windows servers that can be inexpensively clustered and failed-over as needed, along with dramatic improvements in disk drive capacity. Nowadays the most critical issue is usually data unavailability, data loss, or poor performance, rather than the loss of a single host or server. Discovering the cause of these can be both time consuming and difficult, in complex environments involving multiple vendors, machines, software products and topologies, in an infinite number of combinations. It is no longer possible to expect a single expert to quickly find and resolve such issues. A better approach is needed, to allow both the accumulation of knowledge with guided trouble shooting techniques, along with interfaces to all other relevant knowledge bases.