Abstract

This chapter introduces Problem Frames as a framework for the analysis of sociotechnical problems. It summarizes the Problem Frames approach, its techniques and foundations, and demonstrates, through theory and examples, how it can be applied to simple sociotechnical systems. The chapter continues with the description of an extended Problem Frame framework that allows the treatment of more general sociotechnical problems. This extension covers social components of a system — individuals, groups or organisations — bringing them within the remit of the design activity. The aim of the chapter is to make the Problem Frames framework more accessible to the software practitioner, especially those involved in the analysis of sociotechnical problems, as these problems have so far received only scant coverage in the Problem Frames literature.
Introduction

By sociotechnical system we mean a collection of interacting components in which some of the components are people and some are technological. In this chapter we focus on the requirements analysis of sociotechnical systems in which some of the technological subsystems are computer-based, these systems forming the largest part of modern software design problems.

More precisely, there are two (not necessarily disjoint) sub-classes of sociotechnical systems that we will treat in this chapter. The first sub-class contains those systems in which existing components or sub-systems (that is, domains) are to be allowed, through software, to interact. An example from this first class might be the problem of designing software for the operator of heavy machinery. The larger second class contains those systems for which software, a user interface, and user instruction are to be designed to enable a new process or service. An example of this second class might be the development of a new customer call centre.

The use of Problem Frames (PFs) underpins our requirements analysis process. As described in Jackson (1998), PFs are a concretization of the ideas of Michael Jackson and others in the separation of machine and its environment’s descriptions. This separation is generally accepted as being a useful principle for requirements analysis. We will have cause, later in the chapter, in dealing with a more general class of sociotechnical problems, to further detail this separation, but nothing we do compromises its fundamental status.

The usual representation of the separation of machine and environment descriptions is as the “two ellipse” model, illustrated in Figure 1. In that figure world knowledge $W$ is a description of the relevant environment; $R$ is the statement of requirements; $S$ is the specification that mediates between environment and machine; $M$ is the description of the machine; and $P$ is the program that, on machine $M$, implements the specification $S$. The role of $W$ is to bridge the gap between specification $S$ and requirements $R$. More formally (Gunter, Gunter, Jackson, & Zave, 2000; Hall & Rapanotti, 2003; Zave & Jackson, 1997), $W, S |- R$.

One of the aims of the PF framework is to identify basic classes of problems that recur throughout software development. Each such class should be captured by a problem frame that provides a characterization for the problem class. Sociotechnical systems are
Related Content

Knowledge Integration in Problem Solving Processes: A Case Study - Perceptions of Workers
[www.igi-global.com/article/knowledge-integration-in-problem-solving-processes/119656?camid=4v1a](www.igi-global.com/article/knowledge-integration-in-problem-solving-processes/119656?camid=4v1a)

Preliminary Evaluation of a Software Security Learning Environment
[www.igi-global.com/article/preliminary-evaluation-of-a-software-security-learning-environment/120088?camid=4v1a](www.igi-global.com/article/preliminary-evaluation-of-a-software-security-learning-environment/120088?camid=4v1a)

Model-Driven Impact Analysis of Software Product Lines
[www.igi-global.com/chapter/model-driven-impact-analysis-software/49163?camid=4v1a](www.igi-global.com/chapter/model-driven-impact-analysis-software/49163?camid=4v1a)
Analysis of Existing Software Cognitive Complexity Measures
[www.igi-global.com/article/analysis-of-existing-software-cognitive-complexity-measures/204524?camid=4v1a](www.igi-global.com/article/analysis-of-existing-software-cognitive-complexity-measures/204524?camid=4v1a)