

High Level Process Modeling for SCM Systems

Jacky Estublier, Samir Dami, Mahfoud Amiour
LSR, Bat 8 Actimart, Av de Vignate. 38610 Gieres FRANCE
{jacky|dami|amiour}@imag.fr

Abstract. SCM processes require a close control of the product evolution, a global view of the involved activities, and user friendly interfaces. In other words a good support of SCM processes requires services currently found in different domains; Software Process technology of course, but also CSCW, WorkFlow or Software design.

Currently, most SCM systems have fixed processes either hard wired or through a specific tool which implements these processes. The other SCM systems define processes through State Transition Diagrams, exceptionally through a straightforward control flow; the formalisms used for describing these processes being script like languages.

This rather rudimentary technology contrasts with the demanding processes involved in SCM. We advocate in this paper, that SCM systems need a high level, wide scope and executable process formalism, intuitive enough to be understood by managers as well as by developers.

The paper focuses on how processes, and more specifically concurrent engineering ones, can be described and supported based on a high level modeling tool called APEL (Abstract Process Engine Language). Based on its underlying formalism and concepts, we show that concurrent engineering policies can be addressed along four orthogonal dimensions controlling respectively the topology, isolation, public area and concurrent changes. Any actual cooperative policy is a point in this four dimensional space.

1. Introduction

The size and complexity of software products, together with the resources and time needed for their development and maintenance continue to grow in concert with the pressure towards reducing time to market and improving software quality. Many companies have realized that this considerable challenge can be met by relying on software process technology, based on the principle that product improvement cannot be achieved without improving the processes through which they are constructed. It is thus not surprising that Software Configuration Management (SCM) applications pay considerable attention to process technology.

It is generally accepted that processes must not be hard wired in an SCM system but should be explicitly defined. Further, without an explicit model at hand the effects of