Adding Control Integration to PCTE

Huw Oliver
Hewlett-Packard Laboratories
Bristol BS12 6QZ, England
e-mail: heo@hplb.hpl.hp.com
Copyright ©1991, Hewlett-Packard Company

Abstract

The PCTE interfaces provide data-integration services. In a good Software Engineering Environment (SEE), however, more is needed: control integration to automatically start tools and share services. We report on our intermediate practical experience of adding control integration to PCTE. More precisely, we show how Broadcast Message Services can be layered on the PCTE platform thus forming a SEE framework that spans the tool integration dimensions.

Keywords: CASE, Software Engineering Environments, PCTE, Soft-Bench.

1 Introduction

The computer support environment provided for software engineering today typically consists of a set of standalone tools. These tools are monolithic. These tools do not usually cooperate. They cannot access each other's functionality. They have no access to each other's data (and would not be able to understand it if they could). Their user-interfaces differ widely.

The tools are monolithic in that they provide many of the services more naturally provided by the framework within which they operate or by other tools. For instance, some document processing tools today offer version control even though it is also provided by configuration management tools. Such tools provide so much because the tool providers have no way of composing tools from small modular pieces.

We are interested in how the framework can provide different types of composition or integration services. These integrating services would help tools to be smaller, more modular and built into the support environment as needed by the software engineer.

A complete support environment for software engineering will be a large, complex system. Neither the high level of financial resources nor the wide range of expertise required to provide all the elements of a support environment will be found within a single organisation. The use of open standards for these elements is an essential enabling factor for the production of quality SEE implementations.
We have looked at two technologies which provide elements of a support environment and investigated how they might be combined. The technologies are SoftBench [2] [4] and PCTE [7] [8] [9].

SoftBench is a product of Hewlett-Packard. It consists of an integration framework and an integrated set of tools.

PCTE stands for “a basis for a Portable Common Tools Environment”. PCTE defines an interface to support CASE tools and development environments. PCTE itself does not provide any tools: it is a framework on which to build and integrate tools. The development of the interface has culminated in the ECMA PCTE abstract specification [9] which the ECMA general assembly adopted as an ECMA standard in December 1990.

We have undertaken a prototyping activity to show how these components can be combined. The goals of our prototyping activity are to investigate how to construct a support environment, to learn how to use it and to examine the benefits of working with it. We are using an implementation of version 1.5 [7] of the PCTE interface in our prototyping activities. We report here our intermediate technical results from constructing the prototype SEE framework in the HP research laboratories.

2 Integration Services in an SEE Framework

The ECMA CASE environment framework reference model [1] identifies and defines integration services that a framework may provide to support a SEE, and groups related integration services together. Figure 1 shows the overall structure of the reference model (this is a conceptual architecture not an implementation architecture).

The reference model (RM) can be used to categorise the services offered by an SEE framework. Although the ECMA RM activity was spawned from the ECMA PCTE Standards committee, the RM is completely independent of PCTE. The RM can be used to position standards proposals and commercial products, and helps to understand the relationships between different framework offerings.

This section quickly sketches the services required of an SEE framework in terms of those detailed in the RM.

The RM identifies three main aspects of tool integration:

- Data Integration (addressed by the data repository plus data integration services) is the sharing of data and descriptions of that data (schemas) between the users and tools of the support environment.
- Control Integration (addressed by the task management plus the message services) is the management of cooperation between independently developed tools to achieve a coordinated effect.
- User Interface Integration (addressed by the user interface services) is a common look and feel for tools.

2.1 Data Integration

The maintenance, management, and naming of data entities or objects and the relationships among them is the general purpose of the data repository services.