An Introduction to Process-Centered Environments

B.C. Warboys,
Department Computer Science,
Univertsty of Manchester ,
Manchester M13 9PL, U. K.

brian@cs.man.ac.uk

R. A. Snowdon,
ProcessWise Portfolio Centre,
ICL, Kidsgrove,
Staffordshire ST7 1TL U. K.

ras@kid01pml.icl.co.uk

To be published in:

'Software Process Modelling and Technology', edited by A. Finklestein, J. Kramer and B. Nuseibeh, Advanced Software Development Series, Research Studies Press.

CHAPTER 1

An Introduction to Process-Centered Environments

R.A. Snowdon and B. C. Warboys

ProcessWise Portfolio Centre, ICL, Kidsgrove, Staffordshire ST7 1TL, U.K., and Department of Computer Science, University of Manchester, Manchester M13 9PL, U.K.

1. Introduction

Development Environments based on the ideas of Process Modelling are concerned with the problem of how computer systems can be used in the development of software based information systems. Two popular acronyms used to describe such environments are PCSEE (Process-Centered Software Engineering Environments) and IPSE (Integrated Project Support Environment). These have come to describe computer based systems providing a variety of facilities working together to support people who develop software based information systems. "IPSE" was derived from the similar "Ada Programming Support Environment" (APSE) describing an integrated set of tools for the development of programs in Ada, and will be adopted throughout this chapter.

The problem area of concern to an IPSE is that of computer systems which provide a set of facilities to support people (primarily software engineers) and organisations (as opposed simply to individuals) concerned with all aspects of computer systems development. In fact the concerns are both more limited and more general than this.

They are limited in the sense that the scope of an IPSE is inevitably ill-defined; it is always possible to think of something else that could be provided in an IPSE. They are more general in the sense that it is quite clear that not only is there a vast number of different computer systems, but there are also a great many ways of

developing them. Systems constructed to support the development of computer systems must therefore be, to some extent, general purpose.

2. Motivations for a support system

At the risk of over simplification, the behaviour of most companies can be described in terms of the model shown in Figure 1.

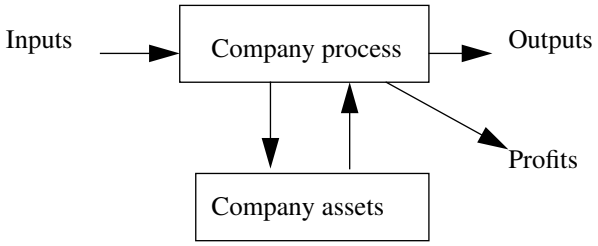

Figure 1 The company process

A company uses its "assets" (financial, intellectual, material) to add value to its "inputs" in order to produce "outputs" from which it can directly, or indirectly, increase its assets and make profits. This we may call the company "process".

For any given company in any given business it is clear that it must (generally) continue to improve its ability to carry out its business. This might mean that the cost of doing so must be reduced or that the quality of its outputs must be improved, or both. The company process must become more effective. A critical aspect of this is the management of change.

Over the years many basic tools have been produced to support the production of computer systems. These include for example, high level programming languages and compilers, testing systems, tools for building systems from components, management tools and a wide variety of others. However, tools such as these are only part of the answer. It is clear that simply having suites of tools which support specific activities is not sufficient, particularly in the context of the larger and more complex systems that are now being developed.

Thus the idea of a computer based system aimed at supporting the company process rather than the use of computers as individual tools within the process is seen as an important concept. This "process support system" is not a new concept; the basic notion of a business support system is a well established concept in several areas of computer applications. The application of these ideas to the complex area of information systems development is, however, more unusual. The gains to

be made in corporate effectiveness, over and above gains to be made by simple technical advancement are, however, very important.

Whilst a process support system should be of great benefit to a company producing information systems, it is still the case that the individual tools and techniques by which computer systems are developed are at present primitive. The reliability and quality of computer based systems produced by the application of these techniques still leaves much to be desired. A support system which simply allows companies to apply such techniques more effectively does not directly address these basic problems. As new techniques emerge it is necessary for them to be properly, and cheaply integrated into the support system concept.

3. What is an IPSE

An IPSE is a computer based system supporting those parts of the overall business process which are concerned with the technical development and maintenance of an information system - the software engineering process. This process might start at the requirements definition phase, continue through the stage when a product is completed ready for shipment to a customer and conclude with activities concerned with supporting the product in use. An IPSE might be concerned with supporting technical activities outside this scope - the exact delineation need not be well defined. Generally, however, an IPSE will not be concerned with the business or commercial aspects of the overall company process. This does not mean that the management and control aspects of the production process are outside of the scope of an IPSE. On the contrary, these aspects (sometimes referred to as the "meta-process") are very much an IPSE concern! The exclusion of business and commercial aspects is essentially an artificial division and derives from historical reasons as much as anything else.

As in Figure 2, the IPSE provides the environment within which individuals work. As far as each individual is concerned, the IPSE provides him/her with an appropriate working context. Actions carried out by the individuals working within an IPSE will cause changes to those working contexts. Of course, changes may also be brought about by actions initiated from other sources, either external to the IPSE or from within it (e.g. tools). The appropriateness of working environments to the particular tasks to be undertaken will have significant impact on the effectiveness of working within such an IPSE. Of course people play the most important roles in developing information systems. An IPSE must therefore be seen as providing support for human beings as they carry out their activities within the overall development process.

In terms of integration there are two possible directions. 1). Integration of the development process with the product (reflexive systems). 2). Integration of the product with its environment, i.e. inclusion of the users as part of the system definition (workflow, CSCW systems).

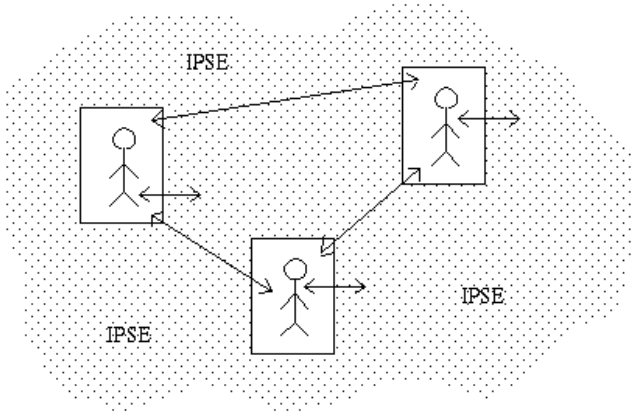

Figure 2 The working environment

Particular IPSEs (i.e. particular computer based systems) will vary in the facilities provided as well as the scope of the overall process which is addressed. It may be the case that a specific IPSE provides support for systems to be developed in only one programming language or for one type of computer. Support may be concentrated on a particular method of systems design or for small or large projects. The degree of "integration" will vary from an IPSE formed from a loosely compiled set of tools to one where the facilities provided are tightly and coherently bound and presented to the users.

In general an IPSE provides a means of storing components and descriptions of the system being developed and a set of tools which can be used to create such components and descriptions, and ultimately the system being developed.

Most existing IPSEs are either almost completely concerned with providing support for the technical functions within the development process or provide relatively separate support for technical and managerial activities. Such IPSEs are based on a "store plus tools" model which emphasises the *product* being produced rather than the *process* which is being followed. This has the effect that such IPSEs are not able to provide appropriately distinguished working environments as there is little or no information available about the state of the process in order to be able to do so.

4. What is a Process-Centered IPSE

The property which primarily characterises a "process-centered IPSE" is that the level of integration is raised above the store-plus-tools level to a point where the concern is for the process by which information systems are developed.

The process of producing an information system is much less well understood and requires a significant amount of monitoring and adjustment. These processes (of monitoring and adjustment) are essential parts of the overall process and must therefore be supported by the IPSE.

An IPSE provides a means of holding information about products and projects. It supports configuration management and version control, it provides means by which existing tools can be used and it is equipped with state of the art UI facilities to ensure the achievement of effective man-machine synergy. One of the major goals of process-centered environments is to provide all of this integrated with Process Modelling in a clear and cohesive manner.

5. What is Process Modelling

The term Process Modelling is used to describe the production of models of software development processes and the use of these models in an IPSE.

The idea may be illustrated by analogy, by considering, for a moment, not a support system for the development of information systems but an automated manufacturing plant, maybe a production line for motor cars. Such a plant might comprise a number of machines, each capable of carrying out a particular operation on objects passing through the plant. The successful operation of the whole plant requires the proper synchronisation and control of all the machines and of the flow of objects between them. This is the responsibility of a "controller", provided in the plant by a computer system. We assume that raw materials are made available to the start of the process (in fact to the first machine in the line), operated upon in the appropriate manner and order by the individual machines in the plant, with the final manufactured objects emerging at the end. The "controller" system must be provided with some sort of overall model of the plant and the required behaviour so that it can cause the individual machines to operate at the right time. This model might be "hardwired" into a special purpose controller which can only control very specific processes, or it may be loaded into a more general control machine much as a computer program is loaded into a general purpose computer. As the control machine "obeys" the model, so the appropriate process is enacted by the real machines in the plant.

There may be some conceptual distance between an automated manufacturing plant ("populated" by machines) and a company producing information systems ("populated" largely by human beings), but the analogy should be clear. The "controller" in the manufacturing plant is part of the system supporting the process.

Thus it might be expected that there be something similar in an IPSE supporting the process of system development. The Process Model to be loaded into an IPSE would hardly be as straightforward as that for a manufacturing plant. It is one thing to describe the sequence of operations for a machine, it is quite a different (and generally unacceptable) matter to expect to be able to prescribe in detail the behaviour of a person or persons carrying out the intellectually demanding tasks of information systems development. It is however, quite reasonable to consider a style of model which is, perhaps, descriptive of the *goal* of a task as opposed to the imperative form that might be characteristic of an automated manufacturing plant.

Of course, the factor of "change" is one which is of much greater significance in the process of producing information systems. In an automated manufacturing plant, the process is relatively well-defined and may be left to run with little intervention (for such is the intention of automation). As mentioned in Section 4, the process of producing an information system is much less well understood.

6. Support and use of Process Models

A key component of an IPSE support system is the Process Engine (PE) which can be loaded with a Process Model of the activities to be carried out by the staff of the company or project using the IPSE. The Process Model could be of an imperative nature, but more generally it will be less stringent, describing the tasks that need to be carried out and the constraints that should be observed.

According to what is described in the Process Model the PE provides appropriate working environments for each of the staff using the IPSE to carry out the tasks involved in developing a system. These tasks may be technical or managerial. In the former case the staff member will make use of tools as provided in his working environment to develop further the system being produced. These tools may perhaps be tools developed elsewhere (and incorporated into the environment through standard interfaces) or may be tools specifically developed for the IPSE and more tightly bound into the environment. The environment will also provide a storage component (and configuration and version control facilities) in much the same way as current support systems do, except that access to the store will be controlled through a working environment and hence, ultimately under control of the Process Model.

The working environment provided for managerial activities will generally provide additional features concerned with the monitoring and changing of Process Models.

Figure 3 shows a view of a process model expressed as a network of activities linked by interaction channels for entities. In this diagram, the current set of activities are indicated by their lying on a time line labelled "present", The model to the left of this line represents the past, whilst that to the right represents a prescription for what can happen in the future. This future is rather like a plan. By definition, this "plan" has been produced by an activity (usually described as a managerial

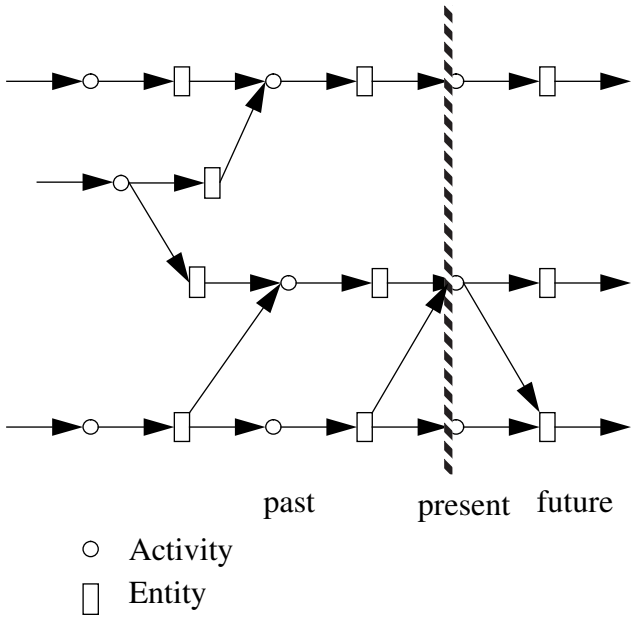

Figure 3 Past, Present, and Future

activity) at sometime in the past. Such an activity is therefore amongst those to the left of the "present" line. Management activities are part of the process model. Thus a process model not only describes the technical activities to be carried out in the development of a system, it also describes the managerial meta-process as well. Very importantly, it is necessary that it be possible to support activities concerned with monitoring and change, so that the process model can be altered as necessary to align better with real intentions. This particular area offers some challenging problems, particularly in terms of the implications of dynamic binding and the need to control the impact of change in an executing model.

In addition to the technical and managerial users, there is an engineer user who is responsible for ensuring that the IPSE itself operates properly. Appropriate working environments for tasks associated with such activities are established in the same manner via Process Models loaded and activated by the PE.

7. Tools

The description so far has concentrated on the idea that software engineers involved in the process of developing an information system are assisted by the

provision of working environments appropriate to the tasks they have in hand. As tasks are carried out so the environment changes, both for a particular individual and for others affected by his or her work. No mention has been made so far about how tasks are carried out, in particular the provision and use of computer based tools. In the development of information systems, it is absolutely necessary that use be made of computer tools to carry out certain tasks. As part of the working environment provided in order to carry out some activity, an IPSE must provide access to relevant tools and ensure that use of these tools is properly integrated with other relevant objects. Thus, if an activity is likely to require the application of some analysis tool to some object, then the IPSE should ensure not only that the analysis tool is available, but that it can be applied to the object. In some circumstances, the use of a particular tool may not require its selection by a person, but rather that the tool is applied automatically as a consequence of some other action. In any case, there is a clear requirement that computer based tools be considered as an integral part of the *process*.

The implication is that the Process Modelling Language, and thus the PE, must provide means by which tools can be incorporated into an IPSE. In fact, there are different types of tool which must be considered. The most important are the vast number of tools which already exist, having been built for particular purposes and, in all likelihood, for particular computers and operating systems. As far as these are concerned, means need to be provided, through PML and through tailorable constructs within the PE, by which such tools can be added to and used in an IPSE system. At the other extreme are tools which are really just a form of process model and are completely defined in PML. These tools are, of course, peculiar to the IPSE culture and, whilst being easily integrated with the rest of a model, cannot be made use of in other situations.

Finally there is an important class of emerging tools which already contain their own form of private embedded process descriptions. These represent a considerable challenge to the architects of process-centered environments in that they require a solution which not only addresses the issue of incorporating tool invocation, but also addresses the problem of the integration of embedded and heterogeneous process model fragments. These range from functions such as configuration management and organisational directories to real process fragments.

8. **Acknowledgements**

We would like to thank ICL Technical Journal for permission to use extracts from “An Introduction to the IPSE 2.5 Project”.