An Implementable Meta-process

Ian Robertson
Informatics Process Group
Department of Computer Science
University of Manchester
Oxford Road
Manchester M13 9PU, U.K.
ir@cs.man.ac.uk
http://www.cs.man.ac.uk/ipg/

To be published in:

Proceedings, Second World Confrence on Integrated Design and Process Technology, Eds MMTanik, FB Bastani, D Gibson, PJ Fielding. Society for Design and Process Science, 1301 West 25th Street, Suite 300, Austin, Texas 78705-4236, USA, December, 1996.

Abstract:

This paper is concerned with the problem of supporting, through I.T., those organization processes which co-ordinate creative and intuitive human activity. It describes the implementation in an I.T. system of a behavioural meta-process, and its use to bring about change to an enacting instance of a simple process. That process is called Simple Banking, and it is expressed in an active model which both monitors the state of the enacting process and also directs activity of process participants. The meta-process, known as the Process for Process Evolution (P^2E), is used to guide the application of changes to the SimpleBanking process.

Introduction

Much work has been done on the modelling of operational or production processes in business. Not only have they been modelled, but a number have been supported by I.T. using workflow-like technologies. Only moderate success has been reported by Garg and Jazayeri (1996), however, and one reason for this appears to be the fact that the technologies do not accurately model the processes, and another is that the technologies do not lend themselves to handle change very well.

Some research workers have noted the existence of the property of 'entropy' in human activity systems which include projects or organization processes. This property is exhibited as the natural tendency of these systems to become more chaotic with time (Humphrey (1990), Lehman (1992)). A consequence of this is that the systems and their incorporated business processes must be well maintained by continual evolution solely in order to maintain initial levels of efficiency and effectiveness. Substantial investments by an organization in developing a (necessarily 'incomplete and imprecise') definition of a business process can be lost if the resulting process cannot freely evolve in response to influences and to changes in circumstances, environment, or resources.

This paper describes the application of an enactable meta process as a way of addressing these problems. 'Enactable' means that it is capable of being represented and executed by some IT system. It thus supports, guides and controls (as appropriate) human activities by interacting with process participants. A 'meta-process' is a process which is responsible for creating and adapting a process.

This work addresses the problem of change by advocating the use of a meta-process as a way of managing inconsistencies between real world and modelled world, providing users with a means of adapting the process model during enaction to eradicate inconsistencies as they arise. The Process for Process Evolution, P^2E, is proposed as a suitable meta-process to address such concerns. Meta-processes have been studied, in one form or another, ever since Royce's landmark paper (1970) and are most recently addressed by, amongst others, Madhavji (1991) and Conradi et al (1994). However, even these recent models seem to be too abstract, too complex, and unable to address the reflexive nature of the problem. Furthermore, they are not readily implementable.

This work has been undertaken in the domain of software engineering, and the software design and development process, however the issues which arise are much wider than this relatively narrow domain. The chosen example is in the filed of financial services.

This paper sets out to focus attention on the need for, and the practical exploitation of, an implementable meta-process. It does this by describing a generic behavioural meta-process and demonstrating its application in supporting the creation and evolution of an organization process. in a controlled fashion, using commercial available enactment technology. The following section briefly introduces the enactment technology used as a vehicle to test the concepts. After a brief discussion on process models, the example operational process is described before introducing the Process for Process Evolution. The implementation of the example process and its evolution are then demonstrated. Ultimately, some conclusions are drawn.

The Technology

An I.T. environment is one which provides a suitable work context for the users of the system. The IPSE 2.5 project (Warboys (1989)) provided such an environment which tailored work contexts of users in a way that was determined by the stage of the process in which the users were engaged. Thus it is representative of Process-Centred Software Engineering environments, or a Process-Sensitive environments. This environment is commercially available as ProcessWise Integrator[1].

The system uses the role/activity/interaction paradigm for defining the mechanism for causing transformation events, and provides for their enactment in the same context. It is a framework within which tools can be effectively co-ordinated to service a process. It comprises three elements: the Process Control Manager (PCM), the User Interface Servers, and the Applications Servers.

The environment holds a model of the process network described in Process Management Language (PML) (Roberts and Jones (1989)). The semantic model is the process programming language, there being no generic model, and the application-specific model is defined as a program in that language. The language identifies four principal classes: *entitie*s (records), *activities* (procedures), *roles* (encapsulations of data and behaviour) and *interactions* (corresponding uni-directional communications channels). These principal classes, along with numerous primitive classes, can be used to define the process network. There are three language interfaces which together control the evolution of the model: they allow a class definition to be compiled into a class value, a new instance of a role class to be created, and a role class value to be re-defined.

The heart of the system is the PCM; that component which identifies, locates, and schedules activities which can be executed. It does this by scrutinising each role in turn and examining the guard property of each action which determines *when* that action is allowed to be executed.

1. ICL, ProcessWise Portfolio Centre, Kidsgrove, UK.

Process participants communicate with roles through the medium of a ‘pseudo role’ known as user agent. The User Interface representation is hierarchic, offering, at the highest level, an agenda of the roles bound to a participant’s user agent. Each of these opens into an action agenda allowing selection by the user.

Change is brought about by re-compiling definitions of role objects, and re-instantiating them. Any user action must be completed before proceeding with the change. A user cannot interact with a role whilst it is being changed. A powerful reflexive feature of the technology is that a role may be defined to apply change to itself.

Process Models

Most process models are passive, in that they represent a subject in the real world at one moment in time. On the other hand, an active (Greenwood et al (1995)) process model is one which maintains a constant modelling relationship to its subject. As the state of the subject changes, so the process model contained in the I.T. system is changed to reflect that state. These changes can, through the process model, be interpreted and fed back to the real world to drive process activity as process participants interact with the IT system. In this way, the process ‘knowledge’ of the organization is maintained in the IT system.

Process definitions exist in a number of forms and the different forms may merit different approaches to evolution. Conradi et a (1994) identify three variants of process model:

- Template model. This is an abstract definition, sometimes referred to as a Generic model, of a set of activities to achieve some desired purpose. It is resource-independent and also independent of organization. It is constructed as to be readily customisable, or tailorable, (or indeed specialisable if definitions possess the property of inheritance). This model is passive and not executable.
- Enactable model. A template model variation expressed in an executable language which has been tailored or specialized to such degree that it could be enacted in a specific organization to achieve a specific purpose.
- Enacting model. An enactable model fully adapted to the resources available and actively guiding human and machine activity by being executed in some computer system.

They also identify three kinds of change propagation: *delayed* (the template only is changed), *busy* (change propagated to enacting and enactable variants), and *local* (expeditious change to an enacting instance where change to the template is not warranted).

In this work we are concerned with template change (in the sense that the template is a generic process which must be adapted), enactable-delayed change, and enacting-busy change. Conventionally, workers have considered process models to be complete if the transformation of inputs to outputs was fully defined, however a process really can only be considered fully defined if the means of its change is also defined. A complete process model thus comprises not only the 'primary' or 'operational' process, which defines activity required to produce artifacts or to provide a service of value, but also some process which will bring it into existence and which will facilitate its evolution in response to external and internal influences during its possibly extended period of existence (for our purposes, the processes of process creation and evolution can be considered to be identical, and so 'evolution' in this work refers to both initial development and subsequent maintenance of a process model). This latter process is the meta-process.

To exemplify the different kinds of process, Figure 3 is a generic or template process, Figure 1 an enactable process, Figures 6 and 7 show enacting processes.

The Example Process

The process to be evolved is the Simple Banking model first described by Snowdon (1992). This is a very simple service process which has been chosen because it can be readily understood and also to illustrate how processes which link responsibilities across all levels of organizational hierarchy can be modelled as an active model. In the scenario, a customer interacts with people at different locations and levels in the bank's organization. Version 1 of this process is shown in Figure 1.

This shows how the different roles which are associated with the organization (Customer, Account, Simple Branch Manager, General Manager) are linked through interactions. A customer (interacting via an A.T.M., say) can only make an initial deposit or terminate an account, the Branch Manager can start off new instances of customer roles, and the General Manager is notified whenever an account is opened or closed. There would typically be one instance of General Manager, a number of instances of Branch Manager, and a number of instances of Customers and Accounts, one Account role for each Customer role.

Large rectangles represent roles - encapsulations of behaviour associated with a task or responsibility. White boxes are actions which involve human participation, black boxes indicate machine activity. Lines between boxes indicate change of state and precedence; arrows represent message passing or implementation of behavioural change

Version 2 of this model is shown in Figure 2, and differs in the Customer and General Manager roles. It allows the customer to make interim deposits and withdrawals, and also provides the General Manager with a running total of the number of accounts opened or closed, as well as the specific notifications mentioned before.

The challenge addressed in this work is to change the process of this fictitious organization from Version 1 to Version 2 at *run time* in a structured and systematic manner.

The Process of Process Evolution

This change is provided by another process model, originating from concepts which emerged in Management Science. Complex 'management' activity (which endeavours to control operational processes) is characterised by apparently random and non-deterministic behaviour. However certain consistent patterns of activity have been identified and thus it seems reasonable to assert that some managerial work is in fact process centred and can therefore be supported by process-centred environments.

Part of the IPSE 2.5 Project was the study of the integration of management and work and the outcome of this was the 'Process Model for Management Support' (PMMS) (9). This is a generic process component which models certain concerns of management and which reflects the commonly recognised but poorly defined 'management cycle' of Planning, Organising, Monitoring, and Controlling. These matters are not remote from the concerns of the design engineer. Humphrey (1990), in the field of software development for example, asserts 'We need more traditional management, not less'. Earlier work, such as (Kellner (1991)) has focussed on the support of discrete management activities: here the management process itself is of prime interest.

The PMMS has been further developed by Warboys (1996) into the Process for Process Evolution (P^2E) to address such specific concerns as the management of complexity (of problem domain), consistency (between real and model worlds) and integrity (of methods). It is modelled in terms of the management cycle in a way which effectively combines an operational process and its controlling meta-process - in other words, 'work' and 'management' are integrated.

Figure 3 shows a graphical representation of P^2E, showing the components with the following generic behaviour:

- Manage: The identification of objectives, the receipt and analysis of feedback

- Technology: The determination of generic methods (process models) to satisfy objectives.
- Realize: The adaptation of methods to suit the problem and organization in which the solution method is to be implemented.
- Doing: The enacting process model. The physical manifestation of the method, where people interact with people and machines, being guided in their work by the I.T. system.

The management cycle, as it is appropriate to process management and control, has been interpreted in the P^2E activity framework as a separation of concerns where 'Controlling' means the evaluation of feedback (if any) and the establishment of objectives, 'Planning' is composed of the determination of the method to address the objectives (Technology), and the developing of this method into an implementable plan for use in the 'Organising' of the work (Realize). 'Organising' is a combination of Realize and Doing, and 'Monitoring' is done automatically with results fed back to Managing for evaluation.

P^2E is thus a loosely defined notion centred around the establishment of a goal or objective, the determination of a method to achieve a goal, its enaction or execution, and the monitoring of this enaction to ensure a successful outcome. It facilitates the determination of a 'Method' (akin to a kind of plan), in the form of an abstract process model, then the structuring of the specialising of this method with respect to both the objectives and the organization. The Method, now in the form of a 'plan', an enactable process model, can be implemented. The implemented process model can then furnish some feedback to Managing allowing, if necessary, re-modelling of the objectives. Thus the cycle can be repeated.

Implementing the process

'Implementing' means the bringing into existence of an enacting process model as an essential component of a real world process. In the current work only technical changes are be addressed: the manifold other concerns involved in change implementation, the human and hardware issues, are outwith the scope of this work.

A) Specialising and implementing P^2E

The behaviour already described for the P^2E provides the essential generic concept (i.e. it is a template) but is too abstract to be of assistance in addressing real organizational objectives, and so there has to be some adaption to enable the P^2E to address the situation in hand. The behaviour of the template model must be modified to tailor it to the subject problem, the organization in which it will be enacted, and the resources available. The result of the specialisation of the template model to the enactable model are shown in Figure 5. The General Manager has the role of GM SimpleBank Inc. and, initially, can initiate Terms of Reference by invoking a text editor, thus committing the objectives of the organization. These are available to the other roles in the P^2E: Realize and Technology. The user bound to Realize can pass the ToR to Technology, where the user must either find an existing generic process model to satisfy the objectives, or invent one and code it in PML. This process model, known as a Method, is returned to Realize where it is implemented and bound to specific users in the organization. 'Doing' represents an interface between P^2E and the operational process. Figure 4 is evidence of its implementation in ProcessWise, each P^2E component interpreted as a ProcessWise role

Properties of the P^2E in the implemented model can be summarised thus:

- It supports the generation of objectives.
- It can generate a structure which can be used to determine or to create a generic process solution for achieving that objective.
- It can support the adaptation of this solution, using the inheritance properties of the process modelling language (PML), for the specific objectives, and for the actual organization context into a suitable enactable process model.
- It can implement this model.
- It can implement feedback from this model to allow revision of objectives if necessary.
- It can change its own behaviour.
- It can reproduce itself to structure a hierarchy of similar components.

Because of the reflexive property of the model, it is not necessary to have a complete definition of a process model in order to achieve the objective, so it is possible to address the frequent situation which arises in the achievement of business objectives, where the method to achieve these objectives might initially be unknown, i.e. the process model must be implemented before its detailed characteristics have been defined.

B) Implementing the operational process

As mentioned above, Realize must adapt the definition to the existing organization context of the model, however in this example there is no further specialization required of the model definition in Realize. Simple Banking at this stage involves only a template change and an enactable-delayed change (as yet, there is no enacting instance in existence). Its outputs are a) the full model definition, and

b) a re-definition of General Manager to enable feedback, consisting of reports, to be received and viewed.

An important point is that the General Manager has behaviour associated with two processes: in the meta-process, to set terms of reference, and in the operational process, to read reports. The users for Technology and Realize do not participate in the operational process of SimpleBanking.

Figure 6 shows the role structure of a full working enaction, with two Customers instantiated. Figure 7A illustrates how a Customer interacts with the system.

Evolution to Version 2

Although at this stage there is a live enacting model in existence, the process for changing it is fundamentally the same as for creating it. In this implementation, the scenario would proceed thus: the General Manager wants a slightly different process. The user interacting with that role addresses the problem by changing the Terms of Reference to include the desired features. In turn, the Method (a new PML definition) is framed as a completely new version of the Simple Banking process model definition (this activity being undertaken in the role Technology by, say, users responsible for process improvement). In this case, the new version is kept as an incremented set of definitions (as opposed to only modifications), and is furnished to Realize.

The enactable variation is easy to apply: the definition of the model resident in Doing is re-compiled with changes resulting in the new variant. This ensures that, from the current time, any new instantiation of the process definition, for example for a newly constituted branch of the organization, would conform to the new definition. This is an example of *enactable* process change

An enactable process model possesses certain bindings to users tools and other system resources. It is often the case that these bindings must be preserved during change. Thus the preferred implementation of a change to an enacting model is to apply the new definitions in a selective manner to the existing instances of process model objects. To change the roles of the enacting instances of the definition (*enacting-busy*), some thought has to be given to the clear sequence of implementation of the change, so that the model is always in a stable state. 'Stable state' means that all activities that can possibly be invoked can be brought to a predictable conclusion during transition. An example of this kind of implementation process would be:

1. Adapt Realize to enable it to implement the changes (i.e. to target appropriate change definitions to instantiated roles).
2. From Realize, change the process definitions of Simple Branch Manager, and to systematically change every instance of Customer and Account in existence.

It is important to appreciate that during the course of this change from Version 1 to 2 the normal operational activity of the process can take place, but at the end of it, all customers will be interacting according to Version 2, and at the conclusion of the change activity all roles follow the definition of Version 2, i.e. customers can open, deposit, withdraw, terminate (see Figure 7B), and account statistics are sent to General Manger as well as opening /closing notifications. The model can operate as the operational process: creating customers, customers opening accounts, terminating them, and Reports of these events being sent to the General Manager, as needed for the business.

Conclusions

This work has described an abstract meta-process, the Process for Process Evolution. It has been demonstrated in an IT implemntation causing evolution to an operating organization process.

A) Lessons Learnt

1. This implementation confirms the possibility of enacting an evolutionary control process over an enacting organization process.
2. This kind of meta-process can form the basis for constructing a holistic model reflecting organizational responsibilities at various levels.
3. The implementation of such an abstract process, even in the crude way illustrated by this work, forces confrontation of further issues, such as 'what is a method', and 'what do we mean exactly by specialize'.
4. A specialization of this meta-process appropriate to other domains such as engineering design seems possible, and indeed may even be practicable.
5. If the implementation is incorrect, feedback can lead to new objectives which will in turn lead to subsequent versions of the process. In this example, feedback comprises the reports giving the status of new and closed accounts, but it could be, for example, advice on inconsistencies for the General Manager to address.
6. It provides an interesting example of the integration of 'management' and 'work'.

B) Concluding remarks

Many organizational processes are well defined, repetitive, and change little during their life. An example would be one for claiming expenses. Others, such as the design process, are poorly defined, must be defined as the instance of the process itself is progressed, and must be sympathetic and responsive to the skills, experience and concerns of the participants. It is very unlikely that it will ever be possible to model such a process in terms of the precision needed to define the parameters of the supporting technology. By using a meta-process such as described in this work, it may be possible to bring the benefits of automatic support to activities which are characterized by patterns of changeable behaviour. By recognizing that inconsistencies are inevitable, P^2E provides a managing framework as an alternative to attempting precise process definitions.

Many different domains of human activity seek process solutions entirely within their own domain. By promoting the concept of generic processes it may be possible to approach solutions more in terms of adaptation of existing abstract solutions rather than defining new solutions *ab initio*. In so doing the reasons why these solutions need to be different will be illuminated.

To this end, further work must address the need to furnish guidelines for specializing the P^2E to suit particular circumstances. There are two issues here: the need to specialize the P^2E itself, and the specialisation of the domain-specific method into an operational process.

An issue which was not possible to explore in this model is to further exploit the reflexive potential of the core meta-process, such that each of the P^2E components, Managing, Realize, Technology are specialized to P^2Es themselves. This can accommodate the difficulties which arise in conventional representations of the meta-process through the need for reflexion. It also seems a promising concept to adopt where the process definition is unknown until enactment stage has been reached - the typical project situation.

This work re-affirms the feasibility and value of using a P^2E as an enactable meta-process and, moreover, to adapt this meta-process to any appropriate organizational domain and context. The work raises the question that there may be little intrinsic difference between the operational process and the process by which this process is controlled. The only significant difference is that the control process can change the operational process in an *ad hoc* manner, whereas the operational process can only change itself in ways that have been forseen. The concerns that we have about the production process: its quality, performance, and integrity, may equally apply to the controlling process. And if a process-centred environment can improve quality and performance of one kind of process, it might also be of benefit to the other.

Acknowledgments

This work was carried out at Manchester University, funded by the EPSRC through the ‘Assisting Process Performance and Utilization through Simulation’ project.

References

Conradi, R., Fernström, C., and Fuggetta, A. (1994). Concepts for Evolving Software Processes. In Finklestein, A., Kramer, J., and Nuseibeh, B., editors, *Software Process Modelling and Technology*. Research Studies Press Ltd., Taunton, Somerset, U.K.

Garg, P.K.,Jazayeri,M. (1996). *Process-Centered Software Engineering Environments*. IEEE Computer Society Press.

Greenwood, R., Robertson, I., Snowdon, R., and Warboys, B. (1995). Active Models in Business. In *Proceedings 5 th. Conference on Business Information Technology*, Manchester Metropolitan University, Manchester M1 3GH U.K.

Humphrey, W. S. (1990). *Managing the Software Process*. Addison-Wesley.

Kellner, M. (1991). Software Process Modeling Support for Management Planning and Control. In *Proceedings of the first International Conference on the Software Process, Manufacturing Complex Systems*. IEEE Computer Society Press.

Lehman, M. (1992). Evolution of Processes and Process Models. *IOPENER*, 1(4). Praxis Systems PLC,Bath, U.K.

Madhavji, N. (1991). The Process Cycle. *Software Engineering Journal*.

Roberts, C. and Jones, A. (1989). Dynamics of Process Models in PML. In *Proceedings of 5th International Software Process Workshop*, Kennebunkport.

Royce, W. (1970). Managing the development of a large software system. In *Proceedings of IEEE WESCON*.

Snowdon, R. (1992). An Example of Process Change. In *Proceedings EWSPT’92*, volume 635 of *Lecture Notes in Computer Science*. Springer Verlag.

Warboys,B.(editor). (1996). *Business Systems Engineering: a process approach*. In preparation.

Warboys, B. (1989). The IPSE 2.5 Project: Process Modelling as a basis for a support environment. In Madhavji, N., Schäfer, W., and Weber, H., editors, *Proceedings of the First International Conference on Software Development, Environments, and Factories*, Berlin. Pitman Publishing.

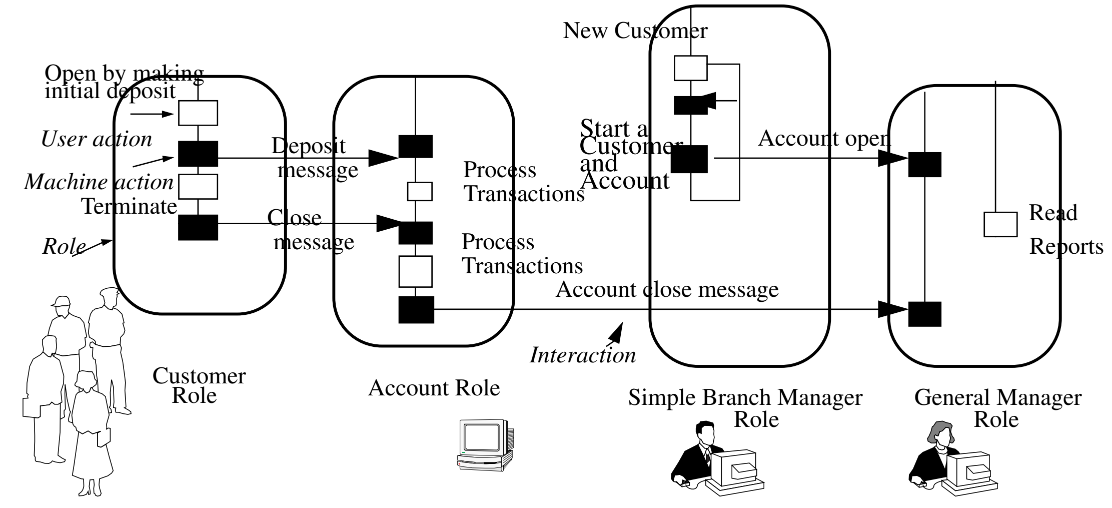

Fig. 1. SimpleBanking - the operational process, Version 1

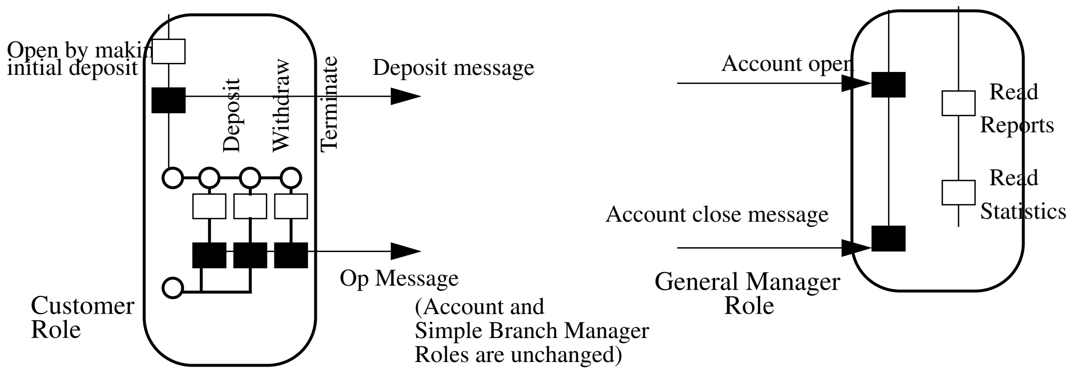

Fig. 2. SimpleBanking - Version 2

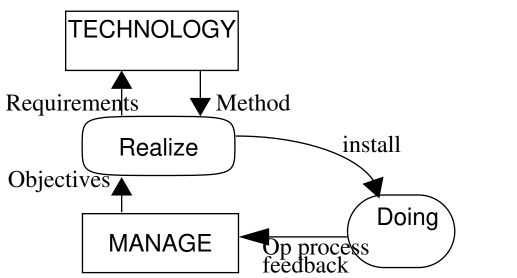

Fig. 3. The Process for Process Evolution (P^2E)

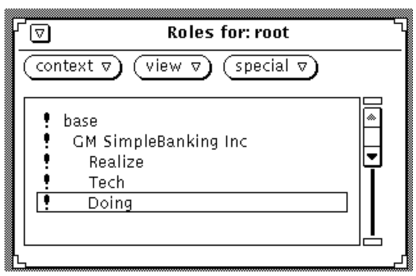

Fig. 4. The implemented P^2E

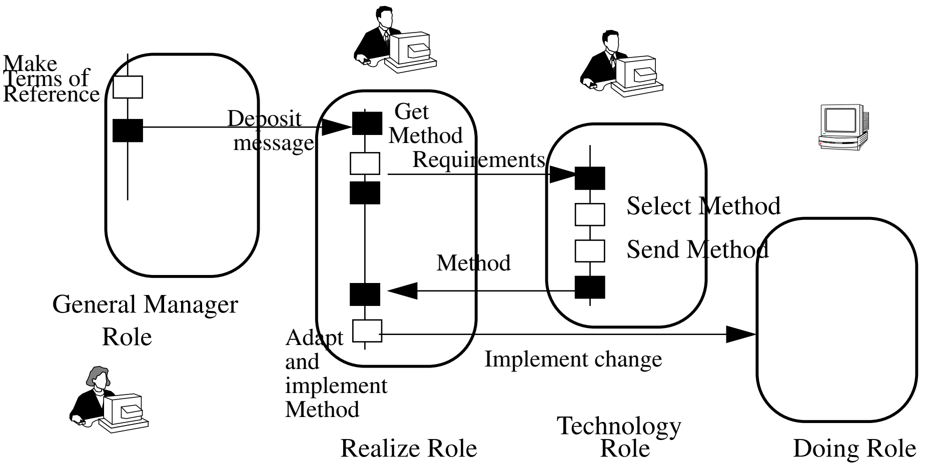

Fig. 5. P²E Specialized into an enactable model

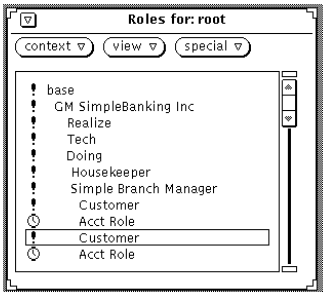

Fig. 6. The complete implementation

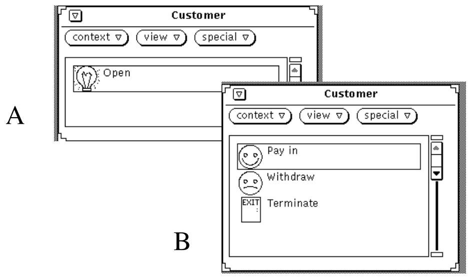

Fig. 7. User Interacting with SimpleBanking