

Supporting Software Processes: A Preliminary Evaluation of an Event Trigger Mechanism *

Noureddine Belkhatir Walcélio L. Melo

*Laboratoire de Genie Informatique
BP 53X, 38041 Grenoble Cedex France
{belkhatir, wmelo}@imag.imag.fr*

Introduction

The Adele product has been designed as a framework for supporting the development of Software Engineering Environments. Special attention was devoted for supporting development, enactment and control of large software processes. The model that was developed enhances our understanding of basic requirements for very large software management. The key requirement is the need to describe and enforce specific software policies, even when different policies are simultaneously managing shared objects.

Our model incorporates Object-Oriented concepts adapted for a better behavior modeling. Object behavior is described in both the object and relation types. Object behavior is modeled using event-condition-action (ECA) rules and interpreted by a trigger mechanism. ECA rule combined with an inheritance mechanism makes it possible to define intrinsic object behavior in two ways. ECA rules when defined in the object types expresses the context independent behavior; when ECA defined in relation types they expresses the context dependent behavior.

From the traditional concept of transaction, our effort has been focused on integrating the trigger system with transaction processing. The tuple (Pre Rules, Procedure, Post rules) forms a single transaction, while the tuple (After rules, Error rules) is executed outside the original transaction. More work is required in the domain of long/short transaction nesting.

Adele is an active Database Management Systems for Software Development Environments. The triggers are used to control state transitions as well as to model processes. The event modeling is based both on an explicit properties of objects and inter-object relationships, and on properties of the system state: the user, the action under way, its parameters, the system state, context, etc.

This system has been used to support European aerospace companies' maintenance activities. Although, good results have been produced, some drawbacks[1, 4] have been found, including: the formalism is low level for human designers; trigger execution is difficult to control; and the software process state is not very simple for human users to follow.

This is first because Adele kernel does not provide high level concepts like Work Environment, User policy, Synchronization, Deadline. This partly intentional, since Adele kernel is supposed to support a wide range of processes, no explicit PM concepts are included in the kernel.

Second, the process model is fragmented into different objects and relations types making it difficult

*Published in the *Proc. of the Process Sensitive SEE Architecture Workshop*, M. H. Penedo (Ed.), Sept. 1992, Boulder, CO.

to have a clear picture of the complete process. For human understanding, it would be preferable to group the process into a single unit.

We are developing, on top of our kernel a software process formalism and a process engine specialized in the specification and control of Work spaces. This language is called TEMPO. Here is presented some characteristics of TEMPO. More information about TEMPO can be found in [2, 3]

TEMPO: a software process model

The TEMPO model is designed for describing software processes in particular for implementing work environment control, process coordination and resource sharing.

The main capabilities of TEMPO are:

- The model used to describe processes is an object-oriented model. Each process step is a working environment which aggregates a set of entities. Each entities is managed by Adele-DB. When an entity is manipulated in a work environment it is considered by TEMPO as process resources. A resource “plays a role” in a work environment. The role concept makes it possible to customize the characteristics and behavior of a resource. Resource attributes can be modified, created and overloaded in order to reach the requirements of a process step. In the same way, resource behavior can be tuned, and specifics communication and synchronization operations can be described in order to take into account events generated inside/outside a working environment.
- TEMPO provides new facilities for working with object history and process evolution. We are extending the Adele’s trigger mechanism for supporting temporal events, i.e, in event-condition-actions rules could be specified with temporal logic predicates. In this way, long transaction can be better controlled; rules could verify object manipulation on the time by analyzing operations performed on it during the software processes; etc.
- TEMPO is implemented on top of the Adele system which is the abstract process machine. Software process models described in TEMPO are translated to Adele concepts (typed objects and relationships, event-condition-action rules). We make considerably use of the activity manager of Adele to support the process enaction. Methods can be associated with pre- and post-conditions to determine the order of execution of methods. Methods are executed only when the conditions are met.

We believe that the unification of these features improves the cooperative work among a team of developers and allows a well defined sharing of resources among a set of processes.

Overview of the TEMPO implementation

We are using the Adele kernel as the virtual machine that interprets and enacts the TEMPO language. The implementation in Adele of TEMPO involves the translation an Object Oriented description of TEMPO model to an O.O. data base with explicit relationships and triggers.

- TEMPO object are directly mapped to Adele object,
- TEMPO processes are represented as ADELE object types,
- roles are represented as active relationships,

- rules are represented as triggers associated to relationships and entities.
- methods and ECA rule's action are directly mapped to Adele methods and rules.

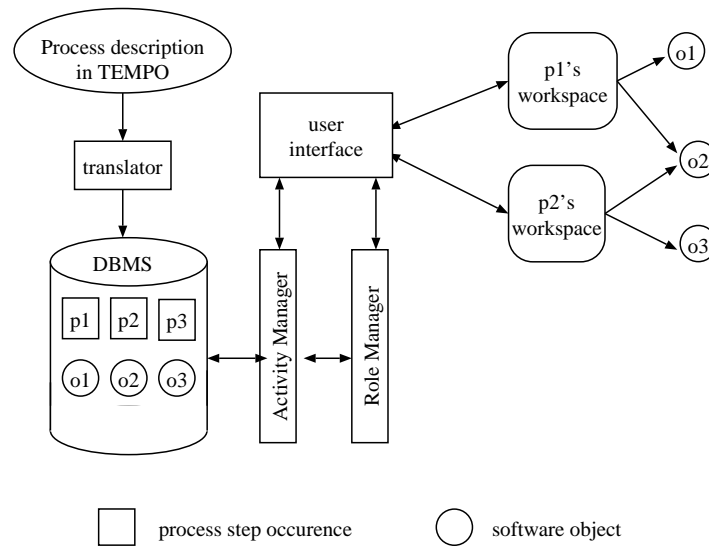


Figure 1: Overview of the TEMPO environment.

References

- [1] N. Belkhatir, J. Estublier, and W. L. Melo. Adele 2: a support to large software development process. In M. Dowson, editor, *Proc. of the First Int'l Conf. on the Software Process*, pages 159–170, Redondo Beach, CA, October 21–22 1991. IEEE Computer Society Press.
- [2] N. Belkhatir, J. Estublier, and W. L. Melo. Software process model and work space control in the Adele system. In L. Osterweil, editor, *Proc. of the 2nd Int'l Conf. on the Software Process*, pages 2–11, Berlin, Germany, 25 – 26 February 1993. IEEE Computer Society Press.
- [3] N. Belkhatir and W. L. Melo. The object role software process model. In J.-C. Derniame, editor, *Proc. of the 2nd European Workshop on Software Process Technology*, volume 635 of *LNCS*, pages 150–152, Trondheim, Norway, 7–8 September 1992. Springer-Verlag.
- [4] N. Belkhatir, W. L. Melo, J. Estublier, and A.-M. Nacer. Supporting software maintenance evolution processes in the Adele system. In C. M Pancake and D. S. Reeves, editors, *Proc. of the 30th Annual ACM Southeast Conf.*, pages 165–172, Raleigh, NC, April 8-10 1992.