Searching for PMIPS: Process Model Instructions Per Second

Mario Baldi, Silvano Gai,
Maria Letizia Jaccheri, Patricia Lago

Dipartimento di Automatica e Informatica
Politecnico di Torino
10129 Torino, Italy

Possible Architectures

The architecture of a generic process support environment (PSE) can be viewed as composed of communication services, operating system (o.s.), tools, workspaces (WSes), data base management system (DBMS), process engine (PrE), and user interface (UI) service. The PrE interprets and executes process models. The DBMS stores and manages both software products and process models. The communication service supports UI protocol (e.g. the protocol exploited by X-Windows), data transfer (e.g. ftp), tool integration, and inter-process communication. A distributed PSE architecture will encompass many machines ranging from terminals to mainframes.

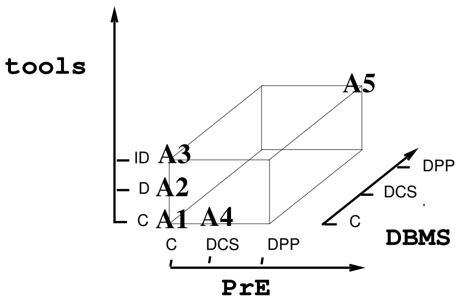

Fig. 1. Distribution possibilities of the main architecture components.

As exemplified in figure 1, there are three main distribution axes. Both the PrE and the DBMS can be either centralized (located on a server that serves the whole configuration)(**C**), or distributed over the sites of the PSE in either a client server (**DCS**) or peer-to-peer (**DPP**) fashion. Analogously, tools may be executed and WSes created and handled either on the process server (**C**) or on the client sites. In the latter case, tool execution can be either integrated (**ID**) or not integrated (**D**).

Five architectures are considered: in the mainframe architecture (A1 in fig. 1), all the components except the UIs run on a server machine. Each user is connected on a client site through a UI that communicates with the PrE by a UI protocol, e.g. that provided by X-Windows.

Architecture A2 is still client server. Tools and WSes reside on client machines, but are not controlled by the PrE, which resides on the server. Tool invocation can only be suggested (not enforced) by the PrE and is the user's responsibility. Data are transferred into and from WSes during the user working session.

Architecture A3 extends A2 concerning tool integration. The process thus enforces tool invocation and manages tool results, e.g. failure management.

In architecture A4, the PrE is distributed. It is assumed that the DBMS resides on the process server, the process model lies on the server, and the PrE manages the model and delegates the single instructions to each client. The differences and enhancements with respect to A3 solution are that UIs are completely managed on the client site, thus avoiding the UI protocol overhead.

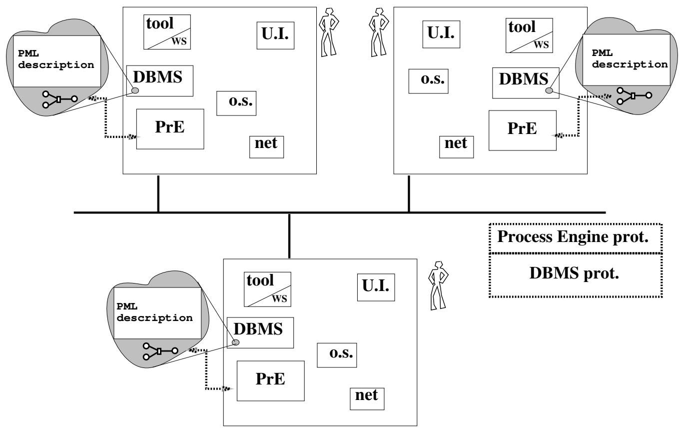

Fig. 2. A5: Distributed process and DBMS architecture

Architecture A5 is depicted in figure 2: the DBMS is distributed. Both software products and process model fragments are managed by peer PrEs. The enactable process model fragments are split among the different sites, while for simplicity a customized process model is assumed to be replicated on each site.

Further, client PrEs communicate with each other, and synchronize the process between sites. There is a process protocol which uses the functionalities provided by the underlying DBMS protocol.

Process Model Instructions

The process model instructions composing a generic process model, can be classified as follows: (1) communication (user-to-user or task-to-task), (2) workspace checkin and checkout, (3) tool invocation, (4) tool execution, and (5) data access.

Our goal is to develop a method to assess a PSE architecture with respect to the process models that will be enacted. Each model is characterized by a mixing of the above listed process model instructions. Architecture may be assessed measuring the average process model instruction execution time according to the given mixing.

We will describe what can be usefully measured as process model instruction execution time. Experimental ways of measuring are omitted here, for space reason.

The cost of a user-to-user communication is regarded as the cost of exchanging e-mail messages. For task-to-task communication we assume that the enacting process model resides on the server machine in all architectures but A5, and hence its cost is that of an inter-process communication on a single machine in all architectures but A5.

The cost of WS checkout is determined by the cost of reading a set of files from the DBMS and storing them in the appropriate WS. It will depend on the number of files, their dimension, DBMS performance, and DBMS/WS coupling.

The cost of tool invocation can be calculated from a weighted average of both interactive and non-interactive tool invocation costs.

The costs of tool execution depend on both machine load and the underlying hardware, while those for interactive tools also on coupling between the tool and its interface. If tool and UI are loosely coupled, i.e. they do not reside on the same machine, the costs are affected by network type and load.

Single data access can be measured from the DBMS access time to an object. Data access can be local data to either a centralized DBMS or a distributed DBMS, and thus may involve data transfers between machines. These data transfers are managed by the DBMS.

Conclusions

We have presented a taxonomy for PSE architectures and process model instructions. A method for measuring the performance of PSE architectures is given w.r.t. process models.

We would have liked to categorize existing PSEs according to the architectures we have presented, but papers about these PSEs always refer to out-of-date versions and it could have been easy to make mistakes or underestimations. An ambitious objective of our work is to yield benchmarks for PSE evaluation.

Further, the different architectures do not only influence execution time, but many other parameters such as change management, transaction management, etc. Work has to be done in those directions.

This article was processed using the LaTeX macro package with LLNCS style