Directions in Process Technology — An Architectural Perspective

Dewayne E. Perry

Bell Laboratories, Murray Hill, NJ 07974, USA
dep@research.bell-labs.com
www.bell-labs.com/user/dep/

1 Basic Trends in Processes and Technologies

There are two important considerations to take into account in proposing new research directions in process technology: the shape of future projects and the shape of future technology.

With respect to the shape of software development projects, there is an increasing trend towards globalization of development projects – that is, projects are increasingly being done in multiple locations, many of them in several countries simultaneously. This results in both geographical and temporal separation. Each of these factors affects the ways in which we need to think about process support. Geographical separation implies that we need to think about the ways in which these separated developments can be tied together and viewed as a single project development. Temporal separation implies that we need to think carefully about the ways in which we coordinate activities and interactions. Note that these characteristics will also apply to other types of business and industrial processes as well.

With respect to the shape of future technology, there are several trends that are extremely important:

- wireless networks are currently about 2 megabits per second and will rise to about 10 in 1.5 years, and 20-30 in about 5 years
- within wireless and cellular, there will be a merger of voice and data within the next 5 years
- the growth in computing power and storage capacity will continue with a growing emphasis on self-contained mobile computers with docking facilities
- the use of personal and point of entry devices will increase over the next 5 years and will be integrated into various business and industrial processes.
- increased inter and intranet bandwidth will easily support the use of multimedia facilities – in particular, the use of interactive video (though wireless network support will still be limited)

Thus there are three primary issues that arise from these process and technology considerations:

- distribution
- portability

– coordination

I will address these issues in the context of process architectures in the next section.

2 Process — An Architectural View

In this section I will address two of the primary issues mentioned above. As to portability, I merely mention that because of the heterogeneous nature of the computing devices to be found on both intra and internets as well as more tightly coupled distributed systems, process support technology (or at least certain parts of it) will need to be portable across a wide number of computing platforms.

Let us assume as a starting point the Perry and Wolf model of architecture in which an architecture consists of a set of elements, a form and a rationale. Elements are either processing, data or connecting elements and the form defines the properties of (or constraints on) these elements and their interactions. The rationale provides the reasons for the various elements and their properties and relationships.

In this architectural context, I will discuss in detail the implications for process technology of the other two primary issues: distribution and coordination.

2.1 Distribution — Architectural Implications

An important architectural principle that has emerged from my work with several development organizations on their various product architectures is that of distribution independence. In the one case, we were trying to define a generic architecture to cover both centralized and distributed systems which supported dynamic reconfiguration. In the other case, we were trying to define a product-line architecture at a domain specific level. In this latter case, the issues of distribution came to be viewed as more of an implementation issue than an architectural one.

I think that this notion of distribution independence is a key one for process architecture as well. Given the assumption (which I make, but not everyone in the process community makes) that the purpose of process support technology is to provide a virtual machine upon which humans perform their various processes, distribution becomes a basic assumption in the support of multiple people performing their processes individually and independently. Given the mobility of both computing and human agents, the system topology will be extremely dynamic. This serves only to enhance the need for distribution independence. By ignoring the problems of distribution and topological considerations at the architectural level, we can concentrate on the important aspects of the process elements and how they manipulate the data elements and how these interactions take place.

Thus, it is the underlying support that takes care of managing the problems and details of process distribution. The easiest way of doing this is to build the

process support on top of an appropriately configured distributed system, rather than to re-invent the solutions to distribution within the process support system.

However, it is key that the process support environment understand the various aspects of distribution and communication within that environment to be able to support the various facilities needed in a seamless and uniform way. As part of this seamlessness is the need to know where the processing agents are and whether they are available. Things like logins and establishing network connections will need to act as registration mechanisms. Similarly logouts and disconnecting from networks will need to act as deregistration mechanisms.

In this way, we should be able to support both people and the variety of independent computing devices that people will use as adjuncts to and support for their processes.

An important aspect of distribution is that of managing the various localities of process state and artifact space. Clearly there will be the need for some form of global process space that is shared across entire projects. Clearly also is the need for local space to be maintained for the various individual processing agents (both human and computational). There may be the need for various intermediate forms of collected space as well depending on the structure of the process architecture and the various aggregations that might be called for to support teams and larger forms of cooperation. Important issues in the management of these separate process state and process artifact spaces are those of visibility, sharing, migration and consistency. Of course there is also the need for both current spaces and historical spaces.

2.2 Coordination — Architectural Implications

Connectors have attained a significant status in the current software architecture community. They are treated as first class architectural components along with data and process elements. One of the reasons for this pre-eminence is that connectors enable one to separate coordination and interaction aspects of the architecture from the computational aspects. This enables one to separate the different levels of concerns in constructing systems out of components. It enables one to tie independent components together in novel and interesting ways via connecting structures control and facilitate the way in which the components interact.

In the past much attention has been given to activities and artifacts and very little to how the interactions between processes (and their subsidiary activities are defined and supported. Work on synchronous interactions have been explored by CSCW, both synchronous and asynchronous explored somewhat in the process community, and activity synchronization by the workflow and process communities.

Given a process system architecture that defines the properties of (or the constraints on) the various interactions among process system process and data elements, it is necessary that the various coordination and interaction fragments are available to be able to construct dynamically appropriate connectors to sat-

isfy both the constraints placed by the architecture and the constraints place by the topology of the underlying distributed system.

Connectors can then range from simple "direct access" to provide direct interactions (via computer facilities or non-computational facilities such as face to face meetings) to exceedingly complex mediated interactions requiring either synchronous or asynchronous cooperation (such as specifically configured CSCW or web-supported asynchronous inspections). Appropriate interaction support fragments would enable the the architecture designer, the process support, or the various interacting agents to create a customized connectors to support the demands and requirements of specific interactions.

Similarly, by specifying the desired properties of the interactions at the architectural level, the appropriate connectors can vary depending on the current topology. For example, if the interaction is required to be synchronized, then depending on the location of the various agents who are executing the interacting processes, the connector can range from a face to face meeting if the agents are all co-located to a customized CSCW connection if they are not with either audio or video depending on the types of topological interconnections.

3 Summary

While a focus on software processes may seem a bit narrow, there are good reasons for doing so. First, there are among the most dynamic kinds of processes to be encountered in business and industrial processes. Support that satisfies this level of dynamism will support any of the other processes. Second, although they are computationally intensive, they are at least as equally people intensive so that both aspects have to be addressed and support that satisfies these processes again will support any of the other types of processes.

The focus at the architectural level, given that we take a distribution independent view of it, enable us to focus on the important process issues such as process state and artifact spaces and the support of interactions among process system components. This means, however, that we must have an underlying process support system that is equally knowledgeable about the underlying distributed system and how to manipulate it and about the overlying process system architecture and how to properly support it in the context of a dynamic underlying distributed topology.