Process Support:
Inflexible Imposition or Chaotic Composition?

Geraldine Fitzpatrick and Jim Welsh

Department of Computer Science

The University of Queensland, Australia 4072

email: g.fitzpatrick,j.welsh@cs.uq.oz.au

tel: +617 365 2787

fax: +617 365 1999

Process Support: Inflexible Imposition or Chaotic Composition?

Geraldine Fitzpatrick and Jim Welsh

Department of Computer Science

The University of Queensland, Australia 4072

Abstract

Current process support systems seek to impose a process structure which controls the work of participants in cooperative work activities. This structure is based on the false premise that the global order emergent from the chaotic dynamics of work processes can be used to prescribe local activity which is in fact situated and unpredictable. We propose a spatial approach to the support of cooperative work processes. A space of work is defined, incorporating service, information, role and artefact objects, to capture the bounds and characteristics of the global order. The situated reality of work processes is supported by enabling participants

to dynamically compose objects into their own process path through the space.

KEYWORDS

process support, workflow, groupware, chaotic dynamics

INTRODUCTION

Computer Supported Cooperative Work (CSCW) is an emerging multi-disciplinary research field concerned with understanding the nature and requirements of co-operative work and how computer-based technologies can support, augment and enhance cooperative work arrangements (Schmidt and Bannon, 1992).

A cooperative work environment arises where there is interdependence in work (Schmidt, Heath and Hughes, 1993). Cooperation involves three broad functional aspects: communication, collaboration and coordination. Research particularly concerned with coordination is taking place in relation to software processes in the software development domain, and business processes and workflows in the business domain. A process is defined as the set of logically related, interdependent tasks and functions that combine to achieve some goal (Davenport and Short, 1990; Feiler and Humphrey, 1993) . This has led to the development of various process support (PS) and workflow management (WFM) systems (Fitzpatrick, Yang and Welsh, 1994).

Because the terms “process support” and “workflow management” are often used interchangeably in the literature and in the marketplace, we make the following distinctions in our use of the terms.

WFM is mainly concerned with simple, asynchronous routing and tracking of individual artefacts, e.g., an electronic document or form, according to some pre-defined script. In high volume form processing tasks such as document circulation, loan approvals and insurance-claim processing, WFM systems have reportedly resulted in positive outcomes such as shorter turn-around and response times, and better management information (Bartram and Youett, 1993).

Process support (PS) systems, on the other hand, attempt to embody a broader notion of work beyond asynchronous document flows, to incorporating multiple artefacts, activities and participants. However, experience with such systems to date, e.g., see Wastell and White (1993). indicate that cooperative work is often hindered, not enhanced, because the systems seek to impose too much control over how the work is performed.

In this paper, we illustrate the underlying assumptions that lead to this imposition of control on participants in the enactment of processes. We then propose an alternate approach to the support of cooperative work processes, employing a spatial metaphor, which allows the participants to compose their own processes in situ within a bounded space of work.

PROCESS SUPPORT BY IMPOSITION

Current process modelling work, e.g., Peuschel and Schafer (1992), proceeds in a similar way to the following simplistic illustration. The illustration is not meant to be representative of any particular methodology for process modelling or PS system, rather its purpose is to highlight some of the general problems of current approaches.

Consider the specification process which is part of a higher level software development process. Figure **??** (a) depicts a process model with computer-based support for process enactment linking the component activities. An activity may also incorporate the use of some other computer-based tool to support participants in the performance of the activity. This environment imposes a process structure providing execution support to control when, how, on what, and by whom, process activities are performed in relation to other process elements. All permissible work options, including path alternatives, concurrent paths, and feedback and feedforward loops, are predefined.

figure=spec2.eps

Figure 1: (a) Specification process model with enactment support; (b) Change of model to capture exception handling and process evolution events

This approach is based on three assumptions:

1. that all of the process can be explicitly articulated and hence modelled;
2. that once described, the model can be used to predict and prescribe work activities;
3. that a controlled process is a more effective process.

Even though process change has been identified as a major research area for process support environments (Huff and Kaiser, 1991), "change" is often interpreted as occasional deviations from the planned process model or as purposeful evolution. For example, in the specification process described in Figure **??**, a "one-off" need to seek the approval of the board before delivery of the requirements document would be an example of exception handling. The decision to include a verification step in all future processes would be an example of planned evolution. A new process model incorporating the change and evolution events is depicted in Figure **??** (b).

While this process example is grossly over simplified, it does capture the rigid, deterministic nature of process models as they are currently interpreted.

It is worth noting here that the determinism at issue relates in a "black box" way to the process components and how they fit together. Process participants still have considerable freedom, within the constraints of the tools that are made available to them, to creatively carry out the activity assigned to them. For example, the process model is only concerned that the requirements document is written, with the given word processor, at the appropriate stage in the process. It is not concerned with how the person actually goes about writing the document, e.g., whether she starts with an outline which she then expands or whether the document evolves with the analysis.

This freedom of the individual participants within each activity contrasts strongly with the lack of freedom to make one-off or evolutionary changes to the overall process. Within current process modelling approaches, such changes are the prerogative of the process manager or engineer.

COOPERATIVE WORK PROCESSES AS DYNAMIC SYSTEMS

CSCW research into the role of plans and procedures in work (Suchman, 1987), and experience with such systems in use (Wastell and White), suggest that process flexibility and situated dynamic enactment of processes play far more important roles in the performance of work than previously understood. This challenges the underlying assumption that a process can be wholly defined *a priori* and then enforced.

The problem here can be seen as being inversely analogous to the chaos problem.

Gleick (1987) describes how many early scientists had difficulty looking beyond the local, chaotic, apparent random behaviour of systems to see the surprising global order to be found there. Process engineers on the other hand, have stood back too far from the human systems of cooperative work and observed a blurred abstraction of the global order. From this perspective, requirements documents are certainly always delivered before specification documents are written in the process of Figure **??**, hence the justification for being able to model the process as a predictable sequential system.

An engineering or automated production line process could possibly be considered a predictable, stable, sequential system (although even this is open to debate). But the cooperative work processes of interest in this discussion have not been totally automated precisely because they need the creative, intelligent input of human beings. In our example, a computer would certainly have difficulty interacting with clients and deriving a set of requirements for a software system. Processes that involve human beings can only be described using terms such as complex, dynamic, non-sequential, unpredictable i.e, a chaotic system, simply because human beings bring free will and the capacity for creative, contingent, adaptive (and maladaptive) behaviour.

The creativity and intelligence of the people involved in the process is required not just for performing the component activities, but also for putting all the work elements together and, more importantly, keeping them together during the enactment of the process. This management work has been called "articulation work" (Strauss et al, 1985).

Process modelling work can be seen as performing some of the articulation work of the process by planning out who does what when how etc. The mistake is in thinking that this is all the articulation that is required. The critical articulation work occurs in responding to local conditions and contingencies as they arise, i.e., dynamic situated management. This is difficult enough when only one person is involved. However, when there is more than one person, as in a specification team or any collaborative activity, articulation work also involves the overhead of communicating with, and coordinating the people participating in the process as the situation dynamically evolves (Schmidt, Heath and Hughes).

Curtis, Krasner and Iscoe (1988) describe clearly the conflicting requirements, communication bottlenecks and breakdowns that occur in real world software processes. In fact, numerous workplace studies e.g., Suchman (1983), have shown that even work thought to be "routine" is in fact situated, contingent and problem solving i.e., in chaos terms, local, sensitive to local conditions and inputs, and unpredictable.

Therefore, we expect cooperative work processes to exhibit *chaotic dynamics.* Global pattern and order *can* be observed, as in the high-level steps of the specification process where delivery of the requirements document always occurs before the specification document is delivered, but global pattern does not imply local predictability in complex systems. Order emerges out of apparent disorder and unpredictability. Even though, as a collection of processes, they will look alike, each individual process will be unique in terms of both performance and articulation in the situated context.

We therefore reject the first two assumptions identified in the previous section: that a process can be defined a priori and that the resulting model can be used to predict and prescribe work.

However, given the enormous cost of inefficient and ineffective processes, we do not dispute that support for process enactment may be able to make an important contribution in facilitating group work and assuring an appropriate level of process validity in complex environments. For the third assumption, that a controlled process is a more effective process, we therefore propose a re-interpretation which reflects the perspective of the participant in the internal dynamics of the process rather than its current "external management" perspective: that a supported process is a more effective process.

PROCESS SUPPORT BY COMPOSITION

To define what a supported, instead of controlled, process means, we start from our notion of cooperative work processes as being complex, dynamic and non-sequential. While local behaviour in complex systems cannot be accurately predicted, it is not totally random. As stated by Stewart (1989) with regard to such systems:

> ... *we* can *still make very accurate predictions – not of the exact long-term behaviour, but of its general qualitative nature. We can impose quantitative limits on it; and we can determine its statistical features.* (p. 286).

We propose that this is also the case for cooperative work processes because the behaviour takes place in a particular work context or "space" where *"Every course of action depends in essential ways upon its material and social circumstances."* (Suchman 1987, p.50), i.e., there is a semantically rich and relatively well defined space, both physically and conceptually, which constrains and bounds the very possibilities and probabilities of that work. It is because of the characteristics of this space of work that we can assume the following: that processes will stay within certain bounds; that while they may never precisely repeat themselves because there are humans responding to local conditions in the space at a particular point in time, they will still trace out a distinctive pattern.

Therefore, instead of defining a single global pattern and enforcing it at a local level, an alternative approach to the support of cooperative work processes is to:

- Define and populate the environment in which the work processes take place, i.e. the process space. This space needs to capture the bounds and characteristics of the actual space in which work takes place, as well as the distinctive patterns of the process, i.e., both its qualitative nature and quantitative limits.

- Provide support for participants to dynamically compose their own process paths within the space, and using the features of the space, in response to local conditions. In this way, responsibility for the performance and appropriate articulation of the process can be returned to the participants, not the process engineers or manager, while still maintaining a level of process assurance.

Process Space Definition

How a process space is defined, bounded and populated is the subject of our ongoing research. The following represents an initial consideration of some of the relevant issues.

As stated previously, we consider that cooperative work processes already take place, not in a vacuum, but in a semantically rich and relatively well defined space that both structures and textures the work. Defining a space is similar to defining the static semantics of any computer-based application.

A top-down approach to the identification of the differentiating features of this space may include consideration of the following:

- The shared business goals and strategic objectives of the organisation/group and of the processes themselves e.g. a software house is concerned with the production of software, and the specification process with the production of a specification document.
- The organisational structures and roles, and the social norms which guide the way in which roles interact.
- The services to be provided by and/or required for the process.
- The customer(s) of the process.
- The "standardised" procedures and business rules that are applicable, (interpreted as guides rather than prescription for actions (Suchman, 1987).

A bottom-up approach can also be undertaken to identify the particular components of the process. It is important here that we look beyond the more formal, visible and easily definable aspects of the process to the informal and less visible aspects. This necessarily implies a consideration of the way in which the process participants work as a group, how they communicate with each other and how they deal with contingencies as they arise. These components may include:

- The activities of the process, the tools that may facilitate the activity and any relevant constraints associated with the activity.
- The temporal and functional interdependencies between the activities.

- The inputs, outputs (and intermediate products) and their constraints, e.g., that a specification document is in a certain format and medium.
- The particular responsibilities associated with roles.
- The common artefacts (Robinson, 1993), and modes and mechanisms of interactions (Schmidt, Heath and Hughes), through which work may be mediated.

figure=specSpace.eps

Figure 2: Populated process space for software specification.

Having defined the distinguishing features of the process space, these now need to be transformed into a computer-based process space. A computer-based space of work can be created by defining objects which encapsulate features and components of the actual space of work or work environment. These objects can then be used to support and augment both the performance of the process activities, and the dynamic negotiation and articulation of activities.

Our choice of objects is based on an understanding that people playing *roles* undertake *activities* often involving the production and/or manipulation of *artefacts*, employing the *services* of various tools and media, and with access to, or contributing to, various *information* resources. The objects may be from any of the following types:

Roles: these define logical sets of responsibilities which are undertaken by the participants in the process.

Activities: these are the basic performance components of the process to be carried out by participants (people in roles). An activity object may comprise a single task or a set of tasks, and their associated constraints. An activity may be a formal step such as "verify specification" or an informal step such as "compose correspondence".

Artefacts: these may be artefacts of "production" such as a specification document, or they may be artefacts used to mediate work such as the project schedule. Artefacts change as a direct or indirect result of process activities.

Information: these are the common information objects in the environment, e.g., publicly available resource or reference materials, "procedure" manuals, a record of the history of previous processes etc. Information objects are relatively stable compared with artefacts.

Services: these may include tools used in the performance of the activities e.g., a theorem prover for "verify specification" or an email tool to "compose correspondence", or other services.

For example, the space in which a specification process can take place is depicted in Figure **??**. The dashed lines represent the multi-dimensional nature of the space. The actual position of objects within the space is irrelevant - the objects can be regarded as "floating" rather than being fixed to any particular point.

Also the bounds of the space are meant to be fuzzy rather than fixed, denoting the most likely, but not exclusive, range of work possible in the environment.

A process space, as embodied in a computer-based support system, need not seek to totally replicate or replace the existing work environment. There are aspects that may serve no purpose if modelled in a support system e.g., the physical layout of the environment. There are other intangible aspects e.g., some of the cultural and political dimensions, which are central to configuring work processes but are difficult if not impossible to identify, let alone model in a computer. Nor is it expected that all work will take place within a computer-supported space.

However, a computer-based process space can make more visible and hence more accessible and manipulable, many of the tangible and definable aspects of work. More importantly, it can provide the mechanisms to facilitate the more informal and intangible aspects of cooperative work.

How is composition supported?

With the integration of a temporal dimension into the process space we have just defined, we can now talk of the work processes themselves. For a specific process at a particular time, process enactment occurs in the first instance via the instantiation of activity, role, and artefact objects, and the invocation of service and information objects as required; and in the second instance by the dynamic composition of the objects into a coordinated process in response to local demands.

Composition is supported in two main ways:

1. *Process plans:* The term “plan” is used in the sense of Suchman’s “guide for situated action”, an abstraction of the global order that must be interpreted in the local context.

   Process guidance, and a degree of process assurance, for composition is provided by defining a partial order over the activity objects, and defining relations between the objects. Process plans can then be developed, where appropriate, for (fragments of) processes.

   Where a process is relatively well understood (in the global sense), a process plan can be developed for the whole of the process. Otherwise, it may be possible to develop plans for process fragments or subprocesses where these can be easily defined. There will also be occasions, however, when the process is unknown except for the initial conditions, i.e., the initial set of instantiated objects, and some understanding of the goal to be achieved.

   Process composition occurs in context at a specific point of time. To aid the composition at each point in the process, the plan provides a range of “most likely” options. The choice of options is based primarily on the partial order.

   At this stage, it could be argued that the process space sounds as confining as the PS systems we rejected earlier. A significant difference here is that no one path, however complex, is assumed to be correct for all processes. While a (restricted) range of choices may be given, there is no obligation to conform. Flexibility and situated dynamic process enactment are central to the notion of process support, rather than being ad hoc and unwanted events. This is facilitated by the support provided for articulation during composition.

2. *Process articulation:*

   Where process plans are incomplete or inappropriate to the situation at hand, process participants are provided with other options to support the articulation and composition of process paths in context.

   (i) The plan options can be augmented by an analysis of previously composed paths or procedure definitions involving processes at a similar stage.

   (ii) The participant may make an entirely new choice of action, with the requirement that they annotate the choice. The annotation of "new" options yields important information for ongoing audit trails, outcomes analysis and hence evolution of the process plans.

   (iii) Auxiliary support for composition is also provided through many of the service objects, e.g., email, and common artefact and information objects. These facilitate the informal, ad hoc interactions among the process team participants, i.e., the negotiation of the composition, the shared decision