Fraunhofer Einrichtung Experimentelles Software Engineering

Process Guides: Effective Guidance for Process Participants

Authors:
Marc I. Kellner
Ulrike Becker-Kornstaedt
William E. Riddle
Jennifer Tomal
Martin Verlage

Accepted for publication in
Proceedings of the Fifth International
Conference on the Software Process,
Lisle, IL, June 1998

IESE-Report No. 012.98/E
Version 1.0
August 4, 1998

A publication by Fraunhofer IESE

Fraunhofer IESE is an institute of the Fraunhofer Gesellschaft.
The institute transfers innovative software development techniques, methods and tools into industrial practice, assists companies in building software competencies customized to their needs, and helps them to establish a competetive market position.

Fraunhofer IESE is directed by
Prof. Dr. Dieter Rombach
Sauerwiesen 6
D-67661 Kaiserslautern

Abstract

A process guide is a reference document for a particular process, providing assistance to process participants in carrying out that process. Although most organizations have such documents (e.g., standards, manuals), they are frequently deficient in both form and content, and often go unused. This paper describes a program of applied research on improving both the usefulness and usability of process guides. It first discusses an exemplary paper-based process guide. It then describes a prototype web-based Electronic Process Guide (EPG). The paper then introduces a key element of our vision for the future, Collaborative Process Guides (CPGs), which will be integrated with other forms of process technology as well as with collaboration technology. An incremental plan is outlined for developing and maturing the "technology" for this advanced process guidance (EPGs and CPGs). This plan can also be interpreted as a strategy for incrementally introducing process technology into an organization, avoiding user resistance and overcoming some of the perceived drawbacks of process-sensitive software engineering environments.

Table of Contents

1 Introduction

A major objective for process technology is to help process participants effectively, efficiently and accurately carry out a process. A variety of approaches have been developed in the past roughly dozen years. One of the most extensively developed approaches has emphasized the use of process machines as a basis for process-sensitive software engineering environments. The goals were primarily visionary [Winograd 86], and even after a dozen years such systems are only very rarely applied in practice [Christie 97]. This is not because process support is not needed. Rather, the wrong kind of support has been provided. Current process machine-based technology requires fine-grained, comprehensive descriptions of well-defined, relatively static processes. What is needed is support for relatively loosely-defined processes which may frequently change.

The work on process guides reported in this paper is a step in this direction. The authors feel that organizations should be able to use process technology right now – with only small shifts from their current behavior – as well as incrementally enhance the technology over time. Among other benefits, this would allow process technology to be incrementally developed, tailored, introduced, modified and corrected to suit real-world needs.

A process guide is a reference document for an intended process, providing guidance to process participants in carrying it out. Process guides contain at least process definitions, and may be extended by services for browsing and searching the definitions, storing process state information, and providing expert guidance. Some benefits of using a process guide are:

- They facilitate communication either off-line or on-line. Off-line communication might occur between authors and readers of process documents. On-line communication might occur between process participants performing related process steps.

- They help process participants track their work by capturing process event information (e.g., process participants might check-off steps as they are completed).

- They help process participants efficiently perform the process (e.g., guidelines reflecting prior experience might help participants select among alternative steps).

- They help participants return to a partially-complete process (e.g., state information might help participants understand what they need to do to restart work after an interruption).

Process guides are not necessarily computer based. Printed guidebooks, standards, process and procedure manuals, and the like, are widely used in industry. Frequently, however, these documents are not felt to be useful by their intended users. In addition, guidebooks are currently the major medium for communicating process changes. However, after participating in a number of process improvement and measurement programs following systematic approaches like CMM® [Paulk 93], IDEAL℠ [McFeeley 96], or QIP [Rombach 95], the authors believe there are major drawbacks to the traditional way of describing and communicating processes:

- Traditional guidebooks, even those made available on intranets, often lack key information.

- Readers of guidebooks can't easily navigate through the pages when their strategy of understanding does not match the document's flow.

- Guidebooks either contain a mixture of information for different audiences, or multiple documents tailored to specific needs require a common process description.

- Notes attached to paragraphs by one reader cannot be shared with others.

- Guidebooks are not designed to store information about the status of a project.

- Version control, especially of example development documents, is not well supported.

- Distribution of a new version of a process to process participants is unreliable.

This paper reports on a program of applied research focused on alleviating these drawbacks and improving both the form and content coverage of process guides. It describes advances in paper-based process guides, on-going prototyping of web-based Electronic Process Guides (EPGs), and a concrete vision for the future that we call Collaborative Process Guides (CPGs).

The central function of any process guide is to facilitate process understanding. Process participants need to understand what is expected of them in perform-

® CMM is a registered service mark of Carnegie Mellon University.
℠ IDEAL is a service mark of Carnegie Mellon University.

ing a given process. Lacking that, they cannot be expected to perform it effectively and efficiently. In fact, without participants understanding their role in the process, assistance – such as provided by technologies supporting workflow and collaboration (e.g., computer-supported cooperative work (CSCW) technologies) – is of little use in supporting process performance. Thus the necessity and importance of the central function of process guides is readily apparent. As will be discussed below, the authors believe that this function can be most effectively provided using information technology, as opposed to employing paper documents, relying on a person's memory, etc.

The outline of this paper is as follows. First, relevant background regarding process representations is summarized. Section 3 then describes previous work on information content that is pertinent to process guides. Next, experiences with building improved paper-based process guides are described in Section 4. Section 5 then discusses our on-going prototyping of EPGs. Section 6 discusses the next steps planned for this prototyping work. Our vision for the future is presented in Section 7. Finally, related work is discussed in Section 8, and in Section 9 we summarize the paper and express some conclusions.

2 Background on Process Representations

A *process representation* is a description, depiction, likeness, portrayal, etc., of a process. Process representations are created in order to support understanding complex processes, communicating about them, evaluating them, and so forth. Process representations are developed for particular users, user needs and requirements, and usage scenarios. Their intended use affects what they contain and how they are structured [Armitage 94b].

Various modes of process representation are employed in practice, primarily distinguished by form and usage, rather than by content. Three key modes are process models, process templates and forms, and process guides [Kellner 93].

2.1 Process Models

A *process model* is a relatively detailed, formal or semi-formal representation of a process. Process modeling is discussed in numerous articles and papers, books, and conference proceedings volumes, such as [Curtis 92, Heineman 94, Finkelstein 94, Christie 95, Fuggetta 96]. The primary users of process models are *process engineers*, who analyze, assess, design and monitor processes for continuous process improvement and process automation, and *process participants*, who perform the processes or are interested in their performance (e.g., project managers). Process models can support a wide range of uses, and some of the most common are:

- as a mechanism to help people understand and visualize a process (especially graphical models),
- as a basis for engineering (i.e., developing, evaluating, improving, etc.) a process [Kellner 96], and
- as the means of formalizing a process for machine-assisted enactment by a process-centered software engineering environment, workflow engine, or the like.

Numerous process modeling notations have been proposed and applied in practice. Several notations are compared in [Kellner 90, Curtis 92, Rombach 95]. Process models have been used to describe existing (as-is) processes and prescribed processes (e.g., standards, regulations), evaluate them for desirable characteristics and improvement opportunities, and develop and analyze new (to-be) processes. See [Kellner 89a, Kellner 89b, Kellner 93, Verlage 97,

Becker 97b] for examples. They have also been used to quantitatively simulate and analyze processes in support of management planning and control, and process improvement; for examples see [Kellner 91, Raffo 98].

2.2 Process Templates and Forms

Process templates and forms [Kellner 93] take the form of highly structured textual representations. Elements of information are organized and structured into predefined, named slots, and are often arranged in a hierarchical fashion within major types (i.e., organized similar to a database). Process templates and forms are intended to support organizing, recording and reporting process information. The primary users are generally process engineers.

For example, templates are often employed when gathering information through interviews during descriptive process modeling. Here they serve as interview guides and as a vehicle for recording the elicited information [Kellner 89b, Madhavji 94]. Some process work has been largely based upon forms and templates. For example, the Software Productivity Consortium has developed an extensive set of forms and templates for recording process information [SPC 96]. Forms and templates have also been used as the medium for reporting process information in a standard format to process participants. Additional uses of process templates and forms are discussed in [Kellner 93, Armitage 94a, Armitage 94b].

2.3 Process Guides

A *process guide* [Kellner 93, Armitage 94b] is intended to describe a particular process for the purpose of supporting human enactment of that process. Thus, process participants are the intended users of process guides. (In contrast, process engineers are the primary users of process models and of process templates and forms, although process participants are sometimes secondary users of them.) A process guide is a structured, work-flow oriented, reference document for a particular process, and exists to support process participants in carrying out the intended process. It should provide an explicit[1] definition of a process that applies to its intended operating context[2], and can be easily understood, communicated and followed. As reference material, a process guide can support and point to training but it is not, by itself, a vehicle for training. A process guide can also support process planning and certification.

---

[1] Explicit means leaving no question as to meaning or intent, i.e., fully expressed without vagueness, ambiguity, or implication.

[2] Operating context refers to the conditions under which the process will be performed. These conditions include environment and organizational infrastructures, who will use the process, their skill base, and their current level of understanding of the process.

Process guides generally employ both text and graphics (e.g., diagrams). They frequently contain portions of an underlying process model, although they should contain much more than simply a formatted process model. For example, they might contain detailed descriptions of specific methods, warnings about pitfalls, examples of work products, etc. They often contain structured narrative text, decision tables or decision trees, graphical aids, examples, work product templates, task checklists, etc. Process guides may be made available through various media, e.g., hard copy or on-line. They may also be organized in various fashions, e.g., traditional narrative text, structured text, or hypertext. One particularly useful technique (method and format) for developing process guides is Information Mapping® [InfoMap].

[Armitage 94b] contains a series of comprehensive checklists to be used in analyzing the content of software process guides. The checklists identify specific content and related issues that a process guide should address, and are discussed more in Section 3.3. More recent work, described in [Gates 97], has led to a comprehensive example of a paper-based process guide, which is discussed more fully in Section 4.

3 Information Content

Prior work [Armitage 94a, Armitage 94b] has identified the information content elements (and their interrelationships) that are recommended for inclusion in a process representation (e.g., a process guide, a process model) in order for it to be considered fully human-enactable (i.e., the process can be carried out as described in the process representation given that the organization provides the proper human and material resources). This section briefly summarizes this prior work to help the reader better understand what the authors feel should be represented in process guides.

3.1 Conceptual framework

Our recent work on process guides is based on our conceptual framework for software processes [Armitage 94a]. The things one typically considers when thinking about a software process are (1) what happens and how it is done, (2) what things are used and produced, (3) who does it, and (4) when it is done. These are illustrated in the bottom half of Figure 1. In our framework, a process is conceptualized as a set of activities, artifacts and agents, the relationships within and among those three entity classes, and the behavior of the entire set of entities and relationships. This framework is illustrated in the top half of Figure 1 and is discussed more fully in [Armitage 94a, Armitage 94b].

The framework identifies two subclasses for the class *agent*. *Functional agents* (commonly called *roles*) allow description of process-related, logical, performer entities in terms of their responsibilities within the context of a process. *Organizational agents* (often just called *agents*), on the other hand, allow description of organization-related performer entities in terms of their capabilities which may be applied in performing various processes. These subclasses are discussed and defined more completely in Section 3.4.

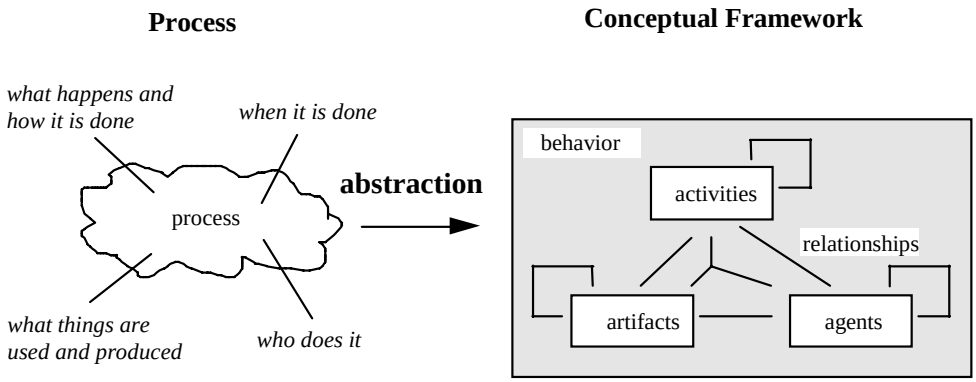

Figure 1: Conceptual Framework for Software Processes

3.2 Critical Process Information

The most critical, highest-priority elements of information and their interconnections are identified in [Armitage 94b]. This set of "basic" elements and interconnections defines a recommended starting point for use in designing and developing processes and their corresponding representations. The set identifies the most critical topics and issues to cover when building a human-enactable process representation. The topics and issues are defined by 16 groups of straightforward questions that must be addressed. All of these "basic" elements and interconnections have been included in our subsequently-developed process guide structures and examples.

3.3 Checklists

A comprehensive set of checklists for analyzing the content of software process guides has also been developed [Armitage 94b]. The checklists are intended to support process definers in reviewing and improving a draft process guide. The checklists identify specific content and related issues that a process guide should address. Nine checklists, comprising 123 questions, are provided.

This extensive set of questions is based on what a process participant would ultimately need to know (or be able to determine) in order to enact the process as defined. However, it is not mandatory for all this information to be documented in a process guide; some could be provided through training, mentoring, common knowledge, etc. Accordingly, [Armitage 94b] recommends that the checklists be tailored by organizations for their own use, often by removing items.

These checklists have heavily influenced our subsequent work on developing and presenting process guides, and the considerations identified in the checklists are reflected in this later work.

3.4 Schema

A conceptual schema for process information has also been previously developed [Armitage 94a, Armitage 94b]. This conceptual schema offers a single, integrated view (in a database sense) of the previously discussed elements and interconnections. The conceptual schema is sufficiently general that it can be applied to a broad range of methods, languages and approaches for constructing process representations. It builds upon the conceptual framework as a foundation and reflects extensive experience with process modeling and definition at both a conceptual level and in actual practice. The schema is quite detailed and contains 125 information content elements.

This schema has been revised and augmented and is being implemented in an object base system. Major extensions have been made to accommodate measurement concerns. With respect to process modeling issues, the schema has also been influenced by experiences gained with the process modeling language MVP-L [Rombach 95] and strategic dependency models [Yu 94], as well as numerous interviews with, and feedback from, practitioners of software process engineering.

In the expanded schema, the "basic" elements and interconnections are mapped onto the following classes:

- *Artifacts*: descriptions of products created or modified during process performance, either as a final or intermediate result of the process or as a temporary, "internal" result created by one step for use by another step.

- *Activities*: descriptions of "how things are done." Activities and Artifacts are associated via 'produces' and 'uses' relationships.

- *Agents*: descriptions of entities who can perform activities. The descriptions are in terms of characteristics such as skills, cost and availability. An Agent may be an individual or a group.

- *Roles*: descriptions of a set of obligations and permissions related to performing activities. Agents and roles are associated by an "assumes" relationship. Roles and Activities are associated by an "involved in" relationship.

- *Resources*: descriptions of computer programs, or other aids, which can be used to support or automate performance of an activity.

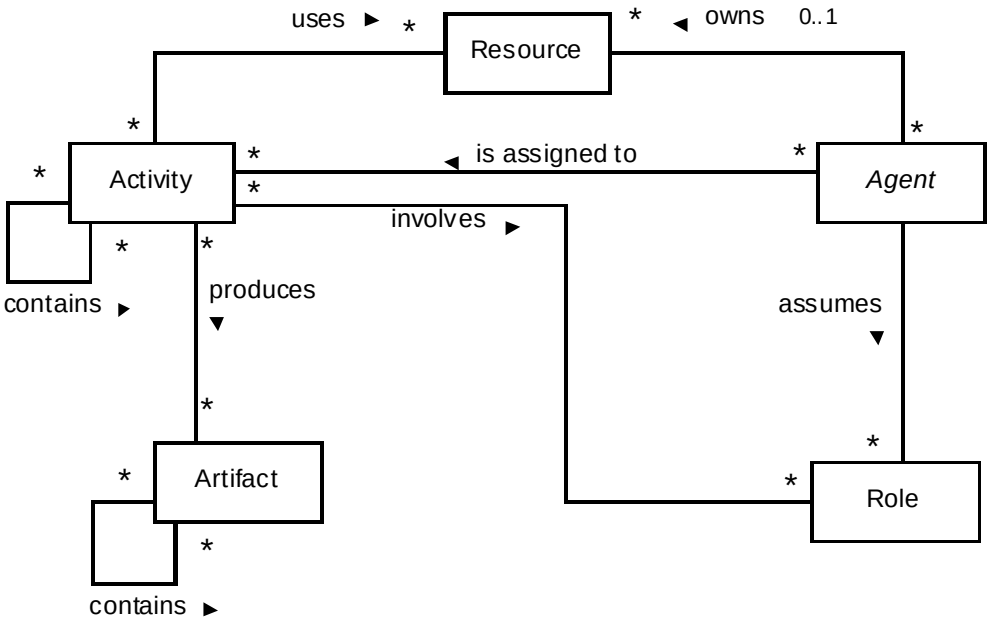

Figure 2: Excerpt of the Comprehensive Schema

In each of these classes, entities can be aggregated to form more complex entities or decomposed into less complex entities, providing a means of structuring process information. Figure 2 shows an excerpt from the expanded schema, illustrating some of the major interrelationships among some of the classes. The expanded schema contains classes and relationships other than those depicted in Figure 2. Some of these are discussed in the rest of this section. (A detailed description of the schema, as well as a comparison of it with the original schema, can be found in [Becker 97a].)

The schema defines the class *Relationship* to capture information about interrelationships among process elements. To support description of the measurement aspects of a process, the schema defines an *Attribute* class which may be used to define measurement-data attributes. A *Value* class is also defined to allow association of concrete values with attributes.

The schema supports one of several different ways to describe process behavior. It allows behavior to be described in terms of entry criteria (i.e., conditions which have to be true before an activity may be started), exit criteria (i.e., conditions which have to be valid at the completion of the activity), and invariants (i.e., conditions which have to hold throughout activity performance). These are subsumed in a *Constraint* class.

This schema is currently being implemented in an object-base system as the tool Spearmint (Software Process Elicitation Analysis Review and Measurement in an Integrated Modeling Environment) [Webby 98]. In its current version, the

Spearmint tool supports the classes described here, as well as product flow, role involvement, and decomposition relationships.

The schema is important for our present work because it provides a basis for storing the information to be presented in process guides. Storing the information in a database/object-base allows for better change management and for dynamic generation of process guide material; this is discussed further in Section 6.1.

4 Paper Process Guides

Many organizations, if not most, make available some form of process documentation – most commonly on paper. This is often in the form of process and procedure manuals, standard operating procedures, life-cycle descriptions and so forth. However, users are frequently dissatisfied with this documentation, and in many cases it is simply not used. An examination of process documentation from numerous organizations, and discussions of their shortcomings with those who are expected to use them, led to the conclusion that existing process documentation often fails to provide the necessary information in a suitable format. That is, most existing process documentation is deficient in both form and content.

This section first discusses what should be covered by a process guide and how it may be organized effectively. Second, some serious limitations of paper process guides are presented, even when they are 'complete' with respect to process information contents.

4.1 Recommended Contents, Structure, and Layout

In previous work on a prototype Process Asset Library (PAL) [Kellner 93], we developed seven process guide examples, ranging in size from 28 to 330 pages. Each of these guides was based on existing documentation from the source organization of that process, but tried new presentation formats and in some cases also elaborated the content. The experience gained through our PAL work, in conjunction with the information content work summarized in Section 3, has led to the development of an exemplary paper process guide.

The exemplary process guide [Gates 97] provides a comprehensive example of a process guide on paper, illustrating recommended contents, structure and layout for explicit process reference material. While it is believed to reflect good document design, it is by no means the only good way to assemble a process guide. The document design has been heavily influenced by the conceptual framework presented in Figure 1, by our work on information content summarized in Section 3, and by the principles and formats of Information Mapping®.

The exemplary process guide describes a descriptive modeling process (DMP) taught by the SEI. It includes structured narrative text, tables and diagrams, and is primarily structured mirroring the hierarchical activity decomposition of the DMP. Thus, there is a chapter for the overall DMP, followed by a chapter for each of the eight major activities into which the DMP is decomposed. Some of

those major activities are further decomposed into sub-activities, which are each addressed in separate sections of their parent activity's chapter. Each major chapter contains sections providing:

- an overview of the activity (e.g., purpose, activity context and decomposition diagram, inputs, outputs, roles and agents involved, effort guidelines, recommended measures, applicable policies and standards, concepts and references, and warnings),

- details regarding the activity itself (e.g., objectives, decomposition, behavioral diagram and notes, entry/exit criteria, functional diagram, input artifacts and sources, output artifacts and destinations, and role/sub-activity relationships),

- details regarding the artifacts pertinent to the activity (e.g., descriptions and applicable states of input, output and internal artifacts; artifact decomposition; other artifact relationships; and artifact storage location and retention period),

- details regarding the roles and agents involved in the activity (e.g., descriptions of participants, role/sub-activity relationships, responsibilities, and agent/role relationships), and

- details regarding each sub-activity in turn (e.g., objectives, context, task descriptions, warnings, behavioral diagram and notes, entry/exit criteria, functional diagram, input artifacts and sources, output artifacts and destinations, and role responsibilities).

To provide a sense of the "look" of the exemplary process guide, sample pages appear as Figures 3 and 4. Following the principles of Information Mapping®, information is highly structured and organized into small, manageable, understandable units called "chunks." Moreover, information is clustered in such a way that, at any point in the process, a participant may readily locate whatever information is needed within a single section or at most a chapter. While this is very convenient for the user of this reference material, it does result in considerable replication of material within the process guide.

The exemplary process guide incorporates diagrams providing functional and behavioral perspectives [Curtis 92] on the process. The functional perspective is essentially a data flow diagram, and the behavioral perspective is in the form of a statechart (see Figure 4). These are simplified and stylized versions of two of the perspectives used in the modeling approach, based on Statemate [i-Logix], developed at the SEI (see, for example, [Kellner 89b]).

3.2 DMP: Activities

Introduction

The DMP produces a descriptive process model. This model is composed of a set of hierarchically and sequentially arranged activities. Each activity is described briefly in this section.

Objectives

The objectives of the descriptive modeling process are to create a process model that provides a foundation for the following:

- understanding and communicating the as-is process
- evaluating the as-is process
- standardizing and managing the process
- analyzing and documenting proposed changes
- developing training and support materials
- pilot testing and installation

Activity Decomposition

The following activities are the high-level activities performed in the Manage the Modeling Work and Model the Process partitions of the DMP.

Activity	Description
Plan	The Plan activity establishes the objectives of the product (the model) and ensures the successful completion of the effort.
Develop the Team	The Develop the Team activity ensures that all DMP team members have been adequately trained and prepared to execute the DMP process.
Manage the Project	The Manage the Project activity assigns resources and tracks the project according to the plan.
Contract With Management	The Contract With Management activity obtains management commitment and support.
Conduct Process Familiarization	The Conduct Process Familiarization activity establishes a modeling frame of reference by translating existing process documentation into an initial model.
Collect Data	The Collect Data activity gathers the data for building the process model.
Construct Model	The Construct Model activity translates interview and other data into a process model.
Review Model	The Review Model activity verifies and validates the representation of the process.

Figure 3a: Sample Process Guide Page #1

3.2 DMP: Activities (continued)

Behavioral Diagram The following diagram illustrates the behavior of the major activities composing the DMP.

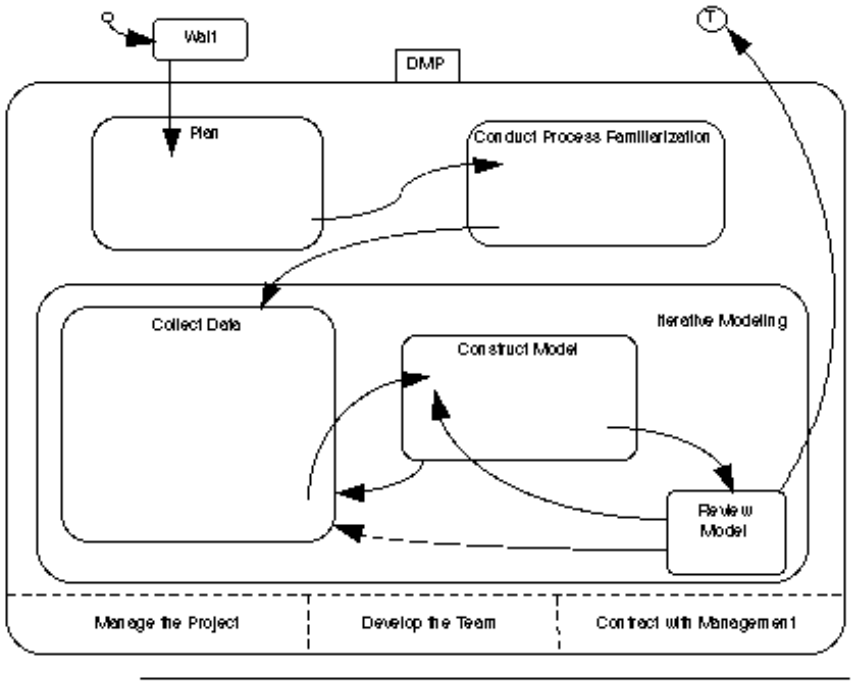

Figure 4: Sample Process Guide Page #2

The example process guide also contains introductory material describing how the document is organized, (including tables to aid the reader in locating specific types of information) and how to read and understand the diagrams (to help those who may not be familiar with the notations). Several appendices contain: guidance on tailoring the DMP to specific organizational circumstances; document-wide summaries of the diagrams, activities, artifacts, roles and agents; templates and examples of selected artifacts; and references.

4.2 Limitations of Paper Process Guides

Paper-based process guides pose certain problems in both usability and development. With respect to usability, paper-based process guides are hard to navigate and search, difficult to keep up-to-date, nearly impossible to customize based upon a user's specific needs, and often hard to structure in order to present different but related information together (e.g., an activity and an example of one of its output artifacts) unless the guide is taken apart and the pages manually reorganized. Moreover, developing paper process guides that are convenient to use for reference purposes is painstaking work. An investigation of standards and process handbooks (e.g. IEEE Std 1074, Cleanroom, military handbooks and regulations) uncovered many problems in existing documents and illustrated the benefits of using formal process modeling languages to develop process documentation [Kellner 93, Verlage 97]. But even when using formal approaches to process definition, there are still serious challenges when developing paper process guides. These include: explicit definition of a process requires considerable care and detail; ease of use leads to repetition of information wherever it is likely to be needed; and the linear structure of paper is quite limiting.

Most of these problems can be alleviated through the creative and careful application of on-line technologies such as used for the World-wide Web (WWW). Our approach to doing this is discussed in the remainder of this paper.

5 Electronic Process Guides

The preceding observations about paper process guides have led the authors to investigate the development and delivery of electronic process guides (EPGs) based on Web technology. Our current efforts are discussed in this section.

It should be noted that within the past year or so many organizations have begun to provide on-line access to their process documentation through an intranet. In most cases this results in one of the following rather straightforward situations:

- Documents are available to be downloaded in a form such as PDF, Microsoft Word format, or Adobe FrameMaker format; generally the Web pages simply offer a list of documents available for downloading.

- Documents have been converted from their word-processor format into HTML for direct display via Web browsers, but without any hyper-links.

- A more extensive conversion has been done so that cross-references within the document (e.g., references to other sections) become hyper-links one can use to navigate within the document. A table of contents may also be presented using hyper-links. However, no links are available across documents nor is the information within a document chunked or organized in any way other than as a flowing narrative.

A few notable cases which go beyond this level of Web-based on-line process guides are discussed in Section 8.

5.1 Basic Requirements and Design Principles

We have used the following as a set of basic requirements for a first version of a Web-based, on-line, process guide:

- An EPG should provide the basic information units described in the enhanced schema (Section 3.4), i.e., activities, artifacts, roles, agents and resources, as well as the major relationships between them, i.e., product flow, role assignment and decomposition. Similarly, an EPG should include all of the most critical, highest-priority information referred to in Section 3.2. Naturally, then, it will also follow the conceptual framework for software processes (Section 3.1) as a foundation.

- An EPG should provide all the information elements and interrelationships contained in a good paper process guide (such as the exemplary guide discussed in Section 4). In other words, information must not be lost when moving from a paper-based to a web-based version. Information should be "chunked" into small, easily digestible units as in a good paper-based guide. Moreover, an EPG should capitalize on diagrams, tables and narrative in a way which provides an effective user interface.

- An EPG should make extensive and effective use of hyper-links to support flexible navigation through the information contained in the guide. For example, intra- and inter-document references should have associated hyper-links, so that following the link moves the user directly to the corresponding information. Diagrams and other graphics should be image-mapped to provide additional information navigation pathways. Links should provide direct access to pertinent information such as examples and templates.

- It should be easy to access desired information via an EPG. Users should be able access frequently used information very quickly. It should be possible to get information on request, and with 'random-access', e.g., if information concerning an activity is needed, it should be possible to access this information directly without having to navigate through other activities or navigate the complete decomposition hierarchy.

- To facilitate orientation and usage, the web pages (windows) should all have the same basic structure, or at least be similar. Additionally, the windows should be structured (e.g., into frames or tables) in a way that is already familiar to the users or that they may easily learn by analogy with things they already know.

- The user should not be overwhelmed with too many overlapping windows. Therefore, the number of windows should be limited and well-managed. The user should have direct control over the opening, closing, sizing, positioning and viewing of windows.

- Current commercial off-the-shelf (COTS) technology for WWW applications should be used to implement EPGs. This will provide a familiar interface and functionality for the user. New window managers, browsers, plug-ins, etc., should be implemented only if absolutely necessary since this would diminish the ability to introduce the EPG smoothly and without large technology shifts.

5.2 Current Status of Prototyping

Our initial prototype is based on the exemplary process guide discussed in Section 4. The focus of the initial EPG is on supporting performer reference before or during activity performance (as opposed, for example, to reference during training).

The current prototype version is based on HTML frames and pages. Thus they can be maintained and updated centrally and process participants can access them by their normal browsers, providing an immediately familiar appearance and basic functionality for users. Using standard HTML and JavaScript has the additional benefit that a lot of useful functionality (e.g., bookmarks) is already available to the user.

The major information categories in the initial EPG are activities, artifacts and roles. Full descriptions of instances of these process elements are displayed in so-called *main pages*. In order to facilitate user orientation, main pages for activities, artifacts and roles all have the same basic structure (which is implemented using frames). A distinctive graphic icon and background color help the user readily distinguish among the three page types. Allowing pages (windows) for different types of process elements to be open simultaneously permits users to simultaneously view related information, such as a description of an activity alongside a description of one of its output artifacts. By default, the user may view only one page of each process element type at a time, i.e., all activities appear in the same activity window, all artifacts in the same artifact window, and all roles in the same role window. This reduces window juggling. (However, multiple windows can be difficult to manage and this aspect of an EPG requires additional work once we have gained experience and feedback.) However, if a user wants a new window for a second instance of some element type (for example, to view two artifact pages together), the option of opening a new window when following a link is already supported by most browsers.

Figure 5 shows a main page for an activity. It consists of four frames:

- The graphical hierarchical view (top right frame) shows the position of this activity within the decomposition tree; the name of the selected activity is automatically highlighted in red. To facilitate navigation within the hierarchy, the user may follow a link from any activity in the decomposition tree to the main page for that activity.

- The overview frame (on the left) identifies the process element by type (with an activity icon and the word *Activity*), by name (*Build Model*), and with a short description. It also identifies the information chunks associated with an activity, such as objectives, product flow, entry and exit criteria, and performers. These include the major information elements and relationships as-

sociated with an activity as described in Sections 3 and 4. The keywords (e.g., *Objectives*, *Tasks*, *Inputs*, *Outputs*) are linked to the corresponding sections in the description frame, e.g., a user interested in seeing the output products may click the underlined keyword *Outputs* to scroll down in the description frame to the list of output artifacts (as shown in the center of the Figure). Thus, like a table of contents, the overview frame provides easy access into various parts of the information displayed in the description frame.

- The description frame (center right) is where full, detailed information about the activity is shown. This includes a narrative description of the activity's objectives, a discussion of the tasks to be performed, lists of input artifacts and output artifacts, lists of entry and exit criteria, etc., as can be seen in the Figure. While the overview frame displays the names of the information categories, the description frame contains the 'contents' of these categories.

- Selecting links in the description frame brings up new information in the glossary frame (at the bottom). This provides fast access to summary information about related process elements within the same window. For example, the user can view short descriptions of artifacts or roles referenced in the description section. If the user needs more detailed information about an activity, artifact or role, he or she may click on its name in the glossary frame to open the main page for that process element.

To give a sense for the preservation of format across main pages, Figure 6 shows an artifact main page. Its structure is essentially the same as that of an activity main page. The only notable difference is that the top right frame shows a list of artifacts, whereas the activity frame showed the decomposition graphically. When there is only a single-level decomposition for the instances of some type of process element, the decomposition is shown as a list. Navigation within an artifact main page and to other pages works in exactly the same fashion as described for activity main pages. The initial EPG prototype also provides hyper-links from an artifact main page to templates and examples of the artifact. Among other things, this provides an EPG user fast access to clarifying examples and the latest versions of templates.

Role main pages provide information about the roles involved in the process. These pages are structured and can be used in the same ways as activity and artifact main pages. Included in a role main page is a list of activities the role is involved in, as well as information about any special skills required for the role. Items in the list of activities are linked to the pages for those activities, allowing a person filling this role easy access to just those portions of the process in which he/she participates. This is a first step towards full support for role-specific views. (In this initial EPG we have chosen to support just roles rather than both roles and agents. This is in keeping with our initial focus on providing

guidance during process performance. Agent information and the making of associations between agents and roles are important in other contexts, project management for example, which we will treat in future work.)

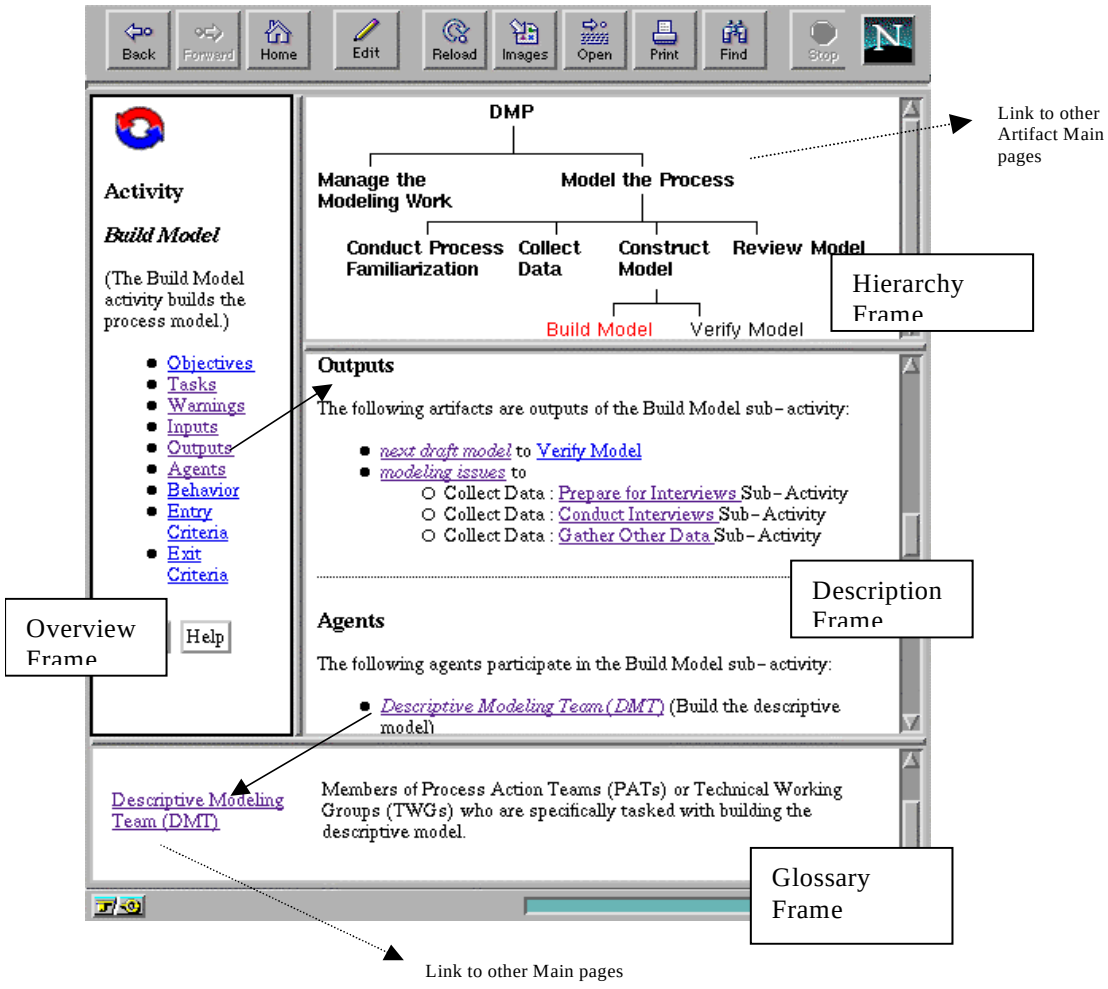

Figure 5: Activity Main Page

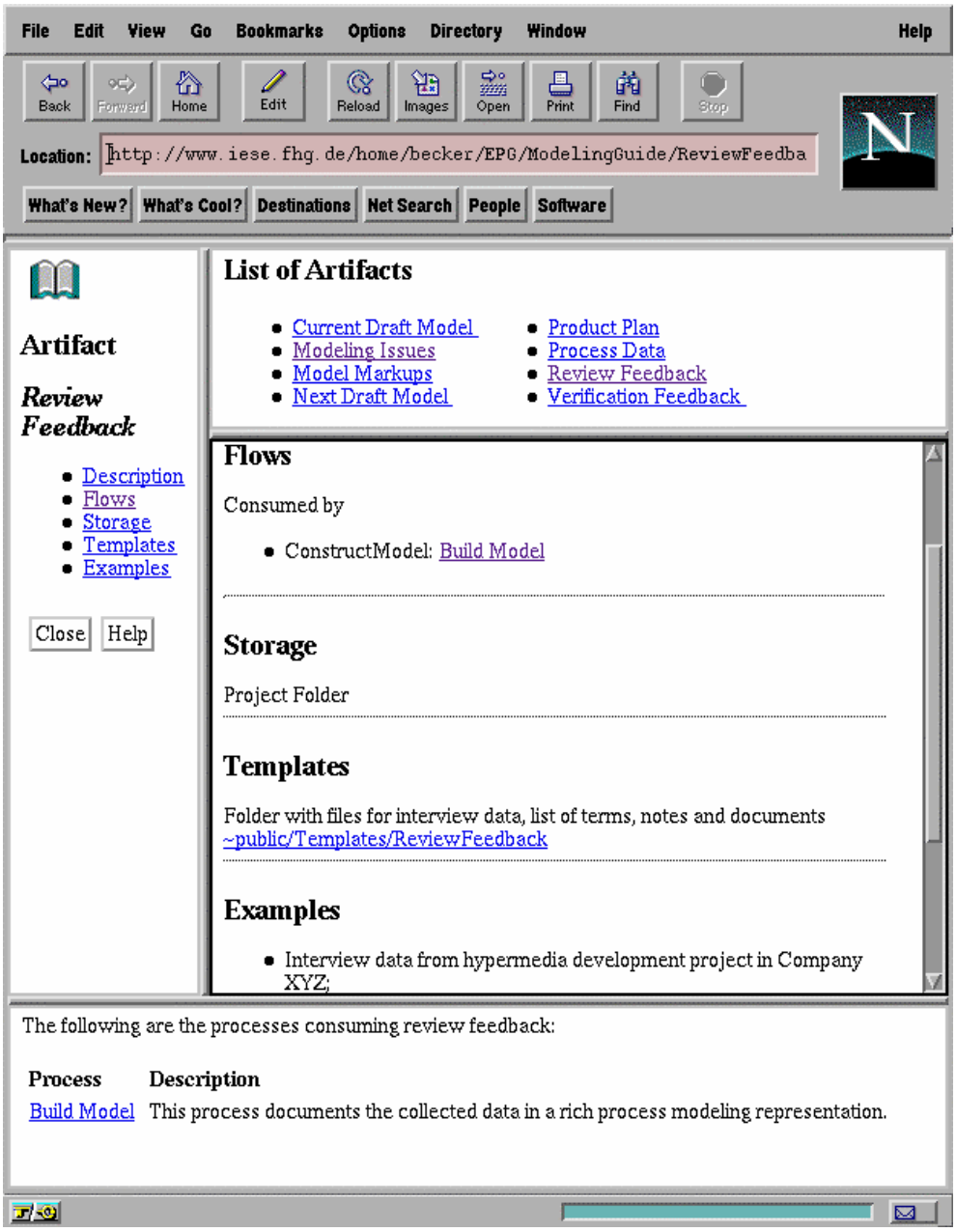

Figure 6: Artifact Main Page

The two diagram types included in the paper-based exemplary process guide described in Section 4 are also included in the EPG. They provide functional and behavioral perspectives into the process. The diagrams assist in visualizing the process and certain relationships among its elements. To support navigation through the information in an EPG, the diagrams are "active" (i.e., they are represented by image-mapped graphics). Thus, clicking on an activity box in a

functional diagram will open the main page for that activity, and clicking on a data flow line (or the adjacent artifact name) will open the main page for that artifact.

In addition to the main pages described above, project overview and help pages are also available. Among other things, the overview page offers direct access to every activity, artifact and role in the EPG. This allows a user to jump into the guide at any point, for example to review an activity that is to be performed, to read about an artifact to be produced and access its template and examples, or perhaps to find all activities performed by the user based on the role that user has just been assigned to. The help pages summarize the EPG and offer tips on its efficient use, describe the layout and operation of each of the page types, describe how to read and understand the diagrams, etc. The help pages demonstrate one way in which multimedia presentations can be beneficially applied within an EPG. For example, help on diagrams is provided not only through a fairly traditional combination of text and graphics, but also as a movie that highlights relevant elements of the diagram in synch with the audio narration.

In order to gain experience with a wide range of different users and their behavior, we have applied some of our initial ideas in the implementation of an EPG for the V-Model process, a German national standard software development process [V-Model-EPG]. Creating this EPG has allowed us to confirm the suitability of our information and navigation structures. Its use will allow us to get feedback from a relatively large user community.

6 Next Steps

The current EPG prototype focuses on user interface issues (e.g., web page design, information presentation, manipulation and navigation). This is a logical first step when moving away from purely paper-based process guides. Our plans for the future are discussed in this and the following section. In this section, we discuss the next two EPG prototypes which will first address better process guide management and then address personalizing an EPG. In the following section, we present our long-term vision of the capabilities needed for truly effective process guidance.

6.1 Better Management of Process Guides

In developing the initial EPG prototype, we have found it time-consuming to directly encode the pages in HTML, even with the use of a WYSIWYG editor. Changes to the layout of the pages have likewise required considerable effort. Similarly for changes to a process guide's content. It is critical that it be extremely easy to change an EPG's format or content because updates to a process guide are frequently required for correction, clarification, elaboration, process improvement, and so forth. Consequently, the following capabilities will be prototyped in the near future:

- EPG pages will be generated (probably dynamically) from an object-oriented database. The schema implemented in the Spearmint process modeling environment [Becker 97a] will provide the basis for the object-base. An initial, preliminary generation mechanism has already been implemented.

- Users should be able to search the process guide. In addition to simple keyword matching we also plan more extensive services, similar to those implemented in the process modeling environment MVP-E [Becker 97b]. Thus, the user will be able to make arbitrary queries against the EPG to extract the information he/she is interested in. For example, a user would be able to ask for all activities using a selected artifact, or to display a list of all activities which are not decomposed. The implementation of these services will likely use the Spearmint object-base, with the retrieved information being transformed dynamically into an EPG web page.

- Configuration management services will be provided for process guides. Process definitions are expected to evolve over time, so version control is clearly needed. Multiple versions of a process guide may be in use at any given time because projects already underway may choose to continue using

an older version that they started with. In addition, process engineers need to be able to study the evolution of a process and its guide. It is also valuable to be able to inform process participants about changes by displaying modifications to a process guide and presenting a list of recent changes. Furthermore, multiple variants of a guide must be supported simultaneously because processes often need to be tailored to the needs of specific projects, products and individuals.

6.2 Personalizing an EPG

The next major step will be to allow the customization and instantiation of process information for individual and team use. Each participant, or group of participants assuming one role (e.g., a test team), should be able to maintain a personal copy of a centrally-available guide as well as attach information about the state of their particular performance of the process. In particular, the following capabilities will be explored:

- Process participants should be able to annotate the guides. Individuals should be able to add their personal tips, comments and remarks to any part of the EPG. These might be entered and displayed in a special window. These annotations should always be accessible to their author, and might also be shared across their group, project team, or other organizational unit. Process engineers should also be able to use these annotations to help assess the current process, or its guide, in order to improve it.

- Copies of a given version of an EPG should be tailorable to account for project and product specifics, individual or team needs, and so forth. Earlier process technology work has identified a need for relatively generic process models which can be tailored to specific contexts [Heineman 94, Paulk 93, Verlage 97]. These variants may be developed by eliminating parts of the generic model, adding more concrete details, etc.

- Checklists should be added to allow users to keep track of progress. These checklists should be available for items such as task steps, entry and exit criteria, and artifact states. An EPG would then provide a simple agenda mechanism for each user, allowing him or her to more easily identify the steps to perform.

- Private work spaces should be associated with process instances. Here individuals and teams can share and save artifacts (and other information) created and used during the process. The process instance serves as a context within which to collect artifacts (and other files of information) that are related to each other not in a structural way but with respect to their content and process association.

- Role-specific views should be supported. Prior work in process modeling [Rombach 95] has suggested that role-specific views make process models especially pertinent and understandable to individual participants. EPGs should provide a mechanism to allow users performing a given role to focus on those activities that involve that role. However, more investigation is needed into which mechanisms are most desirable for these views. For example, an EPG might provide access to only those activities (and the associated artifacts) which are pertinent to a role, or these might be highlighted with all the other information in the EPG also readily accessible.

- EPGs should support resumption of work after an interruption. Interrupts are common during process performance. People wandering by, breaks, telephone calls, and meetings are common reasons for these interrupts. To help process performers cope with interruptions, an EPG can support tracking which EPG web pages are open, so at the very least the display of information can be later recreated. In addition, an EPG could be a basis for tracking information about the current state of the process so that, after an interrupt, a process participant could return to the appropriate state and resume work.

- Questionnaires should be attached to process guides to collect optional data from process participants. The collected data could be analyzed by quantitative methods and the results displayed in order to help guide process performance (e.g., by comparing the current fault detection rate against those of previous development efforts). To protect against misuse, these data should be encrypted and kept strictly personal.

7 Vision

This section describes our long-term vision for process guides. We term this desired form of guidance Collaborative Process Guides (CPGs). Our vision for CPGs has two aspects:

- Our vision for the usage of CPGs is that they provide a process participant with the guidance, and access to the training, needed to most effectively, efficiently and accurately perform the activities in which that individual plays some role.

- Our vision for the evolution of CPGs is that they evolve (i.e., are initially developed and then refined, elaborated, tailored and improved) efficiently, flexibly and collaboratively.

With respect to usage, a CPG will make extensive reference guidance readily available, and this guidance can be (at the user's discretion) within the context of the current state of the process being performed; for example, the CPG could identify and prioritize relevant options based on the current circumstances in the process. Moreover, our vision is that CPGs offer collaborative guidance when needed. For example, when an unusual or unanticipated situation arises (one that is not covered in the existing guidance), the CPG should provide collaborative access to process experts and/or other process participants for their advice and assistance. Similarly, if conflict situations arise (e.g., between different processes that have not been adequately coordinated) the CPG should provide a basis for an effective conflict resolution meeting involving the affected process participants and, as necessary, people with the authority to resolve the conflict. One final part of our usage vision is that CPGs provide context-dependent, convenient, on-demand access to initial training, just-in-time training and refresher training.

Some additional usage-related capabilities envisioned for CPGs are:

- Support should be provided for bringing the process "back on track" when actual performance varies from what was expected. This has been found to be a practical issue in using process technology. Relevant process state information might have to be adapted to the new situation to keep the process "going." "What if" exploration should be available – perhaps through process simulation – to assist in evaluating alternative means of bringing the process "back on track."

- CPGs should be integrated with the concepts of an experience factory [Basili 95]. The personal annotations mentioned above are only a first step in this direction. The discussions, resolutions and resulting outcomes from dealing with unusual or unanticipated situations, conflicts, etc., could be recorded and made available for searching. A case base (containing information about the process used, artifacts produced, measures and commentary) could help people find recent process instances which are similar to a situation that they face. Consulting an experience base would provide valuable guidance based on past experiences and lessons not yet directly incorporated into the process guide.

- More active support should also be made available to process participants. For example, having active agents locate relevant information (e.g., in an experience base or, more broadly, over an intranet or the Internet) and point the user to it may be better than waiting for a search initiated by the user. Anticipatory guidance based on likely next steps could be another part of this capability. Finally, managing a list of process steps ordered by priority rules specified by the user is another example of active support that could be offered via a CPG.

These usage-related capabilities would provide highly beneficial support for organizational learning and rapid dissemination of best practices.

Regarding evolution of CPGs, the term "evolution" is meant to be construed quite broadly, and includes initial development of a CPG as well as subsequent refinement, elaboration, tailoring, improvement, etc. CPG evolution should be very flexible, in the sense that (1) it should be robust across a range of definition rigor and thoroughness, (2) different portions of a single CPG can include substantially different amounts of detail, and (3) a CPG can describe behaviors ranging from highly disciplined to relatively unconstrained. CPG evolution should itself be a collaborative process, involving the process participants in substantial ways and not restricted to only process engineering experts. On-the-fly evolution should also be supported, so that changes can be (carefully) made to the CPG even while the corresponding process is being enacted, and other participants will be automatically alerted to relevant changes. Support should be provided for comparing and evaluating process versions and variants. Both qualitative and quantitative analyses should be supported, and "what if" simulations should be possible to help answer questions about hypothetical process performance. Finally, good evolution paradigms and effective tool support should allow CPG evolution to be performed efficiently.

In order to fully realize this vision for CPGs, we expect they will be a component of an integrated process support system. In addition to a CPG, such a support system would integrate:

- process enactment technology (e.g., process automation, process-sensitive software engineering environments, workflow engines) for well-structured, relatively routine, aspects of the process,

- collaboration technology for those aspects of the process where group work is important and information sharing, coordination and communication are vital,

- computer-based training, distance learning and similar facilities which can be used by process participants to acquire knowledge and skills specifically required for the process at hand, and

- process engineering technology (e.g., process modeling, analysis and simulation) supporting development, evaluation and improvement of the process and its representations.

A comprehensive support system of this nature (and including a CPG) would facilitate learning to perform the process, performing the process, and specifying, designing and evolving the process.

In addition to the envisioned CPG capabilities and characteristics described above, this sort of support system would offer automation of routine process steps and support for collaborative parts of the process. On a meta-level, it would also support comprehensive process engineering activities [Kellner 96] (many of which are performed on process models), including those mentioned above in the discussion of CPG evolution.

Regarding collaboration technology, it is noteworthy that this system would support collaboration in three broad areas:

- performing certain aspects of the process, where information sharing, coordination and communication are vital,

- determining appropriate guidance in unusual or unanticipated situations, cases of conflict, etc., and

- engineering suitable processes and their representations.

Although the full vision for CPGs would be realized through an integrated process support system, a CPG would be extremely useful to process participants on a stand-alone basis. Many of the envisioned capabilities and characteristics for a CPG can be meaningfully developed, explored and delivered without fully integrating all of the technologies called for in the complete process support system. In fact, this is the initial development approach being taken. This relatively stand-alone approach is more manageable technologically.

Our incremental approach to exploring EPG and CPG capabilities can provide a strategy for introducing software process technology into an organization. Most transition approaches pursued within the software process technology community tend to require large-grained, revolutionary steps. In contrast, our development approach suggests an evolutionary, incremental approach to technology introduction which has the advantage of being tailorable and extensible. As people get used to one level of support they will likely become interested in, desire, and be able to adequately handle more sophisticated features of the technology.

Therefore, we are convinced that a good strategy for the widespread introduction of process technology into an organization is to begin with process guidance. It is clear that a good understanding of the process by participants is a prerequisite to effective use of technologies such as workflow and CSCW, making guidance a natural starting point. Beginning with guidance would allow the evolutionary, incremental approach to technology introduction; worries about the system overly controlling people's work would not arise; nor would participants necessarily be faced with a change in the technology they use to do their work (as would be the case with a first application of CSCW or workflow). In fact, because guidance can be provided independently from the environment in which participants do their work, participants can grow comfortable with the technology at their own pace. Once this initial level of process technology introduction is achieved, the additional technologies discussed above (e.g., process enactment, collaboration, process engineering) can be gradually incorporated – leading to smooth introduction of the complete, integrated support system described above.

8 Relationship to other work

The research and development community has given much attention to the development of software process technology. However, there has been only limited attention to process guidance issues. The task of disseminating process information to process participants has been of little interest, except to practitioners. Nevertheless, some examples do exist of using computer-based systems for browsing and presenting process-related information to process participants.

- Some process modeling capabilities have included meta-definition capabilities (for an example, see [MetaCase]). These capabilities allow one to define editors for graphical process modeling notations. They also typically allow the generation of code from a model. These capabilities will be useful in achieving our vision of being able to generate skeletal EPGs and CPGs from process models.

- In recent years, several firms have developed process asset libraries (PALs) [Kellner 93]. One of the most extensive has been developed by Litton-PRC, Inc. [Hollenbach 97]. It contains more than a thousand assets, including process descriptions (with many versions and variants from different projects), example artifacts, artifact templates, and so forth. The assets are organized and indexed to facilitate access and are available throughout the firm via their intranet. However, related assets (e.g., a process instance and an artifact instance which it produced) may not be directly related via hyperlinks, diagrammatic models are not extensively used, and multimedia is not used to present information.

- The Irish firm aimware, Inc. [aimware] offers a product – aimfirst® – based on Lotus Notes®, which (among many other things) helps organize and access process documentation. The "processware" component of aimfirst deals with roles, policies, processes, procedures / guidelines, work products / templates, and life cycles. For each of these categories, a template is provided which suggests a good set of information elements to be recorded. The categories and information elements for each were heavily influenced by earlier SEI work on information content and the CMM. The descriptions are primarily accessed via indices, although the tool employs hyper-links to connect some elements (e.g., a link from an activity to a role is supported, although the reverse is not; procedures link to work products and processes). It also associates roles with the individuals "holding" each role. Annotations are supported, but they are shared among all users, and are associated with the description of an entire entity (e.g., process, procedure) not individual in-

formation elements within those descriptions. Only one description is visible at a time, and the use of graphics is quite minimal.

- The V-Model Browser [V-Model-Browser] provides access to a German national software development standard in widespread use within governmental and industrial software development projects. The browser uses a hyperlinked, electronic version of the standard to provide access to activity, artifact and role descriptions. The browser also provides search capabilities. However, only one web page is visible at a time, so it is not possible to simultaneously see an overview of that page, navigation information and related information (e.g., an activity description and descriptions of one of its outputs). Also it offers no diagrams or other graphics. The browser does not include many of the features we envision for future EPGs and CPGs. It does not distinguish between different users or user groups. There is no ability to extend, tailor or annotate process descriptions. In addition, the V-Model Browser does not provide the ability to store or interpret process status information.

- Process-sensitive software engineering environments [Christie 95] provide contextual information about the processes being enacted. For example, in Process Weaver [Process Weaver] a process participant can see a list (called an *agenda*) of steps he or she is responsible for performing. The agenda indicates whether a step may be performed (i.e., all of its pre-conditions are satisfied). When a process participant selects a performable step, he or she receives a *work context* containing a description of what should be done and providing access to relevant artifacts and resources. The system does not provide an overview of the full process (i.e., a work breakdown structure), so process performance largely proceeds by having participants react to process state changes with little to no insight into the overall process. As with many process-sensitive environments, the design of Process Weaver focuses on coordinating people's work as it is described in process models. The design does not take guidance into account in any substantial ways.

Other communities have also investigated several of the problems we are addressing in our work:

- The information systems "method engineering" community has addressed (separately and concurrently) many of the same issues as have been addressed by the software process community. Method engineering work has investigated notations for representing methods (akin to process modeling discussed in Section 2.1), meta-models for methods (akin to the schema work described in Section 3.4), method bases, computer aided method engineering, tailoring of methods, reusable method fragments (including repositories and composition), and so forth. See, for example, [Harmsen94] for details.

- PALs are an instance of the more general notion of a corporate "memory." Another instance of this notion is a corporate knowledgebase. This is usually a repository of historical data useful for decision-making. Intranets are sometimes used to facilitate access, and models akin to the process models discussed here are sometimes used to assist in organizing the data (for example, see [Kloosters 97]). In comparison to the EPGs and CPGs discussed here, corporate knowledgebases are focused on the decision-making needs of project managers rather than the process understanding needs of process participants.

- To successfully achieve our vision, we must make process and collaboration technology work together. These two technologies have been developed by two relatively independent communities – the process technology community and the CSCW community – and getting the technologies to work together is not easy. In other parts of our work at the SEI [for more information, see SCP], we are developing the infrastructural capabilities needed to select and integrate process and collaboration technology.

9 Summary and Conclusions

Effectively disseminating process knowledge to process participants is crucial in real-world software engineering settings. Process participants need effective guidance when process conformance is important, when a process changes frequently (e.g., as the result of a process improvement program), when new personnel join a project, when a process is complex and long-lived, and so forth. Based on the authors' experiences with process modeling and definition in numerous practical organizational settings, and assessments of the current status of process documentation in companies and governmental organizations, we have concluded that most existing software process documentation is deficient in both form and content. These documents are generally missing important information and are difficult to use, understand and access; most practitioners agree and consequently rarely use the existing documentation. There is a huge potential for improving these guidance-oriented documents – termed process guides – and this would be of real value in practical settings.

This paper has described advances in both form and content for process guides. This work has been motivated and informed by experience gained in a number of industrial and governmental situations. Prior related work on information content for process guides was reviewed in Section 3, culminating in a well-defined schema for representing all relevant process information. Section 4 summarized work on paper-based process guides, which has exemplified a new form for these documents providing needed information content and enhancing their readability and usability. It was observed, however, that paper-based process guides still suffer a number of limitations inherent in paper media. This motivates the need to develop process guides designed specifically for the Web.

The later half of this paper is devoted to Web-based process guides. Although numerous organizations are putting their process documentation on the Web, in almost all cases this means merely making their old documents (designed for paper) easier to access via the Web. Unfortunately, they generally remain deficient in both form and content. Our work focuses on making the information that is really needed available in a form truly designed to take advantage of Web technology.

The early phases of our work entail prototyping what we term Electronic Process Guides (EPGs). Requirements for these prototypes are based on our prior work and real-world experience noted above. The initial prototyping concentrates on the basic information structures to be provided to process participants, the user interface, and services primarily related to browsing complex process information. Section 5 described and illustrated the current prototype

EPG. Section 6 then presented the features and capabilities that we plan to add in the next two major increments. Finally, Section 7 laid out a concrete vision for this technology – which we term Collaborative Process Guides (CPGs) – including its integration with other process technology, collaboration technology, and more.

However, before more sophisticated features are introduced a usability analysis must be performed. Arrangements are currently being made with multiple organizations to test out the EPG concepts and technology on some of their real processes. One such prototype has already been developed for the V-Model process. In these pilot tests, process participants will give feedback on the prototypes in order to enable us to adjust the EPG concepts and technology to best serve real needs. Our goal is to develop and refine the "technology" for EPGs and subsequently for CPGs. This will be accomplished through incremental prototyping, including guides to be used in actual practice along with evaluation of their real-world effectiveness.

Our incremental development of the EPG concepts and technology suggests a strategy for introducing it into real settings. Based on our experience, it is much easier to introduce complex process technology step by step, instead of changing work contexts dramatically by initially inserting process-sensitive software engineering environments. First, an EPG could replace paper-based process guides and handbooks. Later on, when people get used to this style of accessing process information, more sophisticated services could be introduced. This strategy would also allow organizations to pick only those services which are needed in their specific situations. Following this strategy, stable releases and evaluation steps are required not only before starting the next increment, but also for introduction of intermediate results into an organization.

The process guide technology, examples and prototypes described in this paper are designed to meet a real and important need: to provide effective guidance for process participants. In a broader sense, this work helps identify what is actually needed for process participants and when process technology should be used in real-world settings. Along these lines, we feel that research in process technology should pay much more attention to process guidance and presenting process information to process participants than it has heretofore.

Acknowledgements

The authors express their deep appreciation to Arne Könnecker and Sascha Schwarz who implemented large parts of the current prototype. Thanks are also extended to Linda Gates, Ken Goncharoff, Dirk Hamann, Ralf Kempkens, Peter Rösch and Jörg Zettel who provided valuable feedback on developmental versions of the EPG. We also thank Linda Gates, Stephan Greene, Ken Goncharoff and the ICSP reviewers for their helpful comments and suggestions.

10 Bibliography

[aimware] http://www.aimware.com

[Armitage 94a] J. W. Armitage and M. I. Kellner. "A Conceptual Schema for Process Definitions and Models," 153-165. Proc. 3rd Intern. Conf. on the Software Process, Reston, Va., USA, October 10-11, 1994. IEEE Computer Soc. Press, 1994.

[Armitage 94b] J. W. Armitage et al. Software Process Definition Guide: Content of Enactable Software Process Representations (CMU/SEI-94-SR-21) Pittsburgh: PA: Software Engineering Inst., Carnegie Mellon Univ., December 1994.

[Basili 95] V. R. Basili. "The Experience Factory and its Relationship to Other Quality Approaches," 65-82. *Advances in Computers*, 41 (Marvin V. Zelkowitz, ed.). Boston, MA: Academic Press, 1995.

[Becker 97a] U. Becker-Kornstaedt and R. Webby. "Towards a Logical Schema Integrating Software Process Modeling and Software Measurement," 84-88. *Proc. 19th Intern. Conf. on Software Engineering Workshop: Process Modeling and Empirical Studies of Software Evaluation* (Rachel Harrison, ed.). Boston, USA, May 1997.

[Becker 97b] U. Becker-Kornstaedt, D. Hamann, J. Münch, and M. Verlage. "MVP-E: A Process Modeling Environment." *IEEE TCSE Software Process Newsletter,* 10 (Fall 1997): 10–15.

[Christie 95] A. M. Christie. *Software Process Automation: The Technology and Its Adoption.* Berlin: Springer Verlag, 1995.

[Christie 97] A. M. Christie et al. Software Process Automation: Interviews, Survey and Workshop Results. (CMU/SEI-97-TR-008) Pittsburgh: PA: Software Engineering Inst., Carnegie Mellon Univ., October 1997.

[Curtis 92] B. Curtis, M. Kellner and J. Over. "Process Modeling." *Comm. of the ACM* 35, 9 (September 1992): 75-90.

[Finkelstein 94] A. Finkelstein, J. Kramer and B. Nuseibeth. *Software Process Modeling and Technology*. New York: Wiley, 1994.

[Fuggetta 96] A. Fuggetta and A. Wolf (eds.). *Software Process*. England: Wiley, 1996.

[Gates 97] L. Gates, K. Goncharoff and M. Kellner. *An Example Proc-*

ess Guide: Process Guide for a Descriptive Modeling Process (CMU/SEI-97-HB-004). Pittsburgh, PA: Software Engineering Inst., Carnegie Mellon Univ., 1997.

[Harmsen94] F. Harmsen, S. Brinkkemper and H Oei. "Situational Method Engineering for Information System Project Approaches." Methods and Associated Tools for the Information Systems Life Cycle (A-55) (A. A. Verrijn and T. W. Olle, eds.). Holland: Elsiver Science B.V. (1994): 169-194.

[Hollenbach 97] C. Hollenbach, R. Young, A. Pflugrad and D. Smith. "Combining Quality and Software Improvement." *Comm. of the ACM* 40, 6 (June 1997): 41-45.

[Heineman 94] G. Heineman, J. Botsford, G. Caldiera, G. Kaiser, M. Kellner and N. Madhavji. "Emerging Technologies that Support a Software Process Life Cycle." *IBM Systems J.*, 3, 3 (1994): 501-529.

[i-Logix] http://www.ilogix.com

[InfoMap] http://www.infomap.com

[Kellner 89a] M. Kellner. "Experience with Enactable Software Process Models," 85-88. *Proc. 5th Intern. Software Process Workshop: Experience* with *Software Process Models*. Kennebunkport, Maine, October 10-13, 1989. IEEE Computer Soc. Press, 1990.

[Kellner 89b] M. Kellner. "Software Process Modeling: Value and Experience," 23-54. *SEI Technical Review*. Pittsburgh, PA: Software Engineering Inst., Carnegie Mellon Univ., 1989.

[Kellner 90] M. Kellner and H. D. Rombach. "Session Summary: Comparisons of Software Process Descriptions," 7-18. *Proc. 6th Intern. Software Process Workshop: Support for the Software Process.* Hakodate, Hokkaido, Japan, October 28-31, 1990. IEEE Computer Soc. Press, 1991.

[Kellner 91] M. Kellner. "Software Process Modeling Support for Management Planning and Control," 8-28. *Proc. First Intern. Conf. on the Software Process*. Redondo Beach, Calif., October 21-22, 1991. IEEE Computer Soc. Press, 1991.

[Kellner 93] M. Kellner and R. Phillips. "Practical Technology for Process Assets," 107-112. *Proc. 8th Intern. Software Process Workshop: State of the Practice in Process Technology*. Wadern, Germany, March 2-5, 1993. IEEE Computer Soc. Press, 1993.

[Kellner 96] M. Kellner, L. Briand and J. Over. "A Method for Designing,

Defining and Evolving Software Processes," 37-48. *Proc. 4th Intern. Conf. on the Software Process: Improvement and Practice*. Brighton, England, UK, December 2-6, 1996. IEEE Computer Soc. Press, 1996.

[Kloosters 97] M. Kloosters, S. Brinkkemper, F. Harmsen and G. Wijers. "Intranet Facilitated Knowledge Management: A Theory and Tool for Defining Situational Models," *Proc. Intern. Conf. On Advanced Systems Engineering (CAiSE97)*. Barcelona, Spain, June 16-20, 1997. Springer Verlag, 1997.

[Madhavji 94] N. Madhavji, D. Höltje, W. Hong and T. Bruckhaus. "Elicit: A Method for Eliciting Process Models," 111-122. *Proc. 3rd Intern. Conf. on the Software Process*. Reston, Virginia, USA, October 10-11, 1994. IEEE Computer Soc. Press, 1994.

[McFeeley 96] R. McFeeley. *IDEAL: A User's Guide for Software Process Improvement* (CMU/SEI-96-HB-001). Pittsburgh, Pa.: Software Engineering Inst., Carnegie Mellon Univ., 1996.

[MetaCase] http://www.jsp.fi/metacase

[Morisio 95] M. Morisio. "A Methodology to Measure the Software Process," *Proc. 7th Annual Oregon Workshop on Software Metrics*. Silver Falls, OR, 1995.

[Paulk 93] M. Paulk, B. Curtis, M. Chrissis and C. Weber. *Capability Maturity Model for Software, Version 1.1*. (CMU/SEI-93-TR-24, ADA 263403). Pittsburgh, Pa.: Software Engineering Inst., Carnegie Mellon Univ., February 1993.

[Process Weaver] http://webflow.cginn.cgs.fr:4747/PW

[Raffo 98] D. Raffo and M. Kellner. "Modeling Software Processes Quantitatively and Evaluating the Performance of Process Alternatives." Book chapter in *Elements of Software Process Assessment and Improvement*, (K. El Emam and N. Madhavji, eds.). IEEE Computer Soc. Press, forthcoming, 1998.

[Rombach 95] H. D. Rombach and M. Verlage. "Directions in Software Process Research." *Advances in Computers,* Volume 41 (M. V. Zelkowitz, ed.). Boston, MA: Academic Press (1995): 1-63

[SCP] http://www.sei.cmu.edu/scp/scp_welcome.html

[SPC 96] *Improving the Software Process Through Process Definition and Modeling*. Software Productivity Consortium. Intern. Thomson Computer Press, 1996.

[V-Model-Browser] http://www.scope.gmd.de/vmodel/en

[V-Model-EPG] http://www.iese.fhg.de/VModell

[Verlage 97] M. Verlage and J. Münch. "Formalizing Software Engineering Standards," 196-206. *Proc. 3rd Intern. Symposium and Forum on Software Engineering Standards (ISESS '97).* Walnut Creek, California, USA, June 1997.

[Webby 98] R. Webby, P. Rösch and M. Verlage "Spearmint – A Prototype Tool for Visualizing Complex Software Processes" *Proc. World Conf. on Integrated Design & Process Technology (IDPT98)*, Berlin, Germany, July 1998.

[Winograd 86] T. Winograd. "Breaking the Complexity Barrier (again)," *Interactive Programming Environments* (D. R. Barstow, H. E. Shrobe and E. Sandewall). McGraw-Hill, 1986.

[Yu 94] E. S. K. Yu and J. Mylopoulos. "Understanding 'Why' in Software Process Modeling Analysis and Design," 159-168. *Proc. 16th Intern. Conf. on Software Engineering.* Sorrento, Italy, May 1994.

Document Information

Title:	Process Guides: Effective Guidance for Process Participants
Date:	August 4, 1998
Report:	IESE-012.98/E
Status:	Final
Distribution:	Public