In *Proceedings of the 5th International Conference on Enterprise Information Systems*, pages 203–208, Angers, France, April 2003

LIVE-REPRESENTATION PROCESS MANAGEMENT

Daniel D. Corkill
Department of Computer Science
University of Massachusetts
Amherst, MA 01003-4610
corkill@cs.umass.edu

Zachary B. Rubinstein
Department of Computer Science
University of New Hampshire
Durham, NH 03824-3591
zack@cs.unh.edu

Susan E. Lander
Independent Consultant
Amherst, MA 01002
susanlander@attbi.com

Victor R. Lesser
Department of Computer Science
University of Massachusetts
Amherst, MA 01003-4610
lesser@cs.umass.edu

Key words: Decision-support systems, resource allocation and scheduling, business-process representation and execution.

Abstract: We present the *live-representation* approach for managing and working in complex, dynamic business processes. In this approach, aspects of business-process modeling, project planning, project management, resource scheduling, and process automation, execution, and reporting are integrated into an on-line representation of planned and executing processes. This representation provides a real-time view of past, present, and anticipated process activities and resourcing. Changes in the process are immediately reflected in the live representation, so that, at any point in time, the latest information about process status and downstream expectations is available. Managers can directly manipulate the live representation to change process structure and execution. These changes are then quickly propagated throughout the environment, keeping managers and process participants in sync with process changes.

In this paper, we describe the technical and humanistic issues associated with the live-representation approach and summarize experiences gained in developing a commercial implementation that was used with design processes in the automotive and aerospace industries.

Economic pressures are requiring businesses and organizations to work faster and with fewer resources. Managing complex business processes has always been difficult. However, the increased interaction among processes (through shared resources and results) and increased concurrency of activities within processes (to complete processes sooner) now leave process managers[1] with less time to make decisions and expand the implications and scope of those decisions significantly.

Dynamic business processes present substantial problems for today's process managers who must routinely investigate and evaluate process status, handle exceptions and resource problems, worry about potential downstream issues, and change the process structure and details accordingly. For example, key resources may need to be reassigned or may become unavailable, information and results may not arrive when expected, tasks may take more (or less) time than expected, the results may be surprising and suggest other activities, and so on. Due to process dynamics and uncertainty, *these events cannot be anticipated nor can contingencies be established for them beforehand*. Therefore, a significant contributor to the effectiveness of business-process execution is the ability of managers to make appropriate and timely process assessment, guidance, and adjustment decisions during process execution.

In this paper, we describe a novel knowledge-based decision support approach that was applied to automotive and aerospace design processes. Automotive and aerospace design involves dynamic processes that use limited and highly expensive physical and personnel resources and require the careful coordination of diverse organizational units. Although these design processes are being scrutinized, standardized, and documented, they are highly dynamic by nature and require intelligent, timely, and flexible management to meet the competitive pressures to produce better designs in shorter time using fewer resources.

1 Needed: Better Decision Support

A lack of accurate, accessible, and timely information and analysis remains at the root of the management problem. Off-line reporting and analysis tools do not operate at the time scales and level of detail required to identify and address process problems until after they have occurred—if they are identified at all. Reports are commonplace of analysts working on design candidates that had long been eliminated, of design shortcuts taken to meet deadlines that are

[1] We use "process managers" to denote individuals who make management decisions associated with the execution of dynamic business processes. Such individuals need not have an explicit management role in their organization.

no longer relevant, and of unforeseen delays while awaiting critical test facilities or personnel resources. Without knowing what is happening and why *and* what is likely to happen in the future, process managers are forced to rely on intuition and luck in making decisions.

Detecting and responding to process dynamics is onerous for managers. Managers need help with:

- keeping everyone that is involved in a process informed of their anticipated tasks and estimated start and due times
- automatically adapting schedules when the execution of tasks differs from expectations
- redesigning and rescheduling processes as new information becomes available
- alerting managers to potential problems, such as the unavailability of resources, that will cause delays
- notifying managers of scheduled times that violate deadlines

What is needed is an integration of aspects of business-process modeling, project planning, project management, resource scheduling, and process automation, execution, and reporting into a "live" decision-support environment that presents managers and participants with an accurate, on-line view of process status and downstream expectations.

2 Live-Representation Approach

Information technologies have addressed pieces of the process-management problem, but they have not attacked it head on. For example, software technologies have been developed that assist process participants in performing individual tasks, both individually and collaboratively. Many formerly manual tasks have been automated, and slow and expensive exchanges between tasks have been eliminated by using integration technologies. Software tools have also been developed to assist process managers in planning, executing, monitoring, measuring, and documenting process and organizational activities but, despite efforts to make these many technologies interoperate, these efforts have not significantly improved the way that dynamic processes are managed.

In the live-representation approach, processes are not designed on paper, modeled and simulated off line, and eventually analyzed and reported on long after the work is done. Instead, process design and management activities are performed in conjunction with process execution, all utilizing an on-line, executing representation of processes, plans, participants, and resources.

The live-representation approach addresses the reality of dynamic process management by facilitating continual proactive process guidance and adjustment. The approach presents a significant advantage over reactive methods that limit effective interventions. In essence, the live-representation approach provides a knowledge-intensive, proactive, highly responsive, and flexible environment that supports dynamic-process management decisions. This environment provides an up-to-date view of process activities and the current best estimate of downstream activities and potential problems. It assists managers in coordinating multiple processes and shared resources and ensures responsive execution of management decisions. The objective is to provide the support needed to keep process-management activities ahead of the rate of change in the processes.

The live-representation approach also addresses the following goals:

- improve process coherence by keeping participants and managers informed of the current state and downstream expectations of processes
- enable proactive response to process and resource problems before they occur by providing a clear picture of what the processes will do in the future without intervention
- push appropriate information to participants when it is needed
- allow customization of individual activities based on process context
- support resource allocations among processes
- support cross-organizational collaboration processes via inter-organizational process linking and resource allocation strategies

Required Capabilities The the following five capabilities are required in the live-representation approach:

1. **Complete process representation**—The approach begins with a knowledge-intensive definition of the dynamic process, with sufficient detail to allow automated execution and to make reasonable expectations of downstream activities.
2. **Direct execution**—The process definition is instantiated and executed for each process so that the representation matches exactly what is happening as the process is executed. Direct execution is important for validation of the process representation and to ensure that on-the-fly modifications to the representation will be reflected in the executing process.
3. **Integrated downstream forecasting**—Dynamic scheduling of downstream activities and resources

are needed to allow time for proactive intervention. The scheduling must be tightly integrated with the direct execution of process representations and must balance the need for process flexibility with the need to maintain process stability whenever possible. It must be able to make timing and resourcing decisions across multiple processes using criteria provided by process designers and execution managers.

4. **Live presentation of execution status, history, and expectations**—The latest details of process state and downstream expectations must be presented. The presentations should be tailored for understandability, relate directly to the process representations, and quickly focus attention on problem areas.
5. **On-the-fly process modification**—Managers must be able to change the process structure and execution details of executing processes in response to unanticipated situations and problems.

In concert, these live-representation capabilities allow managers to quickly comprehend current process status and potential downstream problems, make proactive interventions, and have those interventions immediately reflected in the executing processes.

3 Issues and Implementation

KPM (Lander et al., 1999) is a substantial commercial implementation of the live-representation approach that was targeted to automotive and aerospace design processes. In developing KPM, we had to address a number of issues in the following categories:

- **Representing process definitions**—How to represent the process, data, and execution knowledge needed to forecast, execute, and present process status and downstream expectations.
- **Generating and changing process plans**—How to generate an initial process plan for a process and how and when to change to other process plans due to new information or because of dynamic edits performed by managers on executing processes.
- **Changing resource availability**—How to detect and handle unexpected gains and losses in resource availability.
- **Updating the execution model**—What to do when tasks either miss their scheduled start and end times or have opportunities to execute at times that are better than their scheduled times.
- **Supporting human interaction**—How to capture and support the activities that managers and participants perform during the execution of a process.
- **Operating in an enterprise setting**—How to provide a flexible software configuration that can be adapted to diverse organizational settings, support concurrent process development and management, perform efficient distributed scheduling and execution, and resume executing processes that are interrupted due to hardware or software failure.

We next discuss these issues in greater detail.

Process Representation The live-representation approach places a diverse set of requirements on process representation. First, the representation must be fully executable. Everything that needs to be known for process execution must be represented: software, resources, data, interfaces, etc. Second, the representation must include the knowledge needed to perform downstream scheduling. This knowledge includes contextual expectations of task durations and resource requirements, likely conditional and looping choices, and so on. Third, the representation must be expressive and intuitive to process managers and participants. The representation should enable rapid understanding of what is happening and will happen and support both abstract and detailed presentation. Fourth, the representation must be amenable to in-process structural modification, including moving entire process subtrees.

A variety of process representations have been developed, including the workflow standard (Workflow Management Coalition, 1999), IDEF3 (Mayer et al., 1995), PIF (Lee et al., 1996), TÆMS (Decker, 1996), PSL (Knutilla, 1998), and LITTLE-JIL (Lerner et al., 1998). While each of these representations captures a variety of constraints and semantics of processes, they do not capture the control knowledge necessary for predictive, downstream scheduling. For example, to schedule processes with loops and conditionals requires estimating the number of iterations a loop may take and the specific branches to be taken.

Although workflow management systems track the progress of process execution and facilitate the passing of data among tasks and the invocation of automated tasks (Cichocki et al., 1998), they do not schedule downstream activities and resource allocations (Corkill, 2000). The recent BPML specification for describing business processes within Web services (Arkin, 2002), has been augmented with richer control semantics, but it is designed for execution and modeling, not dynamic scheduling. Lastly, these representations are not designed to quickly present process status and information to non-technical users—an important criteria in the live-representation approach.

We represent process definitions as a hierarchy of tasks, where the constraints among child tasks are determined by the task type of their parent task (Figure 1). There are two categories of task types: container and non-container. Container tasks have control semantics that are used when generating the process plan for a process and are not explicitly part of the process plan. The container task types are: task structure, serial, parallel, branching, parallel-branching, looping, and quantified. Non-container tasks have no children and become part of the process plan. The non-container task types are: executable, user assistant, task valet, placeholder, component, and return. Non-container tasks can be thought of as the detailed activities in a process, while container tasks form the control structure in which the activities are embedded. A more detailed discussion of these task types and process representations is presented in (Rubinstein, 2002).

Each task in a process definition includes behavioral knowledge based on the task type. For container tasks, the behavioral knowledge controls the structure of the generated process plan. For example, when a process plan is generated for a parallel-branching task during initial scheduling, a set of scheduling-time predicate functions corresponding to the task's children is evaluated to determine which children should be expanded and included in the process plan. Only information that is available in advance of execution can be used in these schedule-time predicates. When the parallel-branching task executes, a separate set of execution-time predicate functions corresponding to the task's children is evaluated to determine which children are actually executed. Non-container tasks include behavioral knowledge that determines the expected duration of the task and the possible sets of resources that can accomplish that task.

The process definition is atemporal. It does not have separate objects for tasks that might be executed more than once, such as the tasks within a looping task. When the definition is first scheduled for execution, an explicit process plan with objects for all scheduled tasks is generated from the process definition.

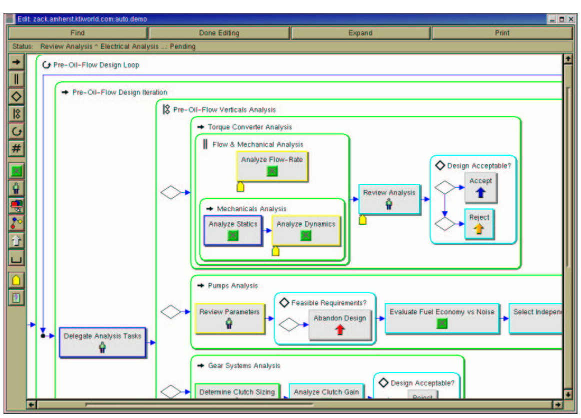

Figure 1: An Executing Process

A process definition represents a family of potential process plans. By traversing through the definition and evaluating the controls for the expected behavior of container tasks, one of the members of the family is instantiated. When the execution behavior differs from this expected structure, such as a loop iterating more than expected, the current process plan is modified, changing the process plan from one instantiated member of the family to another. The process definition represents many related process plans, facilitating the likely switching of process plans during execution and rescheduling. By having alternate process plans represented in a single process definition, dynamics with specific implications, such as unexpected branches or extra iterations being executed, can be immediately incorporated into the executing process plan. The representation captures not only what is expected but what is known to be possible within the current process definition. Through manager interventions, the representation can be updated with any new knowledge of the process, including changing the structure of the process definition itself (and the corresponding family of potential process plans).

Scheduling and Execution Fundamental to the live-representation approach is maintaining the best estimate of how the execution of that process will unfold. Major components of that estimate are a schedule of what activities are expected to be executed, the resources that are assigned for them, and their expected start and finish times. The schedule for a dynamic process is a highly fluid entity that constantly changes in response to process execution, new information, and a changing environment.

To generate and maintain this type of schedule, a new kind of distributed scheduler that is tightly integrated with process execution was needed. The scheduler needed to detect and react quickly to execution-time dynamics, to support distributed scheduling of concurrent processes with shared resources, and to allow modifications to be made to the executing process.

The PROTEUS constraint-based scheduler developed for KPM (Rubinstein, 2002), is the first instance of this new type of scheduler. In PROTEUS, we provide heuristics to evaluate and trigger appropriate responses to execution-time events, balancing the trade-offs among efficient resource allocation, schedule volatility, and rescheduling costs. PROTEUS provides mechanisms for automatically manipulating the process plan in response to events such as the execution of control constructs and structural edits performed by a manager. Shared resources for overlapping executions of processes require that these heuristics have an enterprise-wide perspective; i.e., they must assess the effect that individual process dynamics have on all the other executing processes that interact or share resources with that process.

The interplay between process execution and scheduling is a unique aspect of the live-representation approach. Although there has been significant work done on rescheduling in response to execution dynamics—especially in the area of constraint-based schedulers, such as ISIS (Fox, 1987; Fox and Smith, 1984), MICRO-BOSS (Sadeh, 1991), OPIS (Smith, 1989), and OZONE (Smith et al., 1996)—this work has not addressed dynamics that require changes to the *structure* of the executing processes.

There has also been scheduling work on selecting the best process plan given the current state of execution. Whether a Markov Decision Process is used, as in RTDP/ROUT (Schneider et al., 1998), or the aggregation of statistical distributions throughout the process plan, as in design-to-criteria scheduling (Wagner et al., 1998), this work assumes that the process definition includes all the possible paths that may be taken during execution. This closed-world assumption makes it possible to train the scheduler over a number of process executions.

With the live-representation approach, however, the closed-world assumption cannot be made. Many decisions that affect the process structure cannot be made until execution of the process definition is underway. Furthermore, unexpected results or events, may lead to changes to the process plan that were never anticipated. Since every execution may be radically different, each new execution is effectively a first-time execution that should be executed "first time best." Performing well on average is not good enough.

Humanistic Issues The increased process visibility provided by the live-representation approach can be a concern for managers and process participants. If handled incorrectly, the explicit capture of process activities could be seen as a "big brother" intrusion into what was formerly private workday behavior.

Such humanistic issues were strongly considered in the development of KPM. For example, a system that told professional design engineers and analysts exactly what detailed activities they should do and when they should perform those activities would not be well received. We addressed this issue by representing process participants as complex *multiplexing* resources that are able to choose when to perform activities within bounded assignment windows. The KPM scheduler ensures that there is sufficient time for participants to perform their activities, but only a macro schedule of due dates and estimated start times are provided to participants.

Information hiding can also ease perceived intrusiveness. For example, it is easy to have KPM capture the actual time a participant starts work on an activity (within the assignment window). The start times could be used to better estimate the actual times spent on activities and to warn participants of the need to begin an activity if it is to be completed on time. While appropriate in some organizations, the culture in others might prefer that task assignments are simply acknowledged, leaving responsibility for completing the task by the due date and the details of when the work is done to the participants.

Schedule volatility due to aggressive optimization is also an issue. Participants would be uncomfortable with a system that repeatedly changed near-term task assignments due to rescheduling. So, in KPM the scheduler attempts to limit the volatility of personnel assignments by incorporating the cultural "cost" of reassignment in its scheduling heuristics.

Automated resource allocation and scheduling is itself a potential cultural issue. Process managers are used to making detailed resource-allocation decisions, but that practice leads to delays and inefficiencies associated with manual and localized resourcing decisions. In the live-representation approach, the managerial emphasis shifts from making individual resourcing decisions to providing the system with the criteria to allow it to make detailed resourcing decisions in real time.

4 Acceptance and Evaluation

Quantifying the benefits of using the live-representation approach is difficult. We have found that dynamic business processes have not been fully represented, much less measured. Without such baselines, evaluation of the approach can only be made in terms of more intangible benefits, such as improved responsiveness, resource sharing, process visibility, proactive handling of process problems, and facilitation of managerial innovation, rather than on detailed measurements of resource downsizing and reduced process durations.

Nevertheless, the response to KPM by major automotive and aerospace manufacturers has been enthusiastic, with users obtaining the following benefits:

- **Improved process understanding**—The approach ensures that the process that is represented is the process that is executed, and any mismatches in the way processes are represented and the way they actually perform are quickly discovered. The biggest challenge is eliciting predictive process knowledge associated with processes.
- **Improved process visibility**—The instantaneous view of process status makes it easy to see how dynamic processes are behaving. In particular, the ability for managers to graphically request task-progress updates from participants, and for par-

ticipants to provide unsolicited status updates, has proven to be very useful.

- **Advance notice of problems**—Dynamic scheduling provides early warning of downstream resource-scheduling and timing problems. This capability is particularly helpful in managing resources that are shared among multiple dynamic processes. Warnings are presented explicitly to managers in the graphical representation of the executing process.
- **Graphical managerial intervention**—Manipulating executing processes graphically allows managers to enact process modifications quickly. Changes stemming from these modifications are directly communicated to all process participants effected by them.
- **Capture of execution history**—Details of process execution are captured by the live-representation environment. This information can be used for post-execution analysis, process-definition improvements, and for meeting regulatory auditing requirements.

Given this initial response, we believe that the live-representation approach shows great promise for improving how we manage and work in complex, dynamic business processes.

Acknowledgment This work was supported, in part, by the National Science Foundation (DMI-0122173) and by internal research & development funding from Blackboard Technology and from Knowledge Technologies International. The views and conclusions presented are those of the authors and should not be interpreted as necessarily representing the official policies or endorsement of the supporting organizations.

We also thank Kevin Gallagher, Ken Berthiume, and Suzanne Tromara for their contributions in developing the live-representation approach and KPM.

REFERENCES

Arkin, A. (2002). Business Process Modeling Language, Version 1.0. Technical report, Business Process Management Initiative (BPMI.org).

Cichocki, A., Helal, A., Rusinkiewicz, M., and Woelk, D. (1998). *Workflow and Process Automation: Concepts and Technology*. Kluwer.

Corkill, D. D. (2000). When workflow doesn't work: Issues in managing dynamic processes. In *Design Project Support using Process Models Workshop, Sixth International Conference on Artificial Intelligence in Design*, pages 1–13, Worcester, Massachusetts.

Decker, K. (1996). TÆMS: A framework for environment centered analysis & design of coordination mechanisms. In O'Hare, G. and Jennings, N., editors, *Foundations of Distributed Artificial Intelligence*, chapter 16, pages 429–448. Wiley Inter-Science.

Fox, M. S. (1987). *Constraint-Directed Search: A Case Study of Job Shop Scheduling*. Research Notes in Artificial Intelligence. Pitman Publishing, London.

Fox, M. S. and Smith, S. F. (1984). ISIS—A knowledge-based system for factory scheduling. *Expert Systems*, 1(1):25–49.

Knutilla, A. (1998). Process specification language: Analysis of existing representations. Technical Report NISTIR 6133, National Institute of Standards and technology, Gaithersburg, MD.

Lander, S., Corkill, D., and Rubinstein, Z. (1999). KPM: A tool for intelligent project management and execution. In *Workshop on Intelligent Workflow and Process Management: The New Frontier for AI in Business, Sixteenth International Joint Conference on Artificial Intelligence*, Stockholm, Sweden.

Lee, J., Gruninger, M., Jin, Y., Malone, T., Tate, A., Yost, G., and other members of the PIF Working Group (1996). The PIF process interchange format and framework. Technical Report 194, MIT Center for Coordination Science (CCS).

Lerner, B. S., Osterweil, L. J., Stanley M. Sutton, J., and Wise, A. (1998). Programming process coordination in Little-JIL. In *Proceedings of the 6th European Workshop on Software Process Technology*, pages 127–131, Weybridge, UK.

Mayer, R. J., Menzel, C. P., Painter, M. K., deWitte, P. S., Blinn, T., and Perakath, B. (1995). Information integration for concurrent engineering (IICE) IDEF3 process description capture method report. Technical report, Knowledge Based Systems, Incorporated, College Station, Texas.

Rubinstein, Z. B. (2002). *Efficient Scheduling of Evolving, Nondeterministic Process Plans in Dynamic Environments*. PhD thesis, University of Massachusetts, Amherst, MA.

Sadeh, N. (1991). *Look-Ahead Techniques for Micro-Opportunistic Job Shop Scheduling*. PhD thesis, Computer Science Department, Carnegie Mellon University.

Schneider, J., Boyan, J. A., and Moore, A. (1998). Value function based production scheduling. In *Machine Learning: Proceedings of the Fifteenth International Conference (ICML '98)*.

Smith, S., Lassila, O., and Becker, M. (1996). Configurable, mixed-initiative systems for planning and scheduling. In Tate, A., editor, *Advanced Planning Technology*, pages 235–241. AAAI Press, Menlo Park CA.

Smith, S. F. (1989). The OPIS framework for modeling manufacturing systems. Technical Report CMU-RI-TR-89-30, The Robotics Institute, Carnegie Mellon University.

Wagner, T., Garvey, A., and Lesser, V. (1998). Criteria-directed heuristic task scheduling. *International Journal of Approximate Reasoning*, Special Issue on Scheduling, 19(1):91–118.

Workflow Management Coalition (1999). *Workflow Management Coalition Terminology & Glossary*, volume WFMC-TC-1011, 3.0. Workflow Management Coalition, URL: http://www.aiim.org/wfmc/standards/docs/glossy3.pdf.