Automated Process Support for Organizational and Personal Processes

Kevin Gary, Tim Lindquist, Harry Koehnemann. Ly Sauer
Arizona State University
Computer Science Department
Mail Stop 5406
Tempe, AZ 85287-5406
yfppg@asu.edu

ABSTRACT

We propose two views on process: an organizational view and a personal process view. Information technology applies Automated Workflow technology to define, execute, and track an organization's automated business processes. Calendaring tools provide a form of personal process view through scheduled work items. However, the personal, or individual, view of the process space has largely been ignored. We maintain that as organizations become increasingly decentralized, a single organization's process space is becoming difficult to recognize. Individuals of the organization are asked to do work that spans organizational, functional, and even geographic boundaries. An integrated view of organizational workflows and personal processes is needed to address these new demands. In this paper we argue for the need to integrate organizational and personal processes. We then propose a component-based process modeling approach and supporting process architecture that integrates these process spaces. Finally, we describe our recent efforts at developing Java prototype process tools that realize the proposed modeling technique and supporting architecture.

Keywords: Workflow, Personal Process, Components

1.0 Introduction

Recent changes in industry and technology are imposing more demanding technical requirements on information technology. In industry, organizations are downsizing and becoming increasingly decentralized, often causing projects to be managed across multiple organizations or functional units. Current technology is growing at a rate that can be difficult to track. Industry investments in desktop tools and groupware must be leveraged against growth in local and wide-area networks (LANs and WANs). In particular, the growth of the Internet makes it possible to envision computer support of global, decentralized, business processes.

These changes in industry and technology escalate the pressures put on information technology research. Providing computer support for widely distributed organizations using new technologies such as the Internet, Groupware, Calendar Management, and Automated Workflow is at least an IT systems analyst's headache. Determining the best way to integrate these tools to ensure maximum productivity is at best an IT manager's nightmare. In our view, the ability to define, execute, and track business processes is central to the ability to integrate these technologies in a widely distributed setting *and* make their use productive. Therefore in our research we focus on automated process support. In the business domain, automating business processes is known as Automated Workflow.

Workflow is the study of modeling and enacting business processes by human and computer agents. Automated Workflow adds an emphasis on applying current computer and information technology in a workflow environment, with the desire of automating parts of workflows, or supporting entire workflows.

Automated Workflow has traditionally focused on defining and automating business processes from the organization's standpoint. Little regard is given to managing overlapping workflows in an individual workspace, or for even considering the personal processes of an individual when considering the productivity of the organization. The current solution is to drop a set of personal productivity tools, such as calendaring tools, in the lap of the individual and let her/him work it out.

In order to achieve greater productivity from both workflow and personal productivity tools, a more integrated view of organizational and personal processes must be considered. An integrated view allows individuals the ability to develop their own productive work practices in support of an organization's processes, and allows for a more natural handling of processes spanning multiple organizations and individuals. We are in the beginning stages of our research into the utility of providing such an integrated view. In this paper we propose an open architecture for integrating organizational workflows and personal productivity processes. We motivate the need for an integrated approach, and present a component-based approach to process modeling that provides the interoperability required to achieve the integration. We also present a suite of tools being developed at Arizona State University that realize this architecture.

The rest of this paper is organized as follows. Section 2.0 discusses relevant issues in current workflow and calendaring technology. Section 3.0 argues for an integrated view of organizational and personal process spaces, presents a component-based approach to process modeling, and proposes a general process support architecture. Section 4.0 presents tool prototypes realizing this architecture that were recently developed at Arizona State University. We conclude in Section 5.0 with a summary and discuss future avenues for our research.

2.0 Background

Approaches to developing workflow systems have both commercial and academic origins. Commercial systems have evolved from work on forms-based image processing systems and groupware[14]. The line between workflow and other types of systems is often blurred, with groupware, scheduling, database, and email tools providing some workflow functionality. In addition, several commercial products that advertise workflow capabilities fall far short of providing full-fledged support for defining and enacting business processes. Academic research has focused mainly on process modeling and database transaction issues[9]. Process modeling research has led to the development of workflow representations based on a variety of formalisms. Database transaction research focuses on extending traditional transaction semantics to support long duration[2][9] and/or cooperative transaction[5][10] models. The result is a proliferation of approaches and issues relating to workflow.

Current efforts are attempting to get researchers and vendors to converge on a common foundation for workflow. The Workflow Management Coalition (WfMC) was formed in August 1993 to promote workflow technology. The WfMC has proposed a reference model[22] and a set of interfaces, called WAPIs[1] based on that model[24][25][26][27] as an attempt at standardizing architectural elements of workflow systems and the interactions between those elements. The Process Interchange Format (PIF) Working Group was formed to explore the potential to provide automatic translations between process representation formalisms[15]. Finally, Microsoft is pushing their Messaging API (MAPI) as a defacto standard for implementing workflow systems. Microsoft has recently teamed with Wang to develop the MAPI-WF specification[17], an extension of MAPI for supporting workflow-specific services.

The WfMC is presently the most significant of the efforts attempting to standardize workflow systems. The WfMC Reference Model (Figure 1) identifies the basic architectural components of a workflow environment. At the center of the model is a Workflow Enactment Service (WES), comprised of one or more Workflow Engines. A WES provides services through the WAPIs to workflow-related tools. These include Process Definition Tools for defining processes, Workflow Client Applications for handling user requests for work, Third-party Applications that need to communicate data and operations to the WES, other WESs for providing interoperability between enactment services, and Administration and Monitoring Tools for data gathering for process improvement activities.

The WfMC Reference Model identifies common workflow system components and interfaces. The WAPI interface specifications define a set of low-level protocols for synchronously and asynchronously exchanging workflow data between the tools and the WES. Our basic problem with this approach is that these protocols are too low-level; they imply a restrictive workflow model. Workflow representations that cannot easily convert their process data to conform with this underlying model cannot obtain conformance with the model. This is one of the issues our research addresses.

Recent standardization efforts also address the area of calendaring protocols. One popular calendaring protocol (adopted by Netscape's Calendar Server[18]) is the vCalendar protocol[12]. In the vCalendar protocol, calendaring and scheduling entities, called *events*, are transported between applications that can understand the protocol. This approach is similar to the effort of the WfMC protocols in that it defines a low-level data interchange format that tools must understand to conform to the protocol. Other, more industry-wide standardization efforts are being sponsored by the Internet Engineering Task Force (IETF) based in part on the vCalendar specification. The IETF has recently sponsored the development of three separate calendaring protocols, the Calendaring Interoperability Protocol (CIP), the Core Object Specification (COS), and the Internet Calendar

1. For Workflow API and Interchange

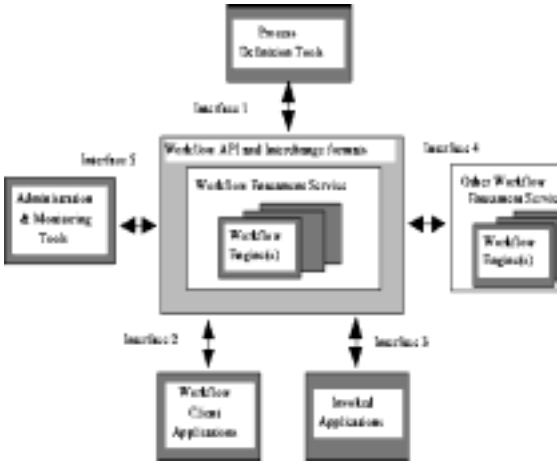

FIGURE 1. WfMC Reference Model ([23])

Access Protocol (ICAP). These protocols specify interface and other requirements on calendaring systems exchanging calendaring data.

The standardization efforts in both workflow and calendaring focus on low-level data interchange and protocols for exchanging such data in a client-server environment. While this is a widely accepted standardization approach, we fear that a stable data format is difficult to obtain due to the maturing of the underlying models in each domain. This is especially true in workflow. In the calendaring domain, a problematic issue is that calendaring formats and tools support only rudimentary dependencies between tasks. These issues are compounded when integrating workflow and calendaring systems. Workflow systems can write events to calendar tools, but are not aware of the personal views of the process of the participating individuals. Likewise, calendaring systems provide a personalized view of work, but do not possess sophisticated enough models to negotiate with workflow systems over the ability to do assigned work.

3.0 Integrated Process Support

We advocate an integrated view of an organization's process space and the personal process spaces of its individual workers. In this view, the organization's workflows are integrated with individual personal process spaces. Section 3.1 discusses this idea in more detail. To support this integrated view, we advocate a component-based approach to process modeling that avoids a reliance on low-level data interchange formats. This approach is called Open Process Components, and is described in Section 3.2. Finally, we propose a generic architecture in Section 3.3 that derives from our integrated view of process. In Section 4.0 we present some Java prototype tools based on our ideas.

3.1 Organization vs. Personal Process Space

Automated Workflow is the specification and execution of a business process of an organization[9]. Workflows are modeled as a collection of process steps, or tasks, assigned to individuals taking on particular roles. Many modern workflow systems work in this way; the process is considered from a single organization's viewpoint. This viewpoint is illustrated in Figure 2.

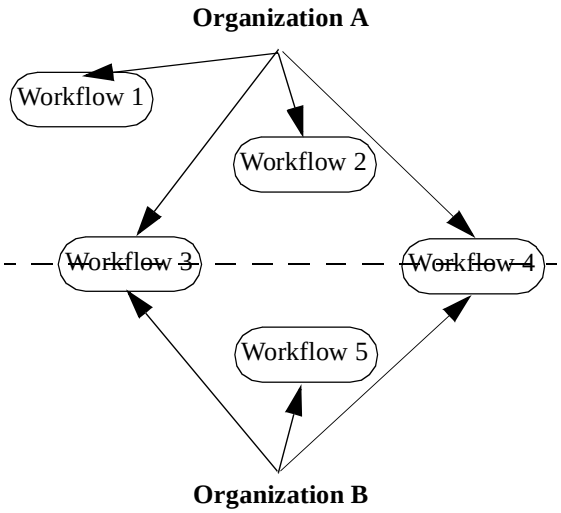

FIGURE 2. Organizational Process Perspective

Figure 2 shows the process space of two organizations, generically labeled A and B. These organizations share two workflows: Workflow 3 and Workflow 4. Interoperability of the underlying process models and process support architecture is required to allow these organizations to share these workflows.

The workflow systems we have experienced or seen in the literature take this organization-centered approach to automating business processes. For example, Action Workflow from Action Technologies[1] operates on a cyclical model where workflow units interoperate to produce customer satisfaction. Different participants are viewed as *customers*, *performers*, or *observers* at each workflow stage of the cycle. While Action Workflow provides client-side functionality to obtain task lists for individuals, it does not provide a structured way for individuals to define personal processes and integrate them into the scope of organizational processes. Another example is the application of groupware-oriented tools such as Lotus Notes to workflow[20]. Notes provides much of the needed infrastructure for managing data and transactions within a workflow. However, again there is no structured way to define personal processes and integrate them into organizational processes. Instead, the approach is again organization-centered, where workflows are defined at the organizational scope,

and personal tasks derived from the workflow model. Still other workflow platforms, such as InConcert[16], emphasize collaborative aspects of workflow execution. Collaborative work is closer in spirit to the idea of integrated process spaces, but differs in that the emphasis is on mechanisms supporting shared access to data. Users still act on tasks delegated to them by the organizational workflow model.

To keep pace with industry trends and technology impacts, this organization-centered viewpoint will have to change in at least the following ways:

- Interoperability between workflows developed across business functional units and/or organizations must be supported.
- The potential for wide-area distributed participation must be supported.
- Individuals must have the ability to define, execute, and track the personal processes they perform to be productive within the context of an organization's business processes and goals.

The work of the Workflow Management Coalition as well as research efforts such as our Open Process Components Framework (see Section 3.2) address the first two issues directly. However, there has not been a lot of consideration for the last issue. At best, current workflow systems notify individuals of new work items through email or custom client applications. Some even have the ability to write to personal calendaring software through interfaces such as Microsoft and Wang's MAPI-WF[17]. But the viewpoint still originates with the organizational process. An agent-centered viewpoint, showing the distribution of workflows an individual participates in, and the set of personal processes an individual employs, is not considered.

The need for supporting the personal process view is just beginning to be recognized in more dynamic process areas such as Software Engineering[11]. In the software process domain, the work of the software developer is considered dynamic in the sense that the developer must be creative in seeking the solutions to design, implementation, and maintenance dilemmas[4]. As workflow extends to more complex and skilled tasks, automated workflow systems will be required to encompass more than just the straightforward document-routing capabilities of image processing systems. Future demands will include the ability to support more of the skilled, or knowledge work, that people perform in the organization. In order to do this, workflow systems must relax the prescriptive constraints it places on performers of the workflow, and allow these workers to perform their own personal processes to carry out the work.

Figure 3 shows an agent-centered viewpoint of the process space. Jill is an agent working for Organization A, Bob works for Organization B. Jill participates in Organization A's workflow 1 and 3. Bob participates in Organization B's workflows 3 and 5. In order to accomplish tasks in workflow 1, Jill employs her Personal Process 1. Likewise, Bob employs his Personal Process 3 in carrying out tasks relevant to Workflow 5. In addition, Bob employs Personal Process 3 to carry out similar tasks in the shared Workflow 3. Jill does not have a relevant personal process defined for her assigned tasks in Workflow 3. Finally, each individual may have personal processes defined that are outside the scope of an explicit workflow for either organization. These may be processes defined solely by the individual's personal productivity initiative.

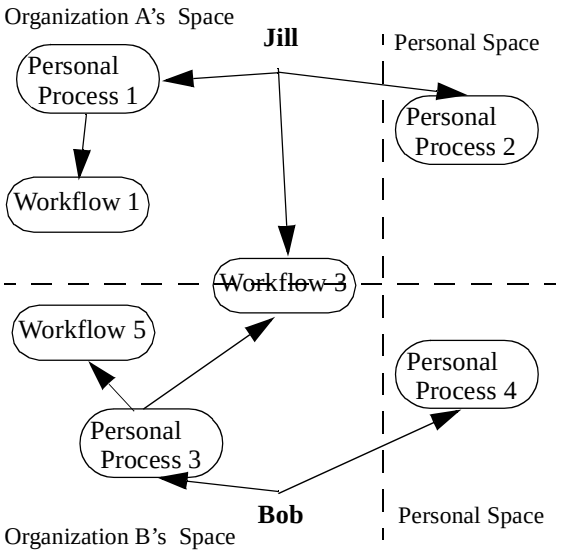

FIGURE 3. Personal Process Perspective

There are several reasons for arguing for an integrated view of organizational and personal processes. Figure 3 shows the overlap of the personal and organizational process space. Defining and executing business processes is motivated in part by the need to ensure business goals are achieved. Workflows are largely assumed to be static, repetitive processes that involve rote decision-making in support of well-defined business goals[9][1]. To expand the scope of processes automated workflow systems can support, more dynamic workflows that include personal processes should be considered. Another motivating reason comes from the diverse set of relationships in which both organizations and individuals participate. Individual workers, particularly at

1. We refer to Georgakopoulos, Hornick, and Sheth's[9] trade press characterization of *administrative* and *production* workflows. Our research is closer to *ad hoc* workflows, though our point is they can be better understood through an integrated view of the process space.

highly skilled levels, perform in a wide variety of diverse business functions. Downsizing and decentralization of organizations coupled with increasing outsourcing of work makes it unrealistic to take the single organization approach. The business processes of multiple organizations must be integrated with the personal processes of the participants.

In order to accomplish this integration, we propose a component-based approach to process modeling and an open architecture for supporting personal and organizational process spaces.

3.2 Component-based Process Modeling

Organizations developing standards in workflow and calendaring focus on low-level data interchange protocols to be applied in a client-server environment. The development of such protocols, particularly the protocols related to workflow definition interchange[1], are too restrictive to ensure widespread adoption. Instead, we propose an object-oriented component-based approach to process modeling and execution. In our research we are developing a component-based framework for process modeling called the *Open Process Components (OPC) Framework*. It is not the focus of this paper to delve into the details of the OPC Framework, but we do provide a brief discussion relevant to the process support architecture presented in Section 3.3. Further details may be found in [8].

There is a need for a unifying framework for representing and manipulating workflow abstractions. We take an object-oriented approach we call Open Process Components. Entities of the workflow domain are represented as objects, with manipulations of those objects defined as object behaviors. The approach is component-based, from the perspective that interfaces are well-defined so that components interact in meaningful ways. The OPC Framework provides a foundation for constructing component-based process models in an extendable fashion.

There are three important aspects to the OPC Framework that allow it to support component-based process modeling. The first is a meta-model that identifies basic process entities and relationships between entities. Basic process entities include *Process*, *Activity*, *Product*, *Role*, and *Agent*. A *Process* is a decomposable entity into subprocesses and subactivities. This allows development of process models in a top-down fashion. An *Activity* is an executable fragment of a process model; it represents a refinement of a portion of a process model down to an executable state. A *Product* is a work artifact that is either consumed as input by an Activity or produced as output. A *Role* is a process-specific definition of the skill set required to perform an Activity. A Role is process-specific as opposed to organization-specific, meaning management must decide how to map organizational roles to process-specific roles. This mapping is the relationship between Roles and *Agents*. The meta-model described briefly here is adopted from the PCIS LCPS meta-model[7]. However, the concepts are similar in a variety of general descriptions of process in the literature[5][9][15][22]. In the OPC Framework, this set of process entities and relationships form the basis for meaningful component interactions.

The second important aspect of the OPC Framework is a state-based encapsulation of execution interfaces. By this we mean each component in a process model possesses a process state, and this state is manipulable by a set of interfaces to the component that are available during various stages of executing the process model. Example interfaces include *start*, *suspend*, *resume*, *abort*, *completeWithFailure* and *completeWithSuccess*. Each component maintains an explicit, independent state during execution of the process model, and the state of process execution at any point in time is the combination of states of the components involved in the process.

The final salient feature of the OPC Framework is a three-tiered object-oriented class hierarchy for defining components. An object-oriented methodology provides several advantages: encapsulation of heterogeneous process representations, an economy of representation through inheritance, and the ability to specialize component definitions through subclassing. From a process modeling perspective, one major advantage of the hierarchy is its ability to be extended. New component definitions and abstractions can be added within the framework without modifying pre-existing definitions. A second important advantage is that specialized component definitions allow heterogeneous process modeling formalisms to interoperate with one another. For example, a Petri-net based process model fragment can interoperate with a process model fragment developed in a scripting language by encapsulating each as a component under the framework. This is especially beneficial in the organizational/personal context of processes we consider in this paper since it should not be assumed that homogeneous process models are generated across these contexts.

As a brief example, consider the ad hoc workflow depicted in Figure 4, taken from [9]. This workflow represents a paper review process. In a component-based process model, each task in the workflow is represented as an activity component. Interactions between the components is governed by the set of interfaces each component supports. The benefit is that the implementation of each component is separated from these interfaces. Different process modeling and

1. More specifically, the Workflow Process Definition Language proposed in WAPI 1[24].

enactment services can be used to define and execute the details of each task. This differs from existing systems where homogeneous models and supporting services are employed.

The workflow in Figure 4 is a relevant example of the utility of integrated organizational and personal process spaces. Consider for example the "*Review*" tasks in the workflow. These are assigned to separate persons fulfilling the role of *Reviewer*. However, there is not sufficient detail in this definition to automate the support of review activities for each reviewer. Furthermore, it is not appropriate to believe that this organizational workflow *should* provide such detail. Instead, it is more natural that each reviewer perform a personalized review process that meets the requirements of the organizational workflow. Therefore, if Jill and Bob were Reviewers in this workflow, each would carry out the review according to her/his own personal process for reviewing papers, employing familiar tools and methods for producing the needed results.

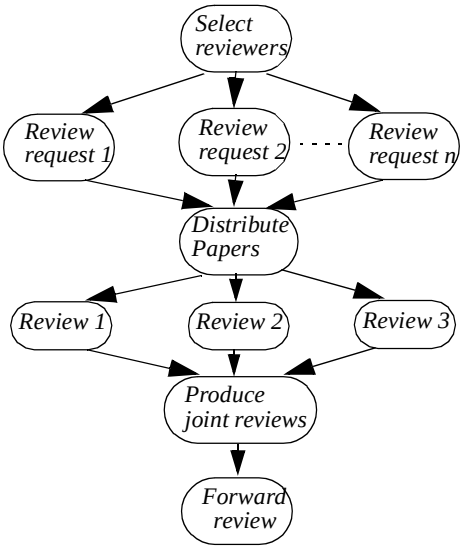

FIGURE 4. Example Workflow
(taken from [9])

Component-based process modeling is at the heart of our research and relevant to the topics discussed in the rest of this paper. However, the elements of organizational versus personal process spaces and process architecture we discuss do not necessarily rely on a component-based approach. One can readily envision modifications to existing tools such as Action Workflow or Lotus Notes discussed earlier that would address process space integration. We encourage the reader to consider process modeling approaches and process space integration issues as independently as possible.

3.3 A Process Support Architecture

To support the integration of the organizational and personal process spaces, we propose an architecture that extends traditional workflow client-server architectures to include support for the personal process space. Figure 6 shows the proposed general architecture.

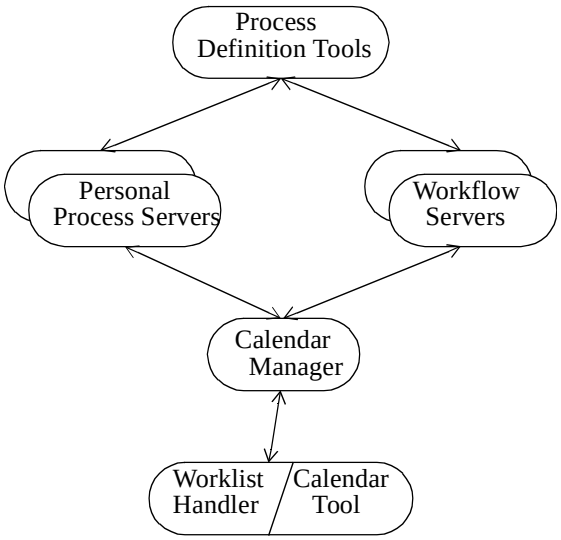

FIGURE 6. Process Support Architecture

The architecture in Figure 6 integrates organizations' workflow servers and personal process servers with calendaring technology to produce a time-oriented view of work for the end-user. Arcs indicate the bidirectional flow of components over the architecture. This architecture extends traditional workflow architectures, such as the Workflow Management Coalition's Reference Model[22], to include the end user's personal process space. The components of this architecture are:

- *Process Definition Tools*

  Process Definition Tools are used to create component-based process models. These tools may query Personal Process and Workflow servers in order to reuse existing process component definitions.

- *Workflow Servers*

  One or more servers create the organizational process space(s). These servers manage component-based workflow models created for organizational units by Process Definition Tools.

- *Personal Process Servers*

  Similar to a Workflow Server, a Personal Process Server manages process definitions for individuals, created from components by Process Definition Tools.

- *Calendar Manager*

  The Calendar Manager is the organizer of an individual's process space. The Calendar Manager manages instances of process models from the individual's perspective.

- *Worklist Handler/Calendar Tool*

  This is a client-side tool that presents the individual with her/his work to do. This may be in the form of a task list, or may be a time-oriented view depending on process constraints and personal scheduling preferences.

This general architecture clearly shows the separation and integration of organizational and personal process spaces. The distinct servers manage personal and organization processes. This distinction is a logical one; in practice a single implemented server may include the functionality to manage both process spaces. Integration of the spaces comes from the Process Definition Tools and the Calendar Manager. A Process Definition Tool creates component-based process models. By accessing the process definitions on both servers, the tool is able to create and reuse organizational process that utilizes process specifications of relevant individuals. The Calendar Manager integrates instances of organizational and personal processes from the individual's perspective. The Calendar Manager has the ability to accept or decline work requests from process servers, or manage changes to the individual's process space when forced to do so. This tool is the focal point of the individual's process space. Finally, the Worklist Handler/Calendar Tool is a combination of a workflow client and a personal calendaring tool. This client-side tool has the ability to host process components and support the enactment of such components in order to carry out the actual work.

The architecture we propose is an integration of current workflow architectures such as the WfMC's Reference Model[22] and calendaring environments such as Netscape's Calendar Server[18]. However, current architectures do not take such an integrated view. We know of no tool that allows for process models to be created that integrate a workflow model and a personal process model. The proposed process definition tool allows for this integration. We know of no environment that provides a componentized personal view of process like the proposed Calendar Manager. One can envision workflow servers writing to an individual's calendar through an interface such as Microsoft and Wang's MAPI-WF interface[17]. However, this requires that the workflow server have explicit knowledge and access rights to individuals' calendars. The proposed Calendar Manager explicitly manages an individual's workspace, negotiating between servers and individual preferences to present the personal process space to the end user. The existence of such a tool enables a component-based architecture that does not require Personal Process and Workflow Servers to communicate directly to negotiate over rights to assign work to an individual.

The proposed architecture is process model independent. It does not favor any particular representation of process. However, we again advocate the use of component-based process models. Component-based process modeling allows for easier integration of organizational and personal process spaces in the Process Definition Tools and Calendar Managers. Without components, there would be a push on each tool to support low-level protocols allowing for heterogeneous process models to be integrated. This is just the type of interoperability that is deficient in current workflow systems, and a major motivating force behind the component-based approach to process modeling described in Section 3.2.

We have developed a set of Java tools realizing the proposed architecture. In the next section we present our progress with this project.

4.0 The Current Prototype

The YFPPG Research Group at Arizona State University has sponsored a series of Master's projects during the Spring 1997 semester for developing a toolset in Java for component-based process modeling and enactment. This toolset conforms closely to the general architecture presented in Section 3.3. The specific architecture is shown in Figure 7.

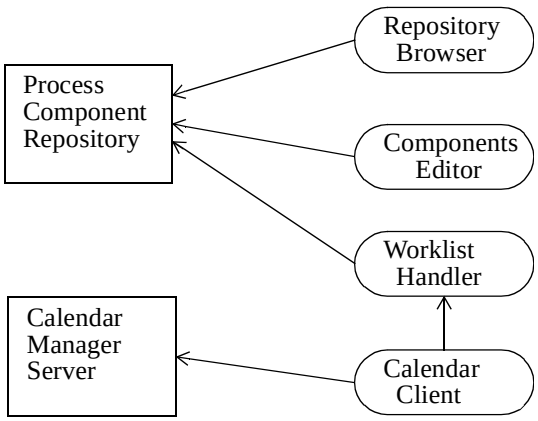

FIGURE 7. OPC Support Architecture

The components of this architecture are:

- *Process Component Repository*

  This is implemented as a Java RMI[22] server that uses Java Serialization facilities to distribute process objects to client tools. The repository stores component-based process definitions and distributed components for

enactment. Multiple named repositories, each storing multiple process models, can be managed by a single server.

- *Calendar Manager Server*

  Java RMI and CORBA[1] versions of this server exist. This server stores time-oriented appointments as well as task lists for individuals.

- *Repository Browser*

  The Repository Browser is a process administration and management tool that allows users to browse through the current objects in a repository. This is implemented as a Java RMI client.

- *Components Editor*

  The Components Editor is another Java RMI client. It allows users to graphically create component-based process models through component creation and reuse. Figure 8 shows the Components Editor GUI with our example process definition from Figure 4.

- *Worklist Handler*

  This client-side tool obtains work items for a user from a repository. The work items are actually Java objects that are serialized and obtained through Java RMI calls. Once the Worklist Handler obtains these objects, it can execute them, changing the state of the process model and invoking tools on work products. Figure 9 shows a Worklist Handler for Bob.

- *Calendar Client*

  The Calendar Client obtains the appointments and task lists for an individual from a Calendar Manager Server. In addition, the Calendar Client can bring up a Worklist Handler to access the Process Component Repository. Java RMI and CORBA versions of this tool exist.

1. Iona Technologies' OrbixWeb[13] was used to implement the CORBA-enabled calendar server and client.

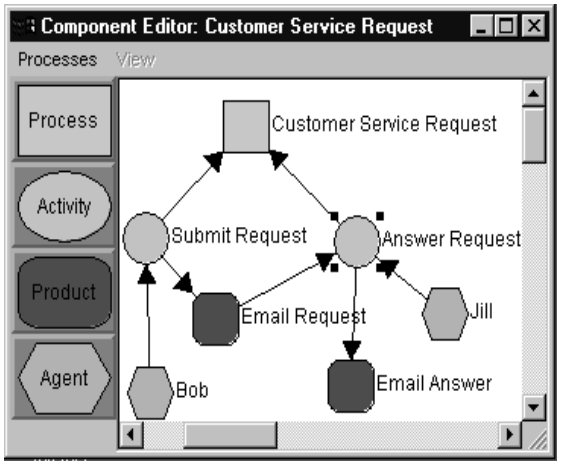

FIGURE 8. Components Editor

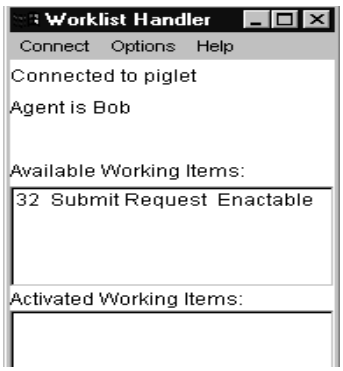

FIGURE 9. Worklist Handler

At this point in the development of our toolset we have yet to implement the full envisioned functionality of the Calendar Manager Server. The overlap of the organizational and personal process space occurs in the Calendar Client, which is responsible for providing the integrated view of the two spaces. The next step is to implement the full negotiation between the two servers, as we discuss in the next section.

We have already learned several lessons during the development and use of this toolset. On the plus side, these tools successfully demonstrate the integration of organizational and personal process spaces. These tools are also demonstrations of forward-looking component distribution technologies such as Java RMI[22] and CORBA[19]. Finally, these tools demonstrate the utility of component-based process modeling. There have been some hiccups however. Managing migrating components in a distributed environment is a difficult configuration management problem. It has proven troublesome to track distributed process components' states and synchronize updates to process models stored in the repository. Despite these problems, we are excited by the possibilities of distributed, component-based process modeling, and are initiating a new set of projects to

update the current environment. Readers interested in obtaining the prototypes or tracking progress of this project may visit the YFPPG website at *http://www.eas.asu.edu/~yfppg*.

5.0 Summary and Future Work

In this paper we have advocated an integrated view of organizational workflows and personal process spaces. In this view, both the perspective of the organization and the perspective of the individual are considered when integrating process spaces. This view allows organizational goals to be pursued while allowing individual workers the flexibility to define how to accomplish such goals. Such flexibility will be required in the not-too-distant future due to the increasing demands on current workflow systems and the current pace of technology.

In this paper we proposed a generic architecture for process support that logically integrates functionality needed for both perspectives. We suggest a component-based process modeling approach to further reduce the dependencies between workflow and calendaring systems by avoiding the need for low-level, brittle data interchange protocols. Finally, we described a set of prototype tools based on component-based process modeling that realizes the generic architecture. Despite the success or failure of our efforts, we hope that the argument for integrated organizational and personal process spaces will have an effect on future considerations in the converging areas of workflow and groupware research.

Given the relatively early stage of this research, there are several avenues we intend to pursue in this area. First, further research is needed to fully understand the nature of the negotiation between organizational and personal process spaces that takes place in the Calendar Manager. We are pursuing research in this area under the topic *Process Component Brokering*, where such negotiation is carried out by having the Calendar Manager provide a brokering service that identifies personal process components that meet organizational process requirements. Second, we are looking at ways to integrate automated planning and scheduling techniques for workflow and personal processes. The result will be enhanced Calendar Managers that negotiate with organizations Workflow Servers to optimize the overlap between organizational and personal process execution. Finally, we plan to validate the proposed architecture by employing our tools in real workflow settings, and extending our work into more dynamic process areas. Specifically, we are looking at ways to support Personal Software Processes and Distributed Learning processes between mentors and students.

6.0 References

[1.] Action Technologies, Inc. Coordination Software: Enabling the Horizontal Corporation. Action Technologies, Inc. White Paper. July, 1994.

[2.] Alonso, G. and Schek, H. Research Issues in Large Workflow Management Systems. Proceedings of the NSF Workshop on Workflow and Process Automation in Information Systems. May, 1996.

[3.] Armitage, J. and Kellner, M. A Conceptual Schema for Process Definitions and Models. Proceedings of the Third International Conference on the Software Process (ICSP3), pp. 153-165, Reston, VA. October, 1994.

[4.] Christie, A., Levine, L., Morris, E., Zubrow, D., Belton, T., Proctor, L., Cordelle, D., Ferotin, J. A Study into the Current Usage of Software Process Automation. Proceedings of the NSF Workshop on Workflow and Process Automation in Information Systems, Athens, GA, May, 1996.

[5.] Conradi, R., Liu, C., and Hagaseth, M. Planning Support for Cooperating Transactions in EPOS. Information Sciences, vol. 20, no. 4, pp. 317-336. 1995.

[6.] Curtis, B., Kellner, M., and Over, J. Process Modeling. Communications of the ACM, vol. 35, no. 9, pp. 75-90, September, 1992.

[7.] Derniame, J.C. Life Cycle Process Support in PCIS. Proceedings of the PCTE '94 Conference. 1994.

[8.] Gary, K., Lindquist, T., and Koehnemann, H. Component-based Process Modeling. Technical Report TR-97-022, Computer Science Department, Arizona State University. May, 1997.

[9.] Georgakopoulos, D., Hornick, M., and Sheth, A. An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure. Distributed and Parallel Databases, vol. 3, pp. 119-153. 1995.

[10.] Godart, C., Canals, G., Charoy, F., and Molli, P. An Introduction to Cooperative Software Development in COO. International Conference on System Integration, 1994.

[11.] Humphrey, W. The Personal Process in Software Engineering. Proceedings of the Third International Conference on the Software Process (ICSP-3). IEEE Press. October, 1994.

[12.] Internet Mail Consortium. vCalendar V1.0 Specification. Available at *http://www.imc.org/pdi/pdiproddev.html*

[13.] Iona Technologies. OrbixWeb 2.0 Programming Guide. November 1995.

[14.] Khoshafian, S., and Buckiewicz, M. *Introduction to Groupware, Workflow, and Workgroup Computing*.

J. Wiley and Sons, New York. 1995.

[15.] Lee, J. Gruniger, M., Jin, Y., Malone, T., and Yost, G. The PIF Process Interchange Format and Framework. Available at *http://www.aiai.ed.ac.uk/pif/index.html*. May 24, 1996.

[16.] McCarthy, D. and Sarin, S. Workflow and Transactions in InConcert. IEEE Bulletin of the Technical Committee on Data Engineering, vol. 16 no. 2. June 1993.

[17.] Microsoft Corporation and Wang Laboratories, Inc. Microsoft MAPI Workflow Framework Concepts and Facilities (White Paper). Available at *http://www.wang.com/sbu/w9602210.htm*. February 21, 1996.

[18.] Netscape Communications Corporation. Netscape Calendar Server 1.0 and 2.0. *Available at http://home.netscape.com/comprod/server_central/product/calendar/calendar2_data.html*.

[19.] Object Management Group. Corba 2.0 Specification. Available at *http://www.omg.org/corbask.htm.* July 1995.

[20.] Reinwald, B and Mohan, C. Structured Workflow Management with Lotus Notes release 4. Proceedings of the 41st IEEE CompCon digest of papers, pp.451-457, Santa Clara, CA. February, 1996.

[21.] Riddle, W. E. Fundamental Process Modeling Concepts. Proceedings of the NSF Workshop on Workflow and Process Automation in Information Systems. May, 1996.

[22.] Sun Microsystems, Inc. Remote Method Invocation Specification. Available at *http://www.javasoft.com:80/products/jdk/1.1/docs/guide/rmi/spec/rmiTOC.doc.html*

[23.] The Workflow Management Coalition. The Reference Model. WfMC Document Number TC00-1003, January 1995.

[24.] The Workflow Management Coalition. Interface 1: Process Definition Interchange. WfMC Document Number TC-1016, Version 1.0 Beta. May 29, 1996.

[25.] The Workflow Management Coalition. Interface 2 Specification. WfMC Document Number TC-1009, Version 1.0. November 20, 1995.

[26.] The Workflow Management Coalition. Interoperability Abstract Specification. WfMC Document Number TC-1012, Version 1.0. October 20, 1996.

[27.] The Workflow Management Coalition. Draft Audit Specification. WfMC Document Number TC-1015. August 14, 1996.