Human-Centered Business Process Management

● Keith Swenson ● Jim Farris

(Manuscript received November 7, 2008)

We present a business process development method that considers human actions and skills up front and then fills in information technology functions to support them. This approach is suitable for business processes identified as "facilitator processes", in which people inherently play parts that cannot be easily replaced by automation.

1. Introduction

When we heard that the Fujitsu Scientific and Technical Journal was going to do an issue focused on human-centered design (HCD), we were immediately reminded of how tough it is to motivate system designers to make business process models human centric. It is not that they purposefully ignore humans in their design: it is more subtle than that. To anyone who works on business processes, it seems obvious that business processes are all about people. However, there is a lot more to HCD than just designing a process that involves people. A process that is human-centered is designed for people to use.

The term "business process" was invented to refer to things that people do, but was later hijacked. It now refers to a program or application that is central to business operations. Along the way, the people-side focus of business processes seemed to get lost, and the more programmer-centric "model driven architecture" (MDA) approaches became more common. While there is nothing fundamentally different in these approaches, the actual practice often drives the organization of the business process in terms of the functions to be performed, and deemphasizes the human side of the process.

The struggle to design human-centered business processes is not a new one for one of the authors (K.S.). In 1993, he published a number of articles describing workflow for humans.[1),2)] At that time, visual programming techniques were relatively new, and the challenge was simply to provide a technical infrastructure to allow people to view and manipulate the graphical process map. Technology has matured a lot in 15 years.

Today, business process management (BPM) technology is a standard budget item on most corporate information technology (IT) budgets. BPM, however, is different from many other IT expenditures. By its very nature, it represents work and represents it in a way that controls what people do. Thus, it provides the means for transforming an organization into a new mode of working. For this reason, the executive management outside IT, sometimes including the chief executive officer, is involved in the selection and implementation of BPM. While IT management (e.g., the chief information officer) is concerned primarily with the implementation and efficient operation of information systems, executive management (e.g., chief executive officer or chief operating of-

ficer) is concerned with the efficient operation of the organization. BPM makes it possible to cross this boundary. If BPM is implemented correctly, it can help an organization become dramatically more efficient. If implemented poorly, it can actually create bureaucracy, shift areas of control and responsibility, and result in no benefit to the organization. Both executive and IT management should pay careful attention to BPM implementation.

2. Definitions

1) Business process

A set of one or more linked procedures or activities which collectively realize a business objective or policy goal, normally within the context of an organizational structure defining functional roles and relationships.[3)]

2) Business process management (BPM)

The practice of developing, running, performance measuring, and simulating business processes to effect the continued improvement of those processes. BPM is concerned with the lifecycle of the process definition.[3)]

When Michael Hammer coined the term "business process reengineering", he meant the "business process" to refer to office work being seen as a process.[4),5)] This specifically means people doing office work, where the steps in the process are tasks that people do. This is to distinguish it from other kinds of processes, such as chemical processes and manufacturing processes, which might also be critical to a business. This is important because in contemporary literature, the term tends to be used quite loosely, sometimes to mean any set of distributed data exchanges critical to a business. Furthermore the "M" in BPM refers to the ongoing management of those processes, not simply the execution of a model of that process—that would simply be business process support. There are two kinds of IT support for human business processes:

3) Automation

The practice of taking a human business process and eliminating the need for people to be involved.

4) Facilitation

For processes where humans have to remain involved, and the process support is to provide those people with the information necessary to do their job and to collect and record their responses when concluding the activity.

Both automation and facilitation of processes can be considered part of BPM, as long as the other aspects of BPM are present, including measurement of process effectiveness with the ultimate goal of continually improving the process.

Continual improvement of processes is an important part of BPM, and that is not new. In fact, BPM can be seen as the modern day equivalent of the Total Quality Management (TQM) movement of the 1980s, also known in some realms as kaizen. TQM morphed into business process reengineering in the 1990s. Workflow was a technology developed during the 1990s, principally to support business processes and often the reengineering of those processes. Like BPM, workflow is a term that has been used for a variety of different things. Some workflow technology focused exclusively on E-mail messages between workers, passing documents as attachments. Other technology was database oriented and collaborative in nature. While other technology was oriented toward integration of servers, a business process management suite (BPMS) includes the capabilities of such workflow systems, as well as a lot of newer capabilities in order to embody the concept that processes are not static.[6),7)]

BPM often involves an activity called business process modeling. By modeling, we mean drawing a picture of the business process. MDA is a popular term promoted by the Object Management Group (OMG), which defines a way to create platform independent models to describe an application. The stated goal of MDA is to "separate business and application logic from the underlying platform". A graphical language for doing this has been developed: the Unified

Modeling Language (UML). UML is fairly well accepted in the IT field as the modeling language for describing software applications and solutions. Many tools have been developed to support UML modeling as well as ultimately bidirectional code generation through to applications and solutions.[8)]

Discussion of business process modeling is often confounded by comparisons with UML. Experts in application modeling look at business process modeling as just a specialized branch of this same subject area. Though a business process should be platform independent, that is not its goal. The goal of business process modeling is to provide a way for business users (read: non-programmers) to participate in control of the business process. This is important because the rate of change in business is so fast that waiting to arrange for a programmer represents an unacceptable delay in responding to market and legislative changes. It is critical that business people can manipulate the process models directly.[9)]

Evidence that business process modeling is distinct from application modeling is seen in the fact that OMG also supports a separate graphical notation called BPMN for modeling business processes. Yet today, most of the BPMN-based tools seem to be oriented toward programmers. This is seen in the way that they describe a business process in terms of data exchanges. There is a "send data" activity, a "receive data" activity, and a "transform data" activity. In our experience, these are not concepts that business managers naturally deal with. They deal instead with the concept of "what people are doing". They want to facilitate work done by humans, they want to draw a diagram of what the humans do, and they want to hide the details necessary to send bits and bytes around the system.

2.1 What is a human activity?

Before anyone will perform a task, they certainly must (a) be informed that the task needs to be done, (b) be given the details of the particular case, and (c) have a way to record the results of the activity. These are part of facilitating any human activity. When modeling human activity, we focus on the work to be done: wash the dishes, feed the dog, write an article, or decide the menu for dinner. Naturally, for a group of people to coordinate on these tasks, there must be communications between them, but we don't model the communications. If I want my son to wash the car, clearly, I have to tell him that I want him to wash the car, but I don't think of that as a separate activity in itself. Instead, it is part of getting the car washed.

It should not come as a surprise that systems designed for supporting human activities allow you to model the work that is to be done at every step in a process, without worrying about how you will tell that person to do the work, or how the results are collected. Such systems often include customizable ways that enable each user to decide how they wish to be informed: some users prefer E-mail, others like to receive a short message service (SMS) message on their mobile phone, etc. As a business manager, I want to focus on the task to be done (e.g., review this document) and should let the system take care of how that user is informed about the work to be done. Similarly, I know that an activity may be concluded with a decision (e.g., to either accept or reject the document), and that may affect the path that the process takes, but I do not want to be too concerned by how the system collects that response.

Beyond the abovementioned three required aspects of a human activity, many human facilitation systems include the concept of (d) a deadline date for an activity, as well as (e) reminders about the activity and warnings that a deadline is approaching. These are convenient built-in capabilities to help manage the work.

So keep in mind that a human activity is a description of actual human work to be done, and that each activity is assumed to have (a) notification, (b) information, (c) conclusion, (d) deadline, and (e) reminders built in. The ten-step method

in Section 3 can be used to create a model of a human process.

2.2 What human-centered BPM is not

In many ways, it is easier to talk about what are not human-centered business processes. Programmers familiar with system design will approach the problem of a business process the same way that they approach any other program. They start with a set of data. It must then be read, transmitted, received, and manipulated. Instead of describing what a person should do, they tend to start by describing the data that will be sent to the person and the data that should be received back. For example, they may draw a box on the process diagram and label it "Send E-mail telling user to review document". This means that a job must be done. Then, they draw another activity and label it "Receive emailed decision from user". These are the things that the computer is doing. The human is "reviewing the document" or "deciding". The distinction is subtle but very important. This should not be surprising. The following is precisely the way that programs are developed:

1) Data values are read and written.
2) Data values are combined and manipulated to make a running program that accomplishes a goal. This can even be the goal of supporting human work.

This description of data flow through the system can be easily modeled using UML, and it can later be converted into an executable form. Some call this BPM, but it is most decidedly not human-centered BPM.

The diagram that describes the input and output of information from the system is useful to a system programmer or to a system administrator. The diagram that describes the human activities is useful to the people in the office that are doing the work. The human-centered BPM diagram can be used to train people. It helps to show people not only what they have to do, but what the other people around them are doing. The system-oriented business process diagram does not tell the people what they are supposed to be doing; instead, it shows what the system will be doing.

3. Method for defining human work

3.1 Step 0: Clarify the goal

Step zero is assumed. Before starting any endeavor you must be clear on what you want to accomplish. As step zero, it is not really part of the process, but success depends upon it to such an extent that it is worthy of mention here. The goal should be clearly written and have an objective measure of success. The process that will be developed will be successful when it is able to achieve this goal.

3.2 Step 1: Identify human work

Start by enumerating the tasks that must be done by people. Ignore for the moment the paper form, the data on the form, and how that form is passed around. Those who expect this to be a programming exercise may be tripped up by this because of the tendency to focus on the artifacts that help people coordinate their work. At this point, we need to look at the work itself. These are tasks that depend upon human skill to be accomplished. We can divide tasks that need to be performed by a human into three categories:

- Some decisions to be made cannot be automated and must be made by a person. For example, the determination of whether an article is fit for publication is a task that depends upon recent current events, suitability of the writing style, and the editorial preferences of a particular publication. Another example is the decision of which candidate is the best fit for an open position. This is a task that depends upon the personalities of the candidates and the teams they would join, as well as an assessment of the skills and ability to perform certain jobs. These decisions must be performed by a person because the most relevant attributes might not

be able to be expressed in a quantitative way, like political correctness or personality. The rules behind what constitutes acceptable quantities of these are tacit and are not consciously known by the people who evaluate such rules. Nevertheless, there are people who are very good at making such decisions. This is work that will never be automated.

- The second category is tasks which might one day be automated, but would require additional preparation. For example, you might need someone to enter figures from a financial report received either on paper or in an electronic format that is not easily usable. For the time being, it is simply less expensive to pay someone to do this than it is to pay a programmer to write code that automatically converts the information. Eventually, these tasks will be automated.
- The third category consists of physical tasks that must be done outside an information system. For example: driving a forklift to load goods from a truck into a place in a warehouse or performing maintenance on a piece of equipment. It might be possible in the far future to automate these tasks with robots, but there are significant barriers to automation due to the physicality of the task. For the time being, we must treat these tasks as human work.

These human tasks are made explicit so that people with the right skills can be identified, or so that people can be trained to do them. Everyone involved in the process needs to know what they are to do—not just those performing the task—so that everyone gains an understanding of how the tasks they do fit in with what the others are doing. The human tasks must be described in a way that the people themselves will understand using the specific vocabulary that the people in that organization use. There will normally need to be additional documentation associated with the tasks that contains detailed information useful for training or skills identification.

Avoid including activities that do not involve humans. For example, running a query on a database is something that might need to be done at some point to support a human task. At this point in the method, however, you simply assume that the right information is available. There is a later step that defines what information must be available, and a final step that defines how that information is retrieved, but those steps should be defined at the right point, which is much later in the method.

3.3 Step 2: Determine activity conclusions (choices)

Human tasks can be concluded in more than one way. For example, the decision of whether to accept or reject an article for publication will be concluded in one of two ways: “accept” or “reject”. The conclusion of an activity is an explicit part of the activity itself. In many situations, there may be a third conclusion to this example activity, which is something that means more or less “I am not qualified to make this decision”. That is a possible way that an activity might be concluded. Some activities will have acceptable time limits and may be concluded simply by the passing of time. Each conclusion is given a name.

Conclusions are important communication events. When you model a human process, you are modeling things that must be communicated to the people involved in the process. Take for example the process of writing a book where many people are involved in various roles such as writer, reviewer, and editor. The writer will at some point declare that the book (a particular draft) is ready for review. While this concludes one phase of writing, more importantly, it tells others that they may start their activities of reviewing and editing the current copy. The conclusion of a human activity is most often a speech act known as a “declaration”.[10] A declaration is a statement that, in the act of being uttered, changes the state of a group of people. Declarations often redefine what many people are expected to be doing. So it

is with a modeled human process: the completion of one activity redefines what other people in the process are expected to do.

A conclusion should be considered a distinct conclusion only if it matters to the group. Take, for example, the task "Answer a question". You might regard the answer to the question as being the conclusion of the activity, and there are one or more answers to every possible question that might be posed. Clearly, it is nonsense to consider every possible answer as a possible conclusion of the activity. Conclusions are grouped into sets that will affect the flow of the process later on. To be specific, if the flow of the process does not depend at all on whether or not the task is completed, then it is sufficient to say that there is only one conclusion: "done". The President of America is given the choice to "sign" or "veto" a piece of legislation, and the process continues in different directions depending upon how this task is concluded. However, there is a time limit, and if congress dismisses the bill before it is signed, then this situation is called a "pocket veto". A pocket veto is considered to be completely identical to a veto as far as the process is concerned, so we would not need a separate conclusion for pocket veto: the timeout rule would simply be another way to conclude the activity as a normal veto. The point illustrated by this example is not that the way you accomplish the task is irrelevant, but that what is important is the result—it doesn't matter in terms of the process how this is accomplished in this case.

3.4 Step 3: Put the tasks into order

The work and conclusions should be identified without getting overly involved in the sequence of activities. In many cases, it is clear that a particular task must be done before or after another related task. There will also be branches and certain tasks that are done only if certain conditions are satisfied. This is where a diagramming tool is useful, but only if it can describe activities at the human level. If one activity must be completed before another, and that other activity can start as soon as the first has been completed, then an arrow is drawn between them.

If an activity can be concluded in more than one way, and if each conclusion would cause the process to proceed in a different direction, then there can be an arrow coming out of that activity for each possible conclusion. Clearly, if the point of an activity is to accept or reject an article for publication, the process that continues after that point will be very different. Because this decision is the very point of the activity, the process becomes easier to read if there is a direct connection between the activity and the direction in which the process goes. Some modeling software cannot represent process direction in this way. Instead, such programs save the conclusion into a variable, which is then tested at a subsequent branch gateway. This is an accepted and common practice, but because the branch is removed from the human task, it is harder to see the direct causal link.

The result is a network diagram of the human activities that must be performed properly set in a process which indicates the conditions and order of the activities.

3.5 Step 4: Determine performers

After the tasks and their order have been identified, one must determine who should perform the tasks. This is highly dependent upon a particular organization. It also changes from case to case. In some cases, there will be a pool of people qualified to do the task, and anyone from that pool might be picked. What must be determined at this point is what set of rules will be used to determine who should do a particular job. It might be that a person with a particular skill is needed, and if a directory listing all the people with that skill exists, then the rule is to find those people and pick one. More often, the requirement will be that a particular person should be chosen because of his or her responsibility in a particular part of the organization. For example, there may

be a person designated to handle requests from a particular customer, or there may be a person who is designated as handling all the purchase requests for a particular department.

Unfortunately such a rule cannot be specified without specific consideration of the organization that will be using the process. Each organization will have unique organizing principles, some of which are based on historical accidents. Even across a single organization, the rules to determine who does a particular activity may not be consistent. Any organization that grew by mergers of other organizations will have some "special" parts of the organization that are not like other parts. There also needs to be consideration for the specific representation of the organization in an organizational directory. If skills are not tracked, then they cannot be used to determine the person to perform the activity.

There will generally need to be an expression of some sort which can be evaluated in the context of the organizational structure that resolves the assignee of a particular task. This expression will usually make use of pre-existing groups and/or job titles in the organizational directory, but it may require new groups or job titles. There may need to be multiple levels of groups that include groups which in turn include other groups. In some cases, it may not be possible to determine a priori who will perform a particular task. In some cases, the assignee expression will narrow a selection down to a group of people, but immediate circumstances (like "who is available") may be necessary to select the final assignee. It might be necessary for the users to self-select for a particular job. There may need to be case-by-case adjustments because it is not possible to know everything in advance.

3.6 Step 5: Determine the information to be used

Here you specify a schema or a set of schemas that carry the informational context within which all activities take place. If the process is for a customer to open a bank account, then there is specific information that must be used for that process, such as the customer name, address, and references to other accounts or credit history. The context schema must be a superset of all information needed for every activity. For example, if there is an activity to assess the property value of a house, then clearly the details about the home address, prior sales information, and various reports about the locale are necessary to perform this activity. If one activity produces a result that is necessary for a later activity, such as the assessed value of a house, then there must be a variable that will hold that information between activities. By considering the information requirements of every activity in the process, you can compile the complete context schema required by the process.

The content information will be modeled differently by different implementation engines. For some, there is a single schema for the context that is shared by all activities (effectively a union of all schemas required by the individual activities). Others have a collection of schemas that are transformed back and forth through the process. Either way, the idea at this point is to identify the information requirements of the entire process.

3.7 Step 6: Define access to information at each activity

At some points in the process, certain parts (variables) within the shared context can be read and updated. At other points, that information can be read but not updated. There is also a point in the process at which the information is completely hidden because it either has not yet been specified at that point in the process or is not relevant to that particular activity.

3.8 Step 7: Determine time limits

An activity may have a requirement to be performed in a particular time period. What happens when that time period is exceeded? Does

the process continue without the activity being completed, or does the process "fail" and go down a different path. There may be reminders to the user that the task has not yet been completed. There may also be escalation to other people or management if the task is approaching the deadline without having been completed. At this point for each activity, all time-dependent behaviors should be considered. Some tasks may have no time dependency at all and may be allowed to remain uncompleted indefinitely.

We know that time equals money, so it is worth considering at this point the cost of every activity, as well as the cost to the organization of either delaying an activity, or not performing it. If you are simulating the execution of the process, these costs entered into the model can be accumulated across a simulation run in order to guide the further design of the process.

3.9 Step 8: Design the presentation of the information

This puts a face on the context information, mapping the schema to a visual presentation. This presentation might be specific to a given activity, or might be the same presentation over the entire process.

Humans don't read extensible markup language (XML) directly. Instead, the information has to be displayed in a way that is meaningful to the user. To be effective, the display should be organized for ease of use. Some of the information may be keys or links to other information, and the display should provide an easy way to access those external sources of information.

Technology for presenting the information is often described as "forms" in the BPM community, but you should keep in mind that any technology that can take data and generate a user interface can be used. The choice will depend on many factors outside the BPM system. Some organizations will choose Visual Basic or Java Swing because they have programmers experienced in those areas. Some might choose PHP or other web techniques. They might have powerful forms software designed specifically for this purpose. The process definition method should not get bogged down at this point in the specific requirements of the technology to be used. Instead, this step should focus on the look and feel of the displayed information.

3.10 Step 9: Integrate into information services

This is where the information needed in a process can be picked up from various sources and sent to various destinations. We use the term "service" in the generic sense of a service oriented architecture (SOA). This might be through web service calls or any other means to access other service types. The point is simply that there is a human activity that needs a particular piece of information, so this is where you specify how that information will be retrieved for that human user.

This is the step where you finally consider how data will be sent and received between computers. Many process designers start by considering how data will be transferred through the system, and it leads them to a communications centric view of the work. It can lead to activities that are optimized for computer communications, instead of being optimized for human work. Since the human costs far outweigh the computing resource costs in most business processes, it is important to start with the human tasks and then work down to the integration tasks.

To enable access to information from a web service, some of the process context information will need to be transformed appropriately into XML code needed as input to a web service. The resulting XML code may need to be similarly transformed to be put back into the process context. For example, if it is in an accounting application, the process may need to access a credit rating service to retrieve the applicant's credit rating for consideration in the application process.

Services are used not only for information retrieval, but also to send the results of human tasks to destinations outside the process context. For example, if the decision is made to approve a loan for a particular amount to a particular customer, then there are various parties that may need to be informed about this decision (e.g., by E-mail) and there would also be calls to services to actually set up the account and initiate the sending of a contract to the parties involved.

4. Technology to support human-centered business processes

Fujitsu's Interstage BPM is a system that was designed to support human activities directly, for the purpose of facilitating human processes. It models human action directly. The description of an activity gives a description of what a person will do. For example, review and approve a document. The human activity is a first-class object[note)] in the system. When you design the process using the Studio application, you place a rounded rectangle shape on the canvas. The properties of this object define the activity that the person will do.

4.1 Expose the facilitation, hide the automation

The human activity may need some information collected for it. In order to prepare for this activity, Interstage BPM has a kind of scripting language known as Java Actions that can manipulate data values, gather data from other places, and convert data into other forms. It is these actions that would allow you to "automate" a business process. There are several sets of actions that are responses to different situations. For example there is a set of actions that are called when an activity starts. There is another set of actions that are called when the activity is completed. In Interstage BPM, we do not call these actions "Activities" as some competitive products do. If we did, they would be activities performed by the computer rather than ones performed by the human in the process. To keep the concepts clear, we keep the terms activity and action distinct and separate.

4.2 Interpreted for direct representation

The process diagram, then, is a collection of activity objects and transitions (lines). The process definition is not compiled into any other form for execution. Unlike in some competitive products, it is not converted to Java byte codes for execution. Instead, the process diagram is preserved in its initial form, and the process is interpreted directly from the definition. This has two distinct advantages. The first is conceptual consistency. Remember, a human-centered process is defined as one that focuses on human activity, and humans need to know what the status of the process is. By preserving the original form, we can color in the diagram to indicate the state and display this graphically. The diagram remains understandable as the designer had drawn it. Conversion to another executable form would lose the shape of the diagram and might lose other details about the process definition. Some systems support bidirectional conversion: to executable code, then back from executable code, but the information lost in the first conversion can never be regenerated.

4.3 Interpreted for dynamic changes

The second advantage of interpreting the diagram directly is that this allows the process to be modified at any time, even while it is running. For example, in an interpreted system, an additional activity can be easily added between two other activities. Or an existing activity can

note) First-class object. In computing, a first-class object, in the context of a particular programming language, is an entity which can be used in programs without restriction (when compared with other kinds of objects in the same language).
(http://en.wikipedia.org/wiki/First-class_object)

have a slightly different description or data set. If you were to compile to another form, that other form would have lost the details needed to allow this kind of change.

4.4 Human activities packaged for reuse

So a human activity represents what the person does, but internally it has a lot of built-in behaviors that are performed by the system. There are timers that can provide alerts and reminders if the activity is not attended to by the due date. All of this is bundled into a package called an "Activity", which can be copied and pasted into multiple processes. This is a unit of work in the eyes of a manger, and the ability to represent that human work is the key to human-centered process design.

5. Conclusion

The nine steps described above lead to a model of a human process. This is not a complete methodology by any means, but still a useful one. The steps are repeated iteratively, with reviews at various points. After each step, there is usually some segment of the organization that is interested in reviewing the progress. It is also true that later steps will turn up details which were left out of earlier steps, so there is some iteration through the method multiple times. A good system will allow simplistic execution of the process before you complete it, so you can try out the process along the way. After Step 3, you should be able to run simulations of the process in order to gain confidence in the correctness of the process. After the process has been implemented and deployed, you can collect statistics on how well it is running and cycle back through this to improve things. We call this "business process management" because you are never completely finished designing the process. This method is repeated as long as the process can be improved, and there are always new ideas about how to improve the process or respond to external changes.

References

1) K. D. Swenson, K. Irwin, R. Maxwell, T. Matsumoto, and B. Saghari: A Business Process Environment Supporting Collaborative Planning. *The Journal Of Collaborative Computing*. Vol. 1, No. 1, pp. 15–34 (1994).
2) K. D. Swenson: Visual Support for Reengineering Work Processes. *Proceedings of the Conference on Organizational Computing Systems*. Milpitas, California: ACM Press, (November) 1993.
3) Workflow Management Coalition: Workflow Management Coalition Terminology & Glossary. Lighthouse Point, Florida: Future Strategies Inc., 1999.
4) T. H. Davenport and J. E. Short: The New Industrial Engineering: Information Technology and Business Process Redesign. *MIT Sloan Management Review*. Vol. 31 part 4, pp. 11–27 (Summer 1990).
5) M. Hammer: Re-engineering Work: Don't Automate, Obliterate. *Harvard Business Review*, pp. 104–112 (July/August 1990).
6) J. R. Jablonski: Implementing TQM. 2nd ed. Technical Management Consortium, Inc., 1992.
7) M. S. S. Morton: Corporation of the 1990s, Information Technology and Organizational Transformation. New York, Oxford University Press, 1991.
8) UML Resource Page: *http://www.uml.org/*
9) H. Smith: BPM and MDA: Competitors, Alternatives or Complementary. Business Process Trends, (July) 2003. *http://www.bptrends.com/publicationfiles/07-03%20WP%20BPM%20and%20MDA%20Reply%20-%20Smith.pdf*
10) M. Schrage: *Shared Minds: The New Technologies of Collaboration*. New York: Random House, 1990.

Keith Swenson
Fujitsu Computer Systems Corporation

Mr. Swenson received a B.S. degree in Physics and an M.S. degree in Computer Science from the University of California, San Diego, USA. He joined Fujitsu in 1991 and started the Regatta Project which produced "TeamWARE Flow" and "iFlow". He was a software architect at Netscape for the internet workflow product "Process Manager" and the developer tool "Visual JavaScript", which helped developers create live JavaScript applications using a technique known today as "AJAX". He returned to the Fujitsu Group in 2002 to direct the development of the Interstage family of products at Fujitsu Software Corporation. He is known as a pioneer of web services and has helped the development of standards such as WfMC Interface 2, OMG Workflow Interface, SWAP, Wf-XML, AWSP, and WSCI. He is currently working on standards such as XPDL and ASAP. From 1995 to 1997, he served as Vice Chairman of the ACM Special Interest Group for Group Support Systems (SigGROUP). In 1996, he was elected a Fellow of the Workflow Management Coalition. In 2004, he was awarded the Marvin L. Manheim Award for outstanding contributions in the field of workflow. He is currently the Chairman of the Technical Committee of the Workflow Management Coalition. His present position at Fujitsu America is Vice President of Research and Development for the Interstage family of products.

Jim Farris
Fujitsu Computer Systems Corporation

Mr. Farris received a B.A. degree in Mathematics from Michigan State University, East Lansing, Michigan, USA in 1982 with course work in Engineering and Computer Science. He is a Senior Technical Writer for the Interstage family of products. For the last seven years, he has written the user manuals for Interstage BPM and managed their development. In this capacity, he has become an expert user of the product. Now, he coordinates an advanced design team that is creating requirements and functional specifications.