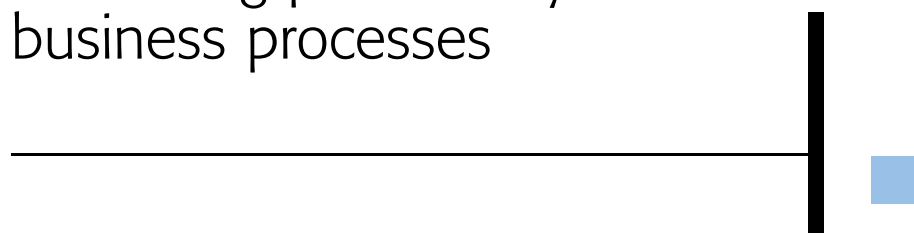


Beyond predictable workflows: Enhancing productivity in artful business processes



C. Hill
R. Yates
C. Jones
S. L. Kogan

Until now, the greatest productivity gains in business processes have been achieved by formalizing the processes into computer-managed workflows. However, many processes have not yielded to this approach, and in its stead, users have depended on ad hoc collaboration tools, such as e-mail and instant messaging, to coordinate their work. While undeniably useful, these tools are disconnected from process methods and can become overloaded and unproductive. Through use cases, we show that many business people are, of necessity, integrators of information technology (IT), but receive inadequate support from centralized IT. We maintain that productivity will be increased by better enabling users to select and integrate IT services as their needs evolve, promoting a shift that we call the *democratization of process*. With the organizing principles of activity-centric computing and the arrival of valuable online services and decentralized methods for integrating them into existing applications, such a shift is now becoming technically feasible—a goal that enterprises should pursue.

INTRODUCTION

In most companies, managers are under pressure to reduce costs and improve productivity. In this paper, we give a practitioner's perspective on some of the challenges of improving workforce productivity and offer some emerging technical solutions that can be used to support an activity-centric approach to managing work.

Industrialization of information work

Businesses have made enormous investments in enterprise applications from vendors such as SAP AG and Siebel, Inc. We begin by discussing the role

of these solutions in enhancing productivity and the limits of their capabilities.

In a complex business process, each actor performs only some of the steps, and few people fully understand how the entire process works. Enterprise applications codify and compartmentalize the steps to guide users through the task at hand.

©Copyright 2006 by International Business Machines Corporation. Copying in printed form for private use is permitted without payment of royalty provided that (1) each reproduction is done without alteration and (2) the Journal reference and IBM copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free without further permission by computer-based and other information-service systems. Permission to republish any other portion of the paper must be obtained from the Editor. 0018-8670/06/\$5.00 © 2006 IBM

Although very expensive to implement, enterprise applications are commercially successful. They are used by a great number of companies and are considered to be mission-critical.

We consider processes managed by enterprise applications to be *industrialized* when they are formalized enough to achieve consistent results that are largely independent of the users. An enterprise application can furthermore achieve economies of scale in a process when the benefits it delivers increase with the number of employees involved in the process. (Other important functions of enterprise applications not directly concerned with employee productivity, such as rapid supply chain communications, regularity, and compliance, are not considered here.)

Prescriptive, highly formalized process applications have enjoyed great success. There are, however, definite limits to this approach. One immediate problem is how to enable business people to use enterprise applications. To extend their value beyond a core group of highly trained users, companies implement self-service user interfaces that enable employees to quickly accomplish routine information-processing tasks without intervention by specialists. Many human resource (HR) processes, such as hiring, promotions, and performance reviews, are candidates for self-service because they require the participation of individual employees and the transmission of personal information.

To support the delivery of self-service user interfaces, IBM enables users of IBM Lotus Notes* to access processes in SAP and other systems,^{1,2} and IBM WebSphere* Portal Server³ can also integrate a variety of systems, including SAP and Siebel. Microsoft is now working with SAP to bring processes into the Microsoft Outlook** client.⁴ These initiatives show that it is possible to significantly broaden access to enterprise applications.

Limits on the industrialization of information work

As laudable as these efforts are, profitable use of enterprise applications for enhancing productivity has its limits. When the cost of formalizing a process is too high, an alternative approach is needed. Some of the factors that limit the industrialization of information work are scale, risk of lock-in, depen-

dency on incumbent systems, and artful processes. We discuss them in the following subsections.

Scale

Because of the high cost of entry, some companies, especially small ones, cannot adopt enterprise applications. In an organization of any size, the cost of implementing a particular process may outweigh the productivity benefits for the users affected. Thus, at least when considering productivity goals, the complexity of the process and the size of the workforce involved need to be considered.

Risk of lock-in

Many companies, regardless of size, choose not to move certain processes into an enterprise application because of the dangers of locking in a process determined by a third-party vendor. For example, rather than use a job-posting module that comes with an enterprise application, a company might use a more efficient online service, such as Monster.com, competing on the open market for new employees and saving costs at the same time. As more compelling online services become available, this consideration becomes more important. On the other hand, the need to differentiate an aspect of customer service from that of competitors may also lead a company to avoid a standard solution and develop a more custom approach.⁵

Dependency on incumbent systems

Most large organizations have many incumbent legacy systems. Because some processes depend on legacy systems that are too costly to replace, the processes cannot be moved into the preferred enterprise application, even if managers wanted to move them. Furthermore, many processes cut across IT system boundaries. For example, to bring a newly hired employee on board can involve such activities as transactions with the HR system, an account request into the systems administration group, bookings for education and training, and communication with the hiring manager. Again, such processes may be too expensive to reimplement.

Artful processes

Aside from the issues of scale, lock-in, and dependency, certain types of work simply cannot be formalized well enough to safely entrust to an enterprise application. The goals and methods of some processes change too quickly over time; for example, the process of designing high-technology

products. In some processes, it is primarily the content in each process instance—rather than the process itself—that determines the outcome; for example, a request for proposal (RFP) process.⁶ Most important, many highly specialized processes are developed or refined locally at the individual or small-team level such that the process cannot easily be separated from the specific people who perform it; for example, managing client relationships in professional services firms. While the framing process may be stable at an abstract level, the key details are not. They depend on the skills, experience, and judgment of the primary actors. We denote these kinds of processes *artful* in the sense that there is an art to their execution that would be extremely difficult, if not impossible, to codify in an enterprise application.

Long tail of business processes?

In certain industries, such as professional services, artful processes are clearly the norm.^{7,8} However, artful work is not always easy to detect. When enterprise applications were first deployed to automate the sales process, over-reliance on the formalized aspects of the process sometimes caused major business failures. With hindsight, no one disputes that there is an art to selling that cannot be captured in a process application. More generally, there are many difficulties associated with accurately modeling business intentions in enterprise applications.⁵ Perhaps more work is artful than is readily apparent.

Indeed, we wonder if there is a long tail of business processes (*Figure 1*). In certain statistical distributions, "... a high-frequency or high-amplitude population is followed by a low-frequency or low-amplitude population which gradually 'tails off'. In many cases the infrequent or low-amplitude events—the long tail—... can cumulatively outnumber or outweigh the initial portion of the graph, such that in aggregate they comprise the majority."⁹ Were it possible to create a distribution of business processes ordered by the amount of resources invested in them, we wonder if the total investment in the many less formalized processes far outweighs those implemented in enterprise applications. If so, a renewed focus on enhancing productivity in these kinds of processes is surely imperative.

Claims of the existence of a long tail of processes have in fact been made before,^{10,11} although we

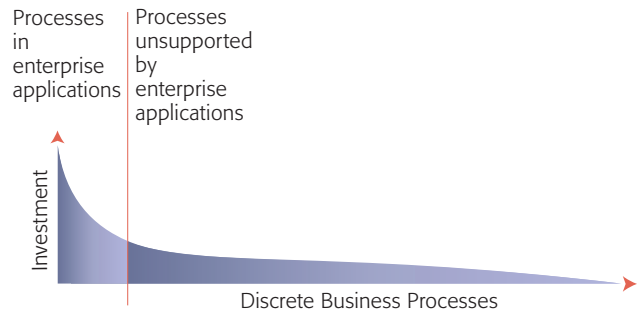


Figure 1
Does a long tail of business processes exist?

have not found any published data to support the claims. This is an important area to address in future empirical research. Meanwhile, one point of evidence supporting this view is that enterprise application vendors are actively seeking to make their systems more accessible to a larger proportion of the employee base. It suggests that most employees today do not directly interact with enterprise applications, and it suggests that the processes the employees are executing are apparently largely unsupported by the enterprise applications.

Outline of our current research

As artful work is clearly central to many businesses, we propose that productivity will be increased by supporting and enhancing artful processes rather than by stripping them down to highly formalized industrial methods. The focus of our research is to find ways to improve productivity by enabling the primary actors—regular business people—to define and continually improve their processes rather than follow a centrally planned model. We call this shift *the democratization of process*. It means creating new decentralized IT architectures that enable business people to more easily exploit a web of existing and emerging IT services in their diverse daily activities. In the remainder of this paper, we present our research into ways of achieving that result.

The widespread use of ad hoc collaboration and personal information management tools to help execute business processes is already documented.^{12–17} In the next section, we present two additional user studies that we conducted to learn how business people get their work done. The first study provides a cross-sectional view of a range of

changeable and flexible processes in five companies. The second study looks in detail at the interactions involved in the hiring process at a small company and identifies specific “pain points” (impediments to productivity) that need to be addressed. In particular, this study illuminates the pivotal role of the lead actor in this process as an integrator of people and information across organizational and system boundaries. Using anecdotes from customers, we confirm that this is not an isolated pattern, but a common concern in many processes.

In the following section, we examine some important bottom-up forces that shape business processes. We need to understand and embrace these forces in order to design an architecture to enhance artful processes. We examine the increasing role of end users in IT decision making, the importance of ad hoc collaboration tools in artful processes, and the rise of decentralized IT services.

In view of these forces, companies need to redesign and reassemble their business processes in a more flexible way that better reflects the way that people really work. The modern business process touches many IT capabilities, including ad hoc collaboration tools, departmental solutions, enterprise applications, and online services. A walled-garden approach in which all services are contained within one software system is unacceptable. Maximizing choice is important, and centralizing all process definitions in enterprise applications is counter-productive.

However, if process definitions are no longer centralized, what alternative organizing principles can be used to avoid a descent into chaos? An activity-centric approach promises the ability to organize artful work productively while preserving user choice over the services employed.^{18–23} The core idea of activity-centric computing is to organize computer-based work in terms of the activities that people are doing rather than in terms of the tools used. We devote a section to presenting some design principles for an activity-centric solution, with particular focus on the need for a decentralized architecture.

There is a long history of attempts to make the computing environment more modular and service-oriented, with the goal of bringing power over service selection closer to the end user. Generally,

these attempts have not lived up to expectations. So why is the time right now? Many previous attempts failed because the cost and complexity of the engineering approach proved too great for widespread adoption. However, the recent emergence of sophisticated applications and integration methods on the Web, known collectively as “Web 2.0,”²⁴ shows new promise because the methods are generally simple and have already proven that they can be widely adopted in the decentralized environment of the Web. We identify enabling technologies that will permit users to capture, share, and reuse work practices, and link them to the widest possible range of supporting services while allowing for their decentralized design, development, and deployment. This approach will provide a foundation for more productive work environments that users can incrementally adapt and refine as their needs evolve.

USER STUDIES

In this section, we first summarize a user study aimed at understanding how people who do knowledge work manage their processes. We also present a detailed use-case analysis of the hiring process in a small consulting firm, and we summarize some related customer anecdotes.

Ethnographic study of business processes

In 2003–2004, we conducted user research at five different companies to learn how knowledge workers get their work done. We did an observational study, shadowing people as they did their work and then interviewing them at the end of the session. At least two people from each site were interviewed, and each visit lasted approximately three hours. Participants were all from the Boston area and included IBM customers, non-customers, and business partners. All were knowledge workers and considered to be subject-matter experts in their field.

After observing participants as they did their work, we asked about the processes and procedures they used to get their work done and how they collaborated with others to get work done. We took note of all the checklists, procedures, and flowcharts. *Table 1* offers a sample of the processes observed and discussed. Many of these were printed and hanging on a wall or bulletin board, some were handwritten, and all were annotated. The annotations made reference to exceptions, so there were several versions of each checklist to be used under certain conditions.

Some processes were simple, such as how to schedule the monthly meeting or how to buy a book. Others were more complex, such as reporting and tracking adverse events in a clinical trial. We also found that some processes were occurring exclusively by e-mail or instant messaging. Recreating the process or tracking it involved searching and filtering e-mail many different ways to ensure that the latest information was available.

Processes were hard to track, difficult to monitor, and hard to reuse. Some participants felt that each time a new instance was initiated, they had to start over from scratch. They described their processes as idiosyncratic and always under modification. One participant said that his work is always the exception—and that's the rule. Most of the communication was happening asynchronously and was partly paper-based. The workflow modeling tools available were considered too complex for these types of processes and were inaccessible to these knowledge workers.

We also collected artifacts from the study, such as a new employee checklist and a work request form. Several versions of the new employee checklist were posted: one for new hires from the United States and another for non-U.S. citizens. Another version was for people working offsite. This work is described in more detail by Kogan and Muller in this issue.²⁵ Here we briefly summarize some key findings:

- Collaboration tools are often used for processes involving time management and meeting logistics.
- Use of formal process systems is supplemented with personal information management tools, such as reminders.
- Use of formal process systems is reserved for enterprise-level processes, not personal or department-level work.
- Participants identified a diverse set of business processes, such as design reviews, requests for proposals, and clinical trial protocol development.
- Participants described their work as idiosyncratic and frequently modified—there are always exceptions, and processes need to be flexible to accommodate these conditions.

Use-case analysis for the hiring process in a small company

The following use case was obtained from interviews with an HR employee at a small information

Table 1 Sample of processes observed

• New employee checklist	• RFP process
• Organizational chart	• Work order request
• Reserving the LCD projector	• Standard operating procedures
• Travel arrangements, travel reimbursement	• Protocol development
• Document management workflows	• Paper/grant review
• Time management	• Personal reminders for formal processes
• Calendaring and scheduling	• Buying books/procurement
• My meetings	• Inventory tracking
• Organize meetings	• Work action plan
• Compliance-based processes	• Sales and forecasting
• E-learning	• Managing shrink
• Clinical trials monitoring report	• Adverse event tracking
• Clinical trials data analysis	• Tagging
• Prospective process—readme file	• Design review

services company. We conducted it to examine in detail how the process really works. The names are fictitious.

Overview

Lucy works in the HR department of a small company. Hiring is one of her main responsibilities. The hiring process at her company varies substantially from one case to the next. The level of the position being filled and the current economic conditions affect the approach taken. For example, hiring decisions for lower-band positions can be made within the HR group, while more senior positions involve the candidate preparing and delivering a presentation to senior analysts within the organization. If the company is in a period of growth, managers are automatically allowed to backfill vacated positions, whereas in leaner times, backfilling is subject to approval.

To execute the hiring process, Lucy and her team expend significant manual effort bridging disparate IT systems, including dedicated tools, such as a vacancy-posting system, and ad hoc collaboration tools, such as e-mail and the phone. The pain points experienced in doing this work are localized to the HR team and, in this small company, are largely

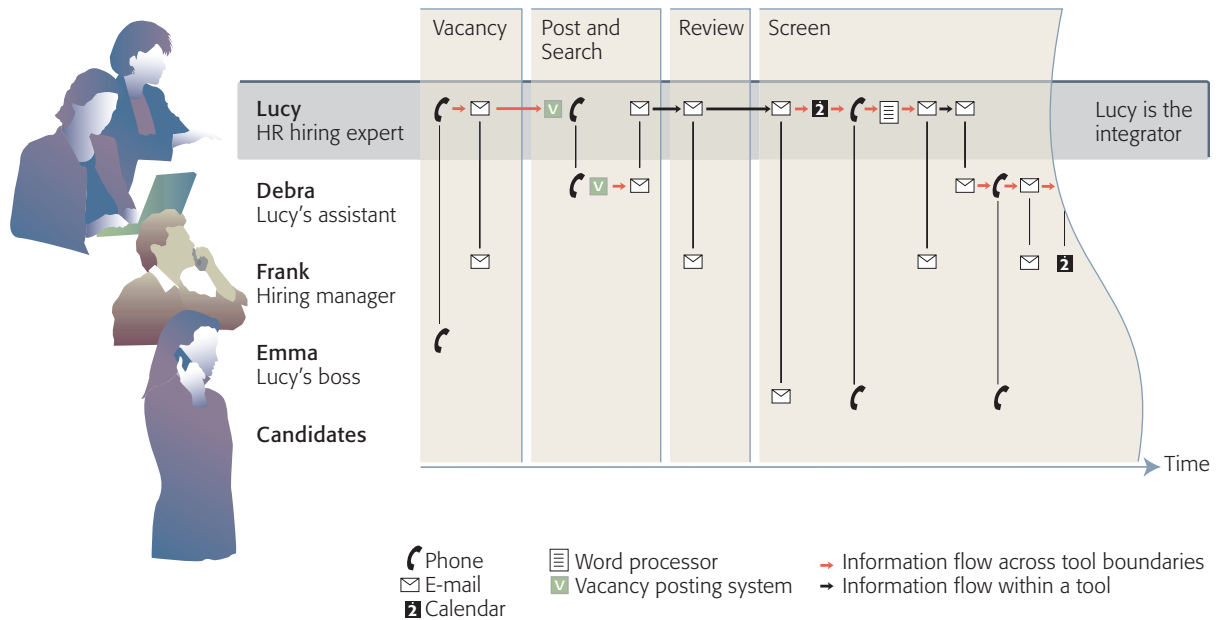


Figure 2
Schematic view of tool use in the beginning of a hiring activity

invisible to the IT department. Even if they were noticed, while it might in theory be possible to run this process as a workflow in an enterprise application, doing so would probably not be cost-effective nor would it fully support the kinds of communication and collaboration required.

The following partial use case describes a portion of the hiring process at Lucy's firm. It starts with the decision to post a vacancy and finishes with the first candidate being screened by phone.

Actors and goals

The following actors participate in this use case:

- Frank, the hiring manager (manager of the group with a vacancy)—Makes the ultimate decision as to whether to hire a candidate
- Lucy, HR hiring specialist—Performs the initial filter of candidates and only forwards to Frank those that she deems appropriate
- Debra, Lucy's assistant—Performs administrative functions for Lucy
- Emma, Lucy's boss

Use case steps

A summary of the use case is given in text form below and visually in *Figure 2*.

1. Emma informs Lucy that Frank has a vacancy.
2. Lucy and Frank exchange e-mails on the text of the job description and then meet to finalize.
3. Lucy posts the vacancy in the online service with the text from the e-mail.
4. Lucy asks Debra to search the online service for viable candidates (i.e., those who have not explicitly applied for the job but have the necessary skills).
5. Debra searches the online service and sends Lucy e-mails with viable candidates by detaching the resumes from the online service and sending them as e-mail attachments.
6. Lucy reviews applicants from the online service (both those who have applied for the position and those whom Debra found).
7. Lucy forwards the candidates whom she deems viable in an e-mail to Frank.
8. Frank responds with an e-mail to inform Lucy of the candidates whom he wishes to pursue and those whom he wishes to reject. He also includes his initial thoughts on the candidates.
9. Lucy sets up interviews by phone for herself and the candidates.
10. Lucy takes notes during the interviews by phone.
11. Lucy informs Frank of the candidates whom she thinks Frank should also interview by phone.
12. Frank sends back a list of the candidates whom he would like to interview by phone.

13. Lucy forwards this list to Debra and asks Debra to set up interviews by phone for the candidates with Frank.
14. Frank takes notes on the candidates during his interviews by phone.

Lucy's pain points

The following impediments to productivity are identified in this use-case analysis:

- *Navigation across system boundaries*—To fill a particular vacancy, Lucy must interact with at least five different systems, namely: her e-mail, her calendaring system, the job vacancy system, the application used when new employees come on board, a word processor (for taking notes), and her phone. There is no simple navigation available to her that allows her to move items in one system to related items in another.

When Lucy's calendaring system informs her that her next meeting is an interview by phone with a potential job candidate, Lucy needs to quickly locate the candidate's resume, the candidate's phone number, and the e-mails from Frank pertaining to the viability of the candidate. All of these documents form a single logical work task: Determine if this candidate is the best candidate for the open position. However, the system boundaries artificially partition the work, forcing Lucy to act as the integrator as she works within each system to find the necessary information.

- *Cut and paste*—Lucy has an alternative approach to finding the information related to the interview by phone. Instead of searching the supporting systems for information related to the calendar invitation just before the interview by phone, she can copy information into the calendar invitation when it is first created. The created calendar entry now contains a copy of the resume and the candidate's phone number. Although this certainly shortens the time spent immediately before the interview by phone, it does not eliminate the effort. This approach merely reorders the work step to an earlier point in the process. Lucy must also copy the resume from the job vacancy application to send it to Frank. Because Frank does not have access to the job vacancy system, he can view the resume only if Lucy e-mails it to him.

- *Lack of semantics*—Today's applications do not exchange semantic information about data, even though semantic information is readily apparent to users. Users can easily comprehend a telephone number, an e-mail address, or a purchase order number while viewing the application user interface. However, these semantics are typically known only to one application and are not shared automatically with other applications. As a result, Lucy is left to interpret the meaning of the data and coordinate manually among the applications that she uses. For example, her company uses a Voice over Internet Protocol (VoIP) application, so she can place calls from her computer. When Lucy is ready to speak to a candidate, she manually integrates the job vacancy system with the VoIP system by typing or pasting the telephone number from the job vacancy application into the VoIP application.

- *Lucy is the integrator*—Although each individual pain point identified above is only a minor annoyance, there are many of them. The result is that Lucy spends much time carefully moving data between the various tools that she uses. Furthermore, because of the lack of integration, each particular tool is capable of representing no more than a small discrete part of the hiring process. Each pain point thus underscores the role of Lucy as integrator. Lucy and Debra, her assistant, manually weave the existing systems together to support their process. This raises several questions. Can we better support Lucy and her team by more closely integrating the systems they depend on? Can systems be built explicitly for more effective integration? If so, what technologies can facilitate this?

Related use cases

We find similar pain points, though not always for the same reasons, for similar activities in much larger companies. In fact, these pain points appear to persist when companies dedicate significant IT resources to the hiring process and are also evident in many other kinds of business activities. Here we summarize some examples provided anecdotally by customers.

Hiring process in a large enterprise

A senior IT executive at a large manufacturing company reported to us that the process of hiring and bringing employees on board in his company is quite onerous, even though the company has

implemented an enterprise application to support its HR processes. It is onerous for three reasons: bringing employees on board involves a large number of discrete steps and many different systems, HR personnel find the user interface of the enterprise application difficult to use, and staff

■ The democratization of process means creating new decentralized IT architectures that enable business people to more easily exploit a web of existing and emerging IT services in their diverse daily activities ■

turnover leads to high training and quality assurance costs. Rather than further centralize the process, the customer suggested that a better solution would be to enable HR workers themselves to create and share checklist-style templates for common activities. Any HR worker could then use these templates as starting points to guide their future work across the various systems involved.

Professional services

Firms providing professional services, such as accounting, legal services, and business consulting, develop highly specialized and artful work processes by the nature of their trade. Despite the high level of art in this work, customers report significant concerns about process: the need to better guide and review the work of junior staff, the need for better awareness of project status and priorities, the focus on minimizing business risks, and the need to streamline collaboration on fast-moving projects.

Customer support

We have also heard from several customers about use cases in customer support. Most companies have dedicated customer-support processes. However, in large organizations in which there are highly specialized job roles, many customers report that their systems are poorly integrated with the informal collaborative activities that occur as the real work of resolving customer issues takes place. As a result, critical information and context is unavailable from within the primary support process application.

System administration

Customers also reported to us the difficulties of implementing consistent administration processes across highly specialized and heterogeneous IT environments. Even more than other business people, system administrators struggle to integrate their work across diverse technical boundaries.

Summing up

Even in the relatively small range of companies contacted in these studies, we identified a great diversity of business processes that are not being served well by enterprise applications. In practice, business processes depend on informal methods of managing work by using collaboration tools and meetings, and on a variety of discrete tools and services. Many processes are not served at all by enterprise applications, and many of the informal methods are very time consuming. Our detailed use-case study revealed several pain points. Although each individual pain point is small, they impose in the aggregate a large burden on the lead actors in the process to integrate their tools as they work, and they arise in a wide array of processes. The lack of cohesive IT support for work activities makes it much harder to track and execute processes, let alone capture and reuse better ways of working.

BOTTOM-UP FORCES ON BUSINESS PROCESSES

Many of the business processes cited earlier appear to be strongly shaped by bottom-up decentralizing forces. Aside from cost constraints, a common reason that many of the processes cannot reach a traditional enterprise application is that the processes are defined by individual business people or small teams as they do their work; they are not prescribed in advance by upper management. Creating and refining processes is delegated, quite appropriately, to the people doing the work. In this section, we discuss three distinct but overlapping bottom-up forces on the development of business processes that we observe from our experience with enterprise customers:

1. End users as IT decision makers
2. Collaboration tools, the entry point to process
3. Decentralized infrastructure and the rise of Internet services

End users as IT decision makers

In recent years, we have seen a trend in large organizations toward giving end users increasingly direct control over IT decisions. In smaller compa-

nies, it has always been the norm. One reason for this shift of power offered by customers is that users often know better what they need than central IT departments. In fact, users are already able to adopt departmental solutions and various online services, whether or not the central IT organization approves. Many successful collaboration systems—such as IBM Lotus Notes, IBM Lotus QuickPlace*, EMC Documentum**, eRoom**, and Microsoft SharePoint**—gained adoption primarily at the departmental level. Online services such as salesforce.com** and Monster.com are even easier to adopt without prior arrangement with IT management. Because it can be difficult to obtain the financial support from the central IT budget needed to solve a problem or because it can take a long time to deploy an application through the central IT department, departmental managers find that these tools offer an alternative that can be adopted locally on a small scale.

However, unplanned software leads to serious risks in terms of noncompliance with, for example, critical policies governing security, retention, and auditing. And once entrenched, departmental solutions can be costly to displace, so they can start to influence central IT policies and system architectures. Recognizing these concerns, many companies now grant significant power over IT spending to end-user representatives in an explicit partnership with the IT organization. As a result, we can expect to see increasingly direct departmental influence over service selection, and we expect business people will increasingly demand methods that enable them to integrate new services that they see as adding value, while staying in compliance with critical IT policies.

Collaboration tools, the entry point to process

In the absence of formalized support for their processes, business people instead take advantage of collaboration tools to communicate and coordinate their efforts. We consider a spectrum of tools, ranging from ad hoc communication tools on one end, through shared workspaces, to enterprise applications at the other end. The spectrum is a refinement of Bernstein's "specificity frontier"²⁶ and is introduced by Geyer et al. in this issue.²¹ Here, we review it from a process point of view.

On the one hand, the simplest ad hoc communication tools, such as the telephone and e-mail (and

increasingly instant messaging), are business-critical for workers in artful processes, just as enterprise applications are business-critical to those who depend on them. There is often no better way to get a quick reply from a co-worker, a customer, or a supplier than to pick up the phone, send an e-mail, or send an instant message. However, for more extensive collaborations, these tools become unmanageable.²⁷

For larger projects, shared workspace tools, such as Lotus Notes TeamRooms (Lotus Domino applications that provide a tool for information sharing and collaboration), were devised to help teams manage shared project content. Tools like these are often first adopted at the departmental level. Like enterprise applications, shared workspaces help to coordinate work among actors in a process. Unlike enterprise applications, the lead actor and the team doing the work typically manage the content and often even maintain their application. Thus, they are better able to adapt the embedded methods to their own needs. In fact, some IBM customers currently operate many thousands of Notes applications, the majority of which were developed by knowledge workers to support numerous kinds of processes. Their complexity ranges from simple uses, such as issue tracking, to processes that rival enterprise applications in both complexity and importance to the business, such as customer support. This proliferation of diverse departmental process applications is evidence of the power of decentralizing the process infrastructure. Today, however, the majority of these applications are largely disconnected from other relevant applications.

At the other end of the spectrum lie enterprise applications, with their ability to industrialize work, within limits. Consistent with our earlier discussion in the Introduction, customers report frustration that very significant collaboration activity continues to take place outside the enterprise applications. As soon as an exception occurs, much of the follow-up work leaves the enterprise application and migrates to ad hoc tools, such as e-mail. Having so much important activity occur outside and beyond the awareness of an enterprise application degrades its effectiveness and management value.

A common problem associated with the proliferation of ad hoc communication tools, shared workspaces, and enterprise applications is information

scatter. As more user-interface destinations are created to help organize the work of different processes, it becomes harder to keep track of all the content, work status, and plans because they are

■ Companies need to redesign and reassemble their business processes in a more flexible way that better reflects the way that people really work ■

scattered among the many participating systems. Dealing with the problem of scatter is a central concern of this research.

Decentralized infrastructure and the rise of Internet services

In discussing the process of bringing employees on board in a large manufacturing company, we noted that many legacy systems were involved in the activity. This situation is common in large companies that have long-established IT systems. Even when IT policy is tightly controlled and strongly centralized, the dominant presence of important legacy systems points to the fact that system design, development, and deployment actions are distributed over time, sometimes decades. When significant numbers of applications are implemented at the department level, internal application development becomes significantly more decentralized.

With the arrival of compelling Internet services such as job postings, travel, information services, and collaboration tools, processes will become even more distributed and decentralized. By using an Internet job-finding service, job candidates gain access to a greater pool of employers than might otherwise be reachable, and employers are able to reach the largest possible pool of candidates. Travel sites give quick access to many flight options so users can choose the best prices or the most convenient itineraries. It is clear then that process support systems must evolve to work well in a decentralized and user-driven computing environment.

Implications for design: Democratization of business processes

We consider the bottom-up forces discussed here to be a mainly positive influence on infrastructure

design, even though they may be disruptive for IT administrators. Increased user influence will presumably lead to more relevant and user-centered solutions. Embracing the role of ad hoc collaboration tools in business processes is clearly necessary, and the rise of online services indicates that the time is right to reform IT architectures to take advantage of the real opportunities offered.

A parallel to the growing influence of end users on business services is found in the democratization of content under the influence of Web 2.0. By *democratization*, we mean a shift from central control of IT services to a greater ability for end users to help themselves. Blogging and wikis²⁸ are recent examples of the trend to democratize content. Given the strong bottom-up forces on process applications, we conclude that there is an analogous need and opportunity to democratize business processes, with real economic benefits to be gained.

Democratization is not just about turning over control of a process and responsibility for operating applications to employees, even though this is a necessary step to supporting more artful processes. It is also about getting the best deal, or about achieving greater access to buyers, employers, or other communities. To take advantage of online services, companies need to redesign and reassemble their business processes in a more flexible way to take advantage of the broadest range of discrete services and infrastructure capabilities.

DESIGN PRINCIPLES FOR AN ACTIVITY-CENTRIC COMPUTING ENVIRONMENT

Business people will not be able to take advantage of discrete IT services unless they are made available in a way that users can readily understand, find easy to use, and accept as a natural extension of their prevailing work habits and collaboration culture. The goal is to make it easier to access relevant services from appropriate work contexts by providing bridging mechanisms that reduce the integration burden on the user. While the detailed design of the end-user environment is beyond the scope of this paper, we propose some design principles.

Activities—the organizing principle

Until now, the business computing experience has been dominated by applications and the boundaries they introduce to working methods. As new integration methods enable users to work more

seamlessly across application boundaries and to mix and match services opportunistically as work progresses, a new organizational model becomes not only desirable, but increasingly essential. The challenge is not simply to make integration easier, but to deal with the problem of scatter without overloading the user with a broader array of information, tools, and services.

Our work builds on the concepts of activity-centric computing, as discussed in detail by Geyer et al.,²¹ Moran,²² Cozzi et al.,²³ and other papers in this special issue. The goal of activity-centric computing is to establish a new organizational concept in the computing environment; that is, the activity, which represents a unit of work. As technical boundaries recede, activities will emerge as the dominant user-centered paradigm for working with distributed systems.

By creating what we call *activity hubs*, which marshal all the tools and information needed to perform work, users will work in a more complete context for their actions and be burdened by fewer manual integration tasks. At the same time, by dividing work into distinct activities, users can focus more readily on a particular activity and deal more easily with interruptions by suspending and resuming activities as needed.

Activities augment applications

Activities do not replace applications. Applications continue to exist as essential technical, operational, and political units and continue to provide critical functionality. For example, an enterprise application that plays a critical role in regulating certain steps in a process will likely continue to play that role. However, through reference-based integration methods, activities introduce new views and organizational schemes that cut across application boundaries, thereby increasing the integrity of the representation of work and mitigating scatter.

In fact, many applications already provide some kind of activity management capabilities to the business processes that they support. For example, in a customer support system, trouble tickets represent root activities, and customer support personnel use the list of open trouble tickets to guide their work. As another example, consider an executive looking at a red flag on a dashboard application that summarizes the status of a complex

set of objectives. The executive should be able to delegate follow-up actions right from the offending line item, so that it is easy to come back to it and see how work is progressing. Activity-centric computing, therefore, must enhance applications without requiring users to go to another destination to manage some aspects of their work.

Self-service

Users gain great benefit from well-designed applications built to address a specific purpose. However, users also need to be able to have control over which services to use in what context. Users need self-service interfaces to access IT services and the methods needed to integrate these services with the user's other activities. This principle favors enabling departmental developers to provide catalogs of services to users so they can easily select and apply them to an activity, rather than, for instance, focusing exclusively on enabling developers to build a specific configuration of services as a fixed application.

Cooperative integration

A key question is whether the concept of activities requires the introduction of a new master application that takes top-down control of the IT environment. This approach clearly cannot work in a completely general way because applications and services remain as centers of policy. In the hiring use case, for example, the job-posting service retains final control over content life cycle and user authorization. It also is undesirable in situations where valuable capabilities for managing activities are already provided by an existing application.

Over time, we can imagine applications beginning to provide negotiated policies, using "webs of trust" (discussed in the section "Enabling technologies") and other methods.

However, a successful activity model should allow some applications to remain in control. For example, the fact that access to a business object, such as a trouble ticket, is controlled by an existing application clearly should not preclude that object from being integrated to a useful extent in an activity that extends beyond the application. Instead of creating a new master application, then, activities need to integrate cooperatively with the IT environment upon which they depend.

Minimally invasive integration

Because a core goal of activity-centric computing is to integrate models of work with diverse IT resources, it stands to reason that the integration methods used should be easy to adopt in the widest range of implementations. The cost of integrating activity-centric computing into existing environments depends on the level at which integration can be done. Integrating each application specially is the most costly approach. Costs and management complexity can be greatly reduced—and a wide variety of systems integrated much more rapidly—by integration either at the platform runtime level through general-purpose extensions or completely outside the services to be integrated; for example, by using forms bound to Web services. The principle is to optimize for broad—as opposed to deep—integration as the default course.

Activity patterns

As discussed elsewhere,^{6,22} activity patterns are a central concept for supporting repeated work. A pattern formalizes the structure and content of an activity and the integration methods it depends on, thereby making it reusable as a template in future activities. For example, an HR worker might create a new hire pattern in place of a paper checklist, and for each hiring activity, start the activity from the pattern in order to take advantage of standard practices. A pattern may include structure, such as a series of steps and the applicable user roles, content describing the activities to be done, and methods for accessing IT resources to get things done. By sharing activity patterns, users can “socialize” best practices and reusable processes (make them part of their company’s social fabric).

Activity service boundaries

Activity management reaches into all sorts of services and business domains. At the same time, the function of modeling and managing activities itself takes the form of an activity service of some kind. There is a danger that the activity service will start to take over the role of other services; for example, content management, various collaboration services, forms design, and runtime capabilities. Going down this path too far would lead to the creation of an entirely new integrated stack of services.

To avoid redundancy with the many existing capabilities, it is important to clearly define and limit the boundaries of the activity service. This service

might be provided by the activity management functions of an existing application, as in the customer support system example, or by a separate service.^{21,23} The core purpose of the service is to associate work products and plans with the activities they support, such as “fill this vacancy,” “interview this candidate,” and so on.

ENABLING TECHNOLOGIES

Until recently, it was assumed that integration was an inherently costly option, and this led to highly formalized integration patterns. Integration also meant strong contractual arrangements between outsourced service providers. Both of these factors placed integration out of reach of knowledge-work decision makers.

With the Web 2.0 generation of new technologies, simpler, more lightweight methods of integration are beginning to emerge. Because these integration methods are relatively easy to implement, they can be adopted widely across a diverse set of services. A simple example is content syndication and alerting through Really Simple Syndication (RSS) and Atom feeds.²⁹ These technologies enable delivery of content to desktop readers in a simple and consistent manner. Widespread adoption of feeds as the delivery mechanism means that, today, any feed reader can display content from a vast number of Web sites.

Widespread standardization and simplification of integration methods enables a looser and more dynamic coupling between service providers and consumers. This technology trend facilitates the democratization of processes in a very tangible way. Although many of the technologies are developer-oriented, they support the goal of giving control to end users because departmental application developers can exploit them on behalf of regular business people, outside the central authority of the IT department, and they establish the base patterns of integration that can ultimately be passed on to end users in more consumable, less technical, forms.

Building blocks

In this section, we explore some of the building blocks that are the technology enablers for an activity-centric approach to business processes.

URLs and REST-based systems

Many artifacts are involved in a business process, often spanning disparate IT systems. For example,

calendar entries are in the calendaring application and job candidates in the job vacancy application. Explicit references to these artifacts are necessary to allow for navigation among related artifacts spanning application boundaries.

The Web provides one such reference system, known as the Uniform Resource Locator (URL). A URL can be the address of a Web site, the location of a particular document, or the starting point of an application. While most readers may associate URLs with the Hypertext Transfer Protocol (HTTP) and Web browsers, it is worth noting that the URL naming scheme addresses many more resource types. Good examples demonstrating the flexibility of URLs include the “mailto:” and “im:” protocols, the latter of which launches an instant messaging session with the corresponding user.

A URL can provide a convenient and universally understood access point, or link, to the artifacts involved in a business process. An important design detail when building Web applications is that URLs can, and frequently do, contain state information. This practice renders the URL ineffective as an immutable object handle. Roy Fielding’s doctoral thesis on Representational State Transfer³⁰ (REST) underscores the importance of URLs that can be resolved to discrete resources and popularized the notion of REST-based services. REST-based services are the first prerequisite for weaving business processes together out of disparate applications. Only when application resources can be addressed externally can relationships and end-user navigation span application boundaries.

DOMs and extensible user interfaces

Web browsers, Hypertext Markup Language (HTML), and JavaScript** have attained widespread adoption. They have popularized and familiarized developers with user interface models based on the concept of a Document Object Model (DOM) tree.³¹ The dynamic nature of the DOM allows scripts to modify the user interface in response to end-user events or asynchronous responses received from a server. Web applications built along these lines are referred to as Asynchronous JavaScript and XML (AJAX) applications.³² AJAX application authors often include scripts that modify the DOM tree as part of the normal processing of the application. However, is it also possible to inject scripts into Web applications, thus adding capabilities not anticipated

by the original authors. The consumer market has seen some progress on this front and, while not

■ Democratization is about getting the best deal or about achieving greater access to buyers, employers, or other communities ■

ready for wide-scale deployment, script injection does offer a glimpse at future possibilities for extending user interfaces.

Greasemonkey³³ is an open-source plug-in for the Firefox** browser which allows user-authored scripts to be injected into Web pages just before the page is rendered in the browser. These scripts can alter the DOM tree and thus change the appearance of the Web page. Popular scripts block advertisements, change the appearance of the page, alter search results, or highlight words on the page.

It is also possible to use late injection of JavaScript to achieve integration among independent Web sites. The script called Book Burro³⁴ demonstrates this point. When this script is injected into Amazon.com, it locates the identifier of the book and then calls services at other online book stores to obtain competitive prices. With all the prices computed, it alters the DOM tree to display these prices as if they were part of the original Amazon.com Web page. In effect, the consumer has taken control of the business process and is able to see the price of any Amazon.com book at competitors’ sites. While we are not endorsing this particular script, it demonstrates a powerful way to weave independent Web applications together to add significant functionality. Put to productive business use, this capability means that applications can be supplemented with additional functions that better incorporate the application into a business process. Thus, modifiable DOM-based user interfaces are the second important enabling technology for lightweight integration.

Semantic tagging

The Book Burro example also highlights a significant difficulty that needs to be addressed. The injected script must parse the product page looking

for the ISBN number of the book and then must parse the page looking for the book's price. There are no semantics in the page that the script can use to discern meaning, so the script is limited to a set of sites whose pages it can programmatically parse. Furthermore, the script is brittle; changes in the Amazon.com page structure can break the script. What is needed is a way to tag the book's ISBN number and price so that it can be extracted more easily and reliably.

To fully realize their potential to integrate disparate applications, scripts cannot be tied to particular Web sites. We need generic scripts capable of understanding the semantic meaning of the data on the page. For example, Book Burro should be capable of acting upon any site that has an ISBN number and a price. Other generic scripts might enable collaborative features on any page that contains a user's e-mail address. This enables weaving together applications based on a common understanding of a person, telephone number, document purchase order, and other data elements. Any application should be able to register an interest in the types of data and then act upon them. In this way, it would be possible to seamlessly integrate Internet services with internal IT systems.

Semantic tagging is already at work on the Web. Google has recently studied the various HTML class attributes on pages that it indexes. For example, "price" is the fortieth most popular class attribute in use in the study.³⁵ Semantic tagging is available today, but lacks standardization and widespread use.

Two groups are addressing the problem of semantic tagging. Microformats.org is a federation of loosely coupled individual contributors, initially seeded by the aggregator site Technorati.com. XHTML2 Meta-information Attributes Module is an effort with similar goals, sponsored by W3C.³⁶ This paper does not weigh the benefits of one approach over the other. For the purposes of our argument it is enough to assume that a semantic tagging approach is emerging and that its implications are far-reaching.

Web of trust

In any business federation of applications, the constituent members must be able to establish identity and trust before exchanging sensitive data. Consider the following scenario in a customer-

service application. A customer support ticket is opened and assigned to an individual. This individual collaborates with others to resolve the issue, and the collaborations are supported by many tools including a chat room service. Access to the chat room should be restricted to the individuals authorized to view the support ticket, which is typically the group of individuals working to resolve the customer problem. Because the chat room and customer-service application are independent, we need a lightweight way to share the user authentication process. The simplest approach is to have the chat-room service delegate the authentication step to the customer-support application.

Several promising approaches to delegate authentication are emerging. Dubbed *Identity 2.0*,³⁷ this collection of technologies provides a lightweight means for sites to trust the credentials of an individual as provided by another site. Interestingly, all of these lightweight identity systems use URLs instead of passing and storing user credentials. The delegating system uses this URL to challenge the user. If the user is allowed to access it, then the identity is accepted. This approach is subtly and powerfully different from most authentication approaches. *Figure 3A* illustrates the flow for delegated authentication and authorization for the chat room example.

Delegated authentication and authorization is now being used at public Web sites to facilitate sharing of private data among hosted Web applications. An interesting example is between QOOP.com and Flickr.com. QOOP.com prints photograph books, and Flickr.com stores photographs. When attempting to print from QOOP.com, a user actually authenticates against Flickr.com. QOOP.com "trusts" Flickr.com to inform it of the user's identity. Additionally, users in Flickr.com must grant QOOP.com access to their photographs before it can start using them for printing. This is achieved as Flickr.com and QOOP.com have agreed on a protocol to exchange identity and authorization information.

While QOOP.com and Flickr.com have established mutual trust, the protocol is not yet widely adopted nor standardized through formal standards organizations. QOOP.com was explicitly coded to work with the Flickr.com model of authentication and authorization. Thus, federated trust and identity

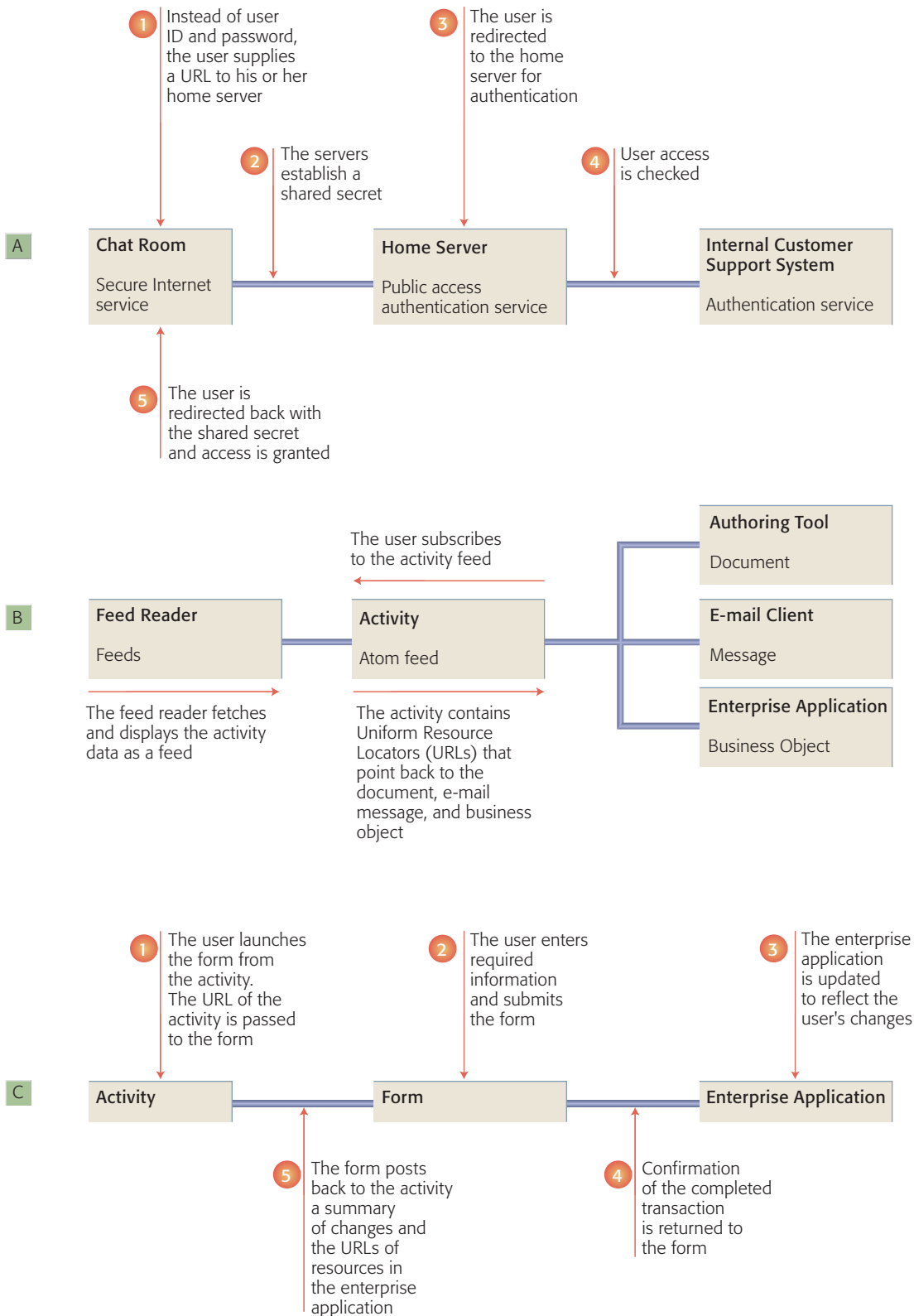


Figure 3

Activity building blocks: (A) delegated authentication and authorization for a chat room service, (B) an activity modeled as an Atom feed, and (C) using a form to loosely couple an activity to an enterprise application

systems are enabling technologies required for activity-centric, democratized business processes.

Aggregation

Activity-centric computing models underscore the importance of clustering work products, such as documents and discussion threads, by activity. The activity can be viewed as a hub that tracks the outstanding work, such as a customer support ticket or a software defect. Performing the actual work

■ As technical boundaries recede, activities will emerge as the dominant user-centered paradigm for working with distributed systems ■

takes place in supporting applications, which form the spokes branching out from this activity hub. As these supporting artifacts need to be associated with the activity, we need some kind of aggregation that provides a view into all the work taking place. How then can ancillary services be easily linked to an overall activity? Are there any standard means to post information to the activity hub so that a user can remain up to date with the changes occurring across the set of disparate systems?

We can use a simple URL addressing scheme to link artifacts to activities. This kind of integration can easily be enabled at the platform level; for example, by adding a button to a Web browser that extracts the address of the Web page being viewed and stores it in the activity. A wide spectrum of applications can be integrated, given that the application artifacts can be addressed by URLs.

To aggregate artifacts and coordinate updates, we turn to standards in the content syndication area. The most widely implemented means for publishing content and tracking updates is the syndication feed. Two standards exist for feeds, RSS and Atom, and both have widespread adoption. Atom has the advantage that, in addition to syndication, it also provides a means to remotely publish content by means of the Atom Publishing Protocol. Ancillary systems can use this protocol to inform the activity of their changes, and the user can track all the work occurring across the system through a feed pro-

duced by the activity system. An activity modeled as an Atom feed is illustrated in *Figure 3B*.

Forms

Many enterprise application systems now provide Web-service application programming interfaces (APIs). Advancements and standardization in forms technology makes it possible to write scripts for custom user interfaces that access and manipulate this enterprise data. XForms³⁸ is an emerging forms standard. XForms espouses a document-centric programming model in which the form is a self-contained entity, transmitted to and from middle-ware systems as an individual artifact. This artifact expresses the visual appearance of the form, its validation and data-processing rules, and the actual data payload.

By posting data back to the activity using the Atom Publishing Protocol, it is possible to build forms to understand the working context of a user's activity. The interaction flow is shown in *Figure 3C*. By using forms in this way, the activity system can support customized round-trips to enterprise applications and other services without modification of the enterprise application and without over-specialization of the activity service.

Addressing the pain points

Returning to the hiring use case laid out earlier, how then do the above technologies allow Lucy to weave her disparate systems together to better support her process?

Navigation across system boundaries

Let us assume that the job vacancy service can act as an activity hub and that the calendaring service acts as an ancillary service. When viewing a candidate in the job vacancy system, Lucy has the option of scheduling a meeting from this page, because the option has been injected into the user interface. The meeting is passed the URL of the candidate resource in the job vacancy system and thus "knows" the context within which it is operating. It can now make an Atom post back to this context, passing along its own URL. Bidirectional navigation from the calendar entry to the corresponding candidate in the job vacancy system is now available to Lucy.

Cut and paste

Cutting and pasting was originally necessary for two reasons. The first was to provide swifter navigation

between the resources used for a particular task. Lucy copied the resume and the candidate's phone number into the calendar entry so that she did not have to look for them when the calendaring system notified her of the appointment. With bidirectional linking in place, this copying is no longer necessary. The second reason that copying frequently occurs is to work around access control constraints. Lucy copies the resume out of the vacancy system and sends it to Frank because Frank does not have access to the vacancy system. By establishing a web of trust between Lucy's company and the external vacancy system, Frank could gain direct access to the resume document. The vacancy system would trust identities issued by Lucy and Frank's company without requiring Lucy and Frank to explicitly maintain user identity profiles in the vacancy system.

Lack of semantics

With the data in Lucy's applications semantically tagged, additional services that Lucy has available can now inject their capabilities. The telephone number displayed on the candidate's page in the job vacancy system is semantically understandable by ancillary applications. The VoIP service can now be injected directly into the job vacancy system as a "click to call" option near any telephone number. This service would be injected in a manner analogous to the Book Burro script inserting prices into the Amazon.com page.

Central challenge for activity-centric computing

We have proposed a set of enabling technologies that, when fully adopted, can empower end users to weave disparate applications together to support their business activities. We have further proposed that a hub-and-spoke approach is needed to tie work products back to an end-user activity, such as resolving a customer support ticket. The hub contains the activities and descriptions of the work to be completed, and the spokes contain ancillary applications that support the activities.

To close this discussion, we return to enterprise applications and consider the relationship between enterprise applications and activity-centric computing. It is surely undesirable to create entirely separate worlds for artful processes and industrialized processes, especially as the boundary between the two worlds is somewhat arbitrary. Hence, a central challenge for activity-centric computing is

how to bridge the world of artful processes supported by a decentralized infrastructure with highly structured and process-oriented applications that exist today. An enterprise application sometimes acts as a spoke in an artful process. Clearly however, enterprise applications are also activity hubs in their own right. Invoicing systems, hiring applications, and customer support systems are all good examples of systems that already contain activities necessary to drive the formal processes throughout a company. These applications have unique data semantics and workflows and provide their own specific business value; yet, they lack support for the "real work" that takes place to complete the tasks.

Can an infrastructure be built that augments these existing business applications with the ancillary capabilities needed in the real world? Can these applications also be participants in a broader activity service? This is an important area for further research.

CONCLUSION

In this paper, we presented arguments and user studies that identify a large array of valuable artful business processes. Such processes are not adequately supported by traditional enterprise applications, which require that processes be strongly formalized.

To enhance artful processes, companies need an alternative approach that gives business people greater direct influence over process definitions and enables them to weave together essential network services at the point of need. We call this the democratization of business processes.

The rise of loosely coupled information systems and compelling online services on the Web creates new expectations about information sharing, new methods of finding and navigating information and interacting with participating services, and new ways of establishing trust. These systems are completely decentralized, yet highly interconnected. With the rise of such capabilities, companies need to redesign and reassemble their business processes in a more flexible way. An activity-centric approach promises the ability to organize artful work productively while preserving user choice over the services employed.

Learning from the lightweight integration techniques emerging on the Web, we have identified several enabling technologies that can permit users to capture, share, and reuse work practices and link

■ We have proposed a set of enabling technologies that, when fully adopted, can empower end users to weave disparate applications together to support their business activities ■

them to the widest possible range of supporting services, while allowing for their decentralized design, development, and deployment.

FUTURE WORK

We have identified several areas for further research. Further empirical research is needed to better understand whether a long tail of business processes exists. Technically, a key challenge is to demonstrate an infrastructure that enables existing process applications to become hubs for activity-centric computing. In this paper we have identified building blocks, but much more work is needed to assemble an activity-centric system that fully exploits the technologies of URLs, REST APIs, DOMs, extensible user interfaces, semantic tagging, webs of trust, aggregation, and forms. In addition, it seems clear that an important challenge is how to provide end users with the right kinds of end-user programming tools to increase the range of process support that can be achieved.

ACKNOWLEDGMENTS

We thank the many customers referenced anonymously in this paper for explaining their needs and constraints lucidly and guiding us to design better solutions. We also thank Irene Greif of IBM Research for inviting us to submit this paper for publication. We gratefully acknowledge the many people in IBM who worked with us on these ideas: Miguel Estrada, Marty Moore, Scott Prager, Joe Russo, Andy Schirmer, Sami Shalabi, Doug Wilson, and Majie Zeller at Lotus; Li-Te Cheng, Steve Farrell, Werner Geyer, Dan Gruen, Tessa Lau, David Millen, Paul Moody, Tom Moran, Michael Muller, John

Patterson, Vova Soroka, and Eric Wilcox in IBM Research; and for their extraordinary management support, Ambuj Goyal, Craig Hayman, Ronnie Maffa, Chris Paul, and Mike Rhodin. For the enabling technology work, we thank Jonathan Feinberg for producing dogear, explaining it to us in detail, and allowing us to finally truly get REST; John Ponzo for his work on DOM programming, Charles De Saint-Aignan for proving out the semantic tagging approach with a person tag; Dave Wilson for helping explore lightweight identity systems; James Snell for all the Atom material; Dan Gurney, for help with hCard; and Vinod Seraphin for his work on AJAX clients and XML services. Finally, we acknowledge the influence of Sam Ruby, who pointed us to Atom and lightweight identity systems, and David Heinemeier Hansson, for showing how powerful really simple things can be.

*Trademark, service mark, or registered trademark of International Business Machines Corporation.

**Trademark, service mark, or registered trademark of Microsoft Corporation, EMC Corporation, salesforce.com, inc., Sun Microsystems, Inc., or Mozilla Foundation in the United States, other countries, or both.

CITED REFERENCES

1. S. Morris, *Integrating IBM Lotus Domino and SAP*, IBM Corporation, <http://www-128.ibm.com/developerworks/lotus/library/lei-sap/index.html>.
2. IBM Solutions for SAP® Software, IBM Corporation, <http://www-142.ibm.com/software/sw-lotus/lotus/general.nsf/wdocs/lotusforsap>.
3. WebSphere Portal, IBM Software, www.ibm.com/software/websphere/portal.
4. Duet: Bringing Together Business Productivity and Enterprise Applications, SAP, <http://www.sap.com/solutions/duet/index.epx>.
5. T. H. Davenport, "Putting the Enterprise into the Enterprise System," *Harvard Business Review* 76, No. 4, 121–133 (1998).
6. P. Moody, D. Gruen, M. J. Muller, J. Tang, and T. P. Moran, "Business activity patterns: A New Model for Collaborative Business Applications," *IBM Systems Journal* 45, No. 4, 683–694 (2006, this issue).
7. T. Davenport, *Thinking for a Living: How to Get Better Performance and Results from Knowledge Workers*, Harvard Business School Press, Boston, MA (2005), p. 22.
8. R. Austin and L. Devin, *Artful Making: What Managers Need to Know About How Artists Work*, Financial Times, Prentice-Hall, Inc., Englewood Cliffs, NJ (2003).
9. The Long Tail, Wikipedia, http://en.wikipedia.org/wiki/Long_Tail.
10. J. Krauss, "The long tail of software. Millions of Markets of Dozens," *Broopy: An entrepreneurship blog*, (March 9,

- 2005), http://bnoopy.typepad.com/bnoopy/2005/03/the_long_tail_o.html.
11. P. Larson, *The Long Tail of Process Automation*, Line56 (November 29, 2005), <http://www.line56.com/articles/default.asp?ArticleID=7124>.
 12. V. Bellotti, N. Ducheneaut, M. Howard, and I. Smith, "Taking Email to Task: The Design and Evaluation of a Task Management Centered Email Tool," *Proceedings of the ACM Conference on Human Factors in Computing Systems*, Fort Lauderdale, FL (2003), pp. 345–352.
 13. T. Davenport, "Saving IT's Soul: Human-Centered Information Management," *Harvard Business Review* **72**, No. 2, 119–131 (1994).
 14. E. Dyson, "Meta-mail: A Medium for Meaning," *Release 1.0* **22**, No. 6 (2004).
 15. D. R. Millen, M. J. Muller, W. Geyer, W. Wilcox, and B. Brownholtz, "Patterns of Media Use in an Activity-Centric Collaborative Environment," *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Portland, OR (2005), pp. 879–888.
 16. M. J. Muller, W. Geyer, B. Brownholtz, E. Wilcox, and D. R. Millen, One Hundred Days in an Activity-Centric Collaboration Environment Based on Shared Objects," *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Vienna, Austria (2004), pp. 375–382.
 17. S. L. Rohall and D. Gruen, "ReMail: A Reinvented Email Prototype," Technical Report 02-13 (2002), IBM Watson Research Center, <http://domino.watson.ibm.com/cambridge/research.nsf/c9ef590d6d00291a85257141004a5c19/dbc7c3853dca5ed785256ca7006d90a1?OpenDocument>.
 18. B. A. Nardi, Editor, *Context and Consciousness: Activity Theory and Human-Computer Interaction*, MIT Press, Cambridge, MA (1996).
 19. D. A. Norman, *The Invisible Computer*, MIT Press, Cambridge, MA (1998), pp. 82–85.
 20. J. E. Bardram, J. Bunde-Pedersen, and M. Soegaard, "Support for Activity-Based Computing in a Personal Computing Operating System," *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Montreal, Canada (2006), pp. 211–220.
 21. W. Geyer, M. J. Muller, M. T. Moore, E. Wilcox, L.-T. Cheng, B. Brownholtz, C. Hill, and D. R. Millen, "Activity Explorer: Activity-centric Collaboration from Research to Product," *IBM Systems Journal* **45**, No. 4, 713–738 (2006, this issue).
 22. T. P. Moran, "Unified Activity Management: Explicitly Representing Activity in Work-Support Systems," *Proceedings of the European Conference on Computer Supported Cooperative Work*, Workshop on Activity: From a Theoretical to a Computational Construct, Paris, France (2005), <http://www.daimi.au.dk/~bardram/ecscw2005/papers/moran.pdf>.
 23. A. Cozzi, S. Farrell, T. Lau, B. A. Smith, C. Drews, J. Lin, B. Stachel, and T. P. Moran, "Activity Management as a Web Service," *IBM Systems Journal* **45**, No. 4, 695–712 (2006, this issue).
 24. T. O'Reilly, *What is Web 2.0? Design Patterns and Business Models for the Next Generation of Software*, O'Reilly, <http://www.oreillynet.com/lpt/a/6228>.
 25. S. L. Kogan and M. J. Muller, "Ethnographic Study of Collaborative Knowledge Work," *IBM Systems Journal* **45**, No. 4, 759–772 (2006, this issue).
 26. A. Bernstein, "How Can Cooperative Work Tools Support Dynamic Group Processes? Bridging the Specificity Frontier," *Proceedings of the ACM Conference on Computer Supported Cooperative Work*, Philadelphia, PA (2000), pp. 279–288.
 27. S. Whittaker and C. Sidner, "Email Overload: Exploring Personal Information Management of Email," *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Common Ground*, Vancouver, British Columbia (1996), pp. 276–283.
 28. "What is Wiki," <http://wiki.org/wiki.cgi?WhatIsWiki>.
 29. B. Hammersley, *Developing Feeds with RSS and Atom*, O'Reilly Media, Inc., Sebastopol, CA (2005).
 30. R. T. Fielding, *Architectural Styles and the Design of Network-Based Software Architectures*, Ph.D. dissertation, University of California-Irvine, CA (2000), <http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>.
 31. D. Flanagan, *JavaScript: The Definitive Guide*, Fourth Edition, O'Reilly Media, Inc., Sebastopol, CA (2001).
 32. J. J. Garrett, *Ajax: A New Approach to Web Applications*, Adaptive Path LLC, <http://www.adaptivepath.com/publications/essays/archives/000385.php>.
 33. M. Pilgrim, *Greasemonkey Hacks: Tips & Tools for Remixing the Web with Firefox*, O'Reilly Media, Inc., Sebastopol, CA (2006).
 34. Book Burro, <http://bookburro.org/>.
 35. "Web Authoring Statistics," <http://code.google.com/webstats/index.html>.
 36. XHTML Metainformation Attributes Module, W3C Working Draft, <http://www.w3.org/TR/2004/WD-xhtml2-20040722/mod-metaAttributes.html>.
 37. D. Hardt, *Web 2.0 High Order Bit—Identity 2.0*, Sxip Identity, http://www.identity20.com/media/WEB2_2005/.
 38. T. V. Raman, *XForms: XML Powered Web Forms*, Addison-Wesley Professional, Boston, MA (2004).

Accepted for publication April 17, 2006.
Published online October 20, 2006.

Charles Hill

IBM Software Group, 5 Technology Park Drive, Westford, Massachusetts 01886 (charlesh@us.ibm.com). Mr. Hill is a Senior Technical Staff Member and a user-experience designer working on collaboration and portal software. He has a Ph.D. degree in mechanical engineering from Imperial College, London, and an M.Des.RCA in industrial-design engineering from the Royal College of Art, London. He was the user-experience design lead for Lotus QuickPlace®. Mr. Hill's current focus is on social software and activity-centric computing.

Robert Yates

IBM Software Group, 5 Technology Park Drive, Westford, MA 01886 (robert_yates@us.ibm.com). Mr. Yates is a senior software engineer on the IBM Workplace team. His present focus is applying open standards, such as Atom, to both existing and emerging products. He has an M.A. degree in mathematics from Cambridge University, England.

Carol Jones

IBM Software Group, 5 Technology Park Drive, Westford, MA 01886 (jcarol@us.ibm.com). Ms. Jones is an IBM Fellow and software architect working on portal and collaboration software. She has a B.S. degree from Augusta College and an M.S. degree in computer science from North Carolina State University. Her major area of interest is in rich Internet applications and their associated programming models. She has led engineering efforts on WebSphere Portal and IBM Workplace Collaboration Services and early versions of WebSphere Studio and WebSphere Application Server. Her current focus is on social software and emerging technologies related to forms, feeds, and real-time collaboration.

Sandra L. Kogan

IBM Software Group, One Rogers Street, Cambridge, MA 02142 (sandra_kogan@us.ibm.com). Ms. Kogan is a user researcher and user-experience designer in Workplace, Portals, and Collaboration Software. She has a B.S. degree from Concordia University and an M.S. degree in computer science and human computer interaction from the University of Massachusetts at Lowell. She has led several design strategy initiatives for Lotus Development. Ms. Kogan's current focus is on empirically derived user models and quantitative analysis. ■