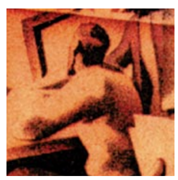
Information systems exist within an organizational web that links people, tools, and products with work processes. PDIs explicitly integrate with this web and offer clear advantages over more traditional intranets in organizational process redesign.

PROCESS-DRIVEN INTRANETS:

Life-Cycle Support for Process Reengineering

WALT SCACCHI AND JOHN NOLL
University of Southern California



Although business process reengineering has been touted throughout the decade as a way to streamline organizations and improve productivity, three out of four BPR efforts reportedly fail outright, fall short of their performance goals, or are otherwise botched.[1] We see primarily two reasons for this. First, most tools and methods for BPR concentrate on upstream process-design activities like modeling and redesign, almost to the exclusion of mid-stream transitional activities and downstream activities that incorporate the implemented processes into work routines. Second, most efforts do not account for the existing "organizational web"—the social *and* technological infrastructure that spans not only computer-based information systems, but also organizational charts, office and building configurations, telecommunications systems, transportation and distribution channels, resource-to-product processes, and workflows.[2]

Designers of intranet- or Web-based networked information systems are increasingly expected to model the organizational web so that an information system can successfully support it.[3] Unfortunately, understanding how people implement, routinize, and incrementally evolve their work processes continues to take a back seat to simpler concerns such as selecting which browsers, servers, and associated resources to use. Thus, the likelihood remains of low-performing organizational

1089-7801/97/$10.00 ©1997 IEEE

processes and marginally usable information systems.[4]

We believe that corporate intranets may offer a successful way to implement more effective systems. In the approach we describe here, *process-driven intranets* support a computer-based representation of an organizational web.[5] This differs from traditional intranets, which have no explicit representation of the processes they support. By directly supporting the explicit modeling of the organizational processes implemented on a wide-area network, PDIs shift the BPR focus from the traditional upstream activities of process modeling and (re)design to midstream and downstream activities, such as facilitating feedback for continuous process improvement.

Our use of PDIs evolved from the requirements of a recent research case study we conducted for the US Office of Naval Research. The study investigated how to formalize and redesign processes for managing research grants, while also adopting a "corporate-wide" intranet. The intranet, which would be used by ONR headquarters and field offices nationwide, would support internal processes related to the funding and tracking of research grants.

In earlier research, we had established a methodology for the life-cycle engineering of an organizational web.[6] Our challenge became how to best redesign ONR's processes so that a wide-area intranet would be central to the solution. In describing our PDI-centered design approach, we draw examples from this case study. ONR estimates operational savings of $10 to $15 million per year with the proposed redesigned processes as well as reductions in the cycle time for processing research grants by a factor of 10. We believe organizations with similar requirements can achieve similar savings.

CHARACTERISTICS OF A PDI

An intranet is a TCP/IP network that provides a foundation for sharing information in a corporate setting. It supports subsystems for applications such as network management, groupware, and conferencing; and it provides access through a Web browser to corporate information in networked file systems and database management systems. However, business processes that access and update information are buried in programs and dispersed across execution scripts and Web pages. As a result, they can appear disjointed, unstructured, or opaque to many users. Furthermore, changing these business processes often requires programming skill, which disables rather than empowers many process stakeholders. For these reasons, CGI programming languages, document markup languages, or database access languages are poor choices for describing business processes. They do not help people describe, understand, try out, and revise the core processes they routinely perform.

PDIs add subsystems that support configurational (re)design, integration, and continuous improvement of organizational processes. PDIs can be implemented in different ways, for example, through workflow and database management systems, network operating environments, or ad hoc Web-based scripting. In our case, they *require* a formal business-process model that can be represented as a semantic hypertext graph. PDIs can use this representation to interpret a process model so that, for example, links to Web pages or frames are automatically updated where process actions occur. An external link repository and server manages the links. This contrasts with conventional intranets that use handcrafted links and CGI programs to deliver business transaction steps, which require tedious editing and reprogramming to accommodate process changes.

The PDI we developed for ONR was based on concepts, tools, and techniques developed at the University of Southern California's ATRIUM Laboratory to support a multi-year research project investigating ways to reengineer financial operations for acquisition and procurement. Central to this work has been the development of organizational process architectures (OPA) in a variety of operations: large-scale software engineering, new product development, supply chain logistics, corporate finance, and military procurement. OPAs model the processes, products, people (roles and work groups), information infrastructure, and tools central to an organization's routine work operations. They constitute a persistent information resource that can be tailored for reuse throughout an organization or, with a composite OPA, throughout an industry.[7]

Process Modeling Language

We specify an OPA using the PML process modeling language,[8] a notation we developed for defining a minimal set of objects, attributes, and relations common to processes, products, user roles, and tool and application invocations. PML specifies process steps within common control-flow structures, including concurrency. Within each process step it specifies

- server-side data resources or products (input or output),
- client-side tools or forms used, and
- user roles.

It also specifies a role-based user interface. Further, it can specify pre- and post-conditions that act as constraints on input and as goals for output values for objects and products accessed or updated per step.

Figure 1 on the next page is a sample PML description for a proposal submission subprocess within ONR's Preaward process. This process has two top-level steps, submit_proposal and submit_supporting_documents. In the first step, an HTML-based form captures a proposal file from a principal investigator and a proposal.id is assigned. The second step comprises two concurrent branching actions. The first action requires an e-mail address (captured in an earlier subprocess, which is not shown) to automatically invoke a mail tool, which then sends the designated Web page to the e-mail address. No person or agent is needed to perform this action. The second action involves the submission of a budget file, in a manner similar to

```
process proposal_submit {
    action submit_proposal {
            requires { "proposal" }
            tool {"http://gilligan.usc.edu:3280/process/onr/forms/submit-proposal.html"}
            provides { "proposal.id" }
            agent { "PrincipalInvestigator" }
    }
    branch submit_supporting_docs {
            action submit_certs {
                    requires { "email-address-PI-BusOffice" }
                    tool {sendmail: "http://www.onr.navy.mil/sci%%5Ftech/special/onrtpr02.html"}
                    provides { "certification-info-sent" }
            }
            action submit_budget {
                    requires { "budget" }
                    tool {"http://gilligan.usc.edu:3280/process/onr/forms/create-budget.html"}
                    provides { "budget.id" }
                    agent { "PrincipalInvestigator" }
            }
    }
}
```

Figure 1. PML excerpt for a subprocess in submitting proposals to the Office of Naval Research.

submitting a proposal, which produces an assigned budget_id.

Although not shown, all required, provided, or agent entities are declared and assigned object types, similar to object-based modeling schema. Process entities may be related (or linked) by different types of dependencies, such as precedence, availability, and decomposition. Similarly, tools are specified via local/remote command scripts, such as URLs and operating system shell scripts. Details about our tool integration and activation scheme are available elsewhere.[5,8]

Compiler

We also developed a PML compiler to build the semantic representation of processes from the OPA into programs and frame-embedded links for user interaction through a browser. The compiler can generate executable programs in Tcl or Java, for example. The programs update frames in the user's browser with links to planned, pending, available, or completed processes' steps. The compiler can alternatively generate Web page content in JavaScript for process prototypes, depending on the code generator configured into it. The generated process execution code serves as input to the runtime environment—a set of services that act as an "operating system" to execute networked processes (workflow automation) and to access and update files or databases. This network operating environment can then be distributed and integrated with compatibly configured intranet/Web servers for intranet or Web-based delivery, access, and navigation-based process enactment.

This ability to generate PDIs directly from PML specifications allows an organization to rapidly implement, distribute, evaluate, and refine incremental process changes. Further, it lets users develop and implement a composite, intralinked OPA that connects processes, products, and people across a network of organizations on the Internet.

PDI-CENTERED REDESIGN

The systematic design of an OPA is a collaborative effort that requires incremental development, iterative user validation and refinement, rapid process visualization and prototyping, and the cooperative redesign of ad hoc process task instances and models. PDIs are an effective implementation mechanism for this work because they let an organization easily manipulate a formal representation of its organizational web.

In this section we review the activities associated with the design or redesign of OPAs: upstream design, midstream implementation, and downstream routinization.[6] PDIs contribute to each set of activities in different ways. Their principal value in the ONR case study was to support process design and implementation, making it easier to involve process stakeholders and get their input.

Upstream Process Design Activities

Traditional BPR tools tend to limit their focus to activities in this phase: process metamodeling, elicitation and modeling, analysis and simulation, and redesign. These activities help in understanding the as-is situation and identifying possible to-be alternatives. In this phase, PDIs enable the corporate-wide sharing of OPA models, as well as the browsing and capture of feedback from analysis, simulation, and redesign activities. This in turn provides a channel for improving how organizational processes are modeled and redesigned. At this stage, complex organizational activities are quickly redesigned but with less regard for how the redesigned alternatives will be implemented or put into routine practice.

Metamodeling. This activity constructs and refines a concept vocabulary and logic—an ontology—for representing process family instances. This, in turn, requires understanding the domain, context, and organizational web for the processes at hand. In the ONR project, we used an ontology derived from similar studies in modeling and formalizing OPAs for related domains.

Elicitation and modeling. This activity focuses on capturing informal descriptions of processes within the organizational web and then converting them to formal process models and OPAs. Since the models and OPAs take the form of directed attributed graphs with typed links and nodes, they can be represented as semantic hypertexts that can be distributed, configured, browsed, navigated, and updated over a PDI or the Internet.[5,8] One of the major tasks in this activity is to understand existing processes before suggesting process alternatives.

For example, in the ONR effort, we involved personnel from all levels of the organization. We needed to educate them about our motives and methods, and they needed to educate us on the generic, circumstantial, and problematic variants of their grants-management processes. We typically performed two or three iterations of process elicitation, analysis, visualization, and refinement, particularly with key process users or subject-matter experts. We did follow-up validations and refinements with personnel not necessarily involved in these iterations. We also established a set of agreements between the research team and ONR personnel:

- The research team would identify more than one opportunity to optimize the redesign of work processes, information flow, and information integration.
- Effort would be directed at improving personnel effectiveness and workflow without increasing anyone's workload.
- The redesign would create no new personnel positions.
- Processes would be developed in three forms: as-is (legacy form), to-be (alternative process architectures), and transition (steps taken in 30-day increments to transform work patterns and processes from as-is to to-be).
- All intermediate and in-progress results would be posted on the USC ATRIUM Web site so that ONR personnel could access this information at their discretion and provide questions or feedback.
- ONR personnel would select the improvement alternatives to be implemented.
- Any improvements had to be self-motivating or otherwise enable local organizational incentives to increase the likelihood of their successful implementation and incorporation into routine.

As a result of these agreements, we captured models of four grants-management processes in the OPA, coded them into PML, and posted the PML and associated process visualizations derived from the PML on an ATRIUM Web server.

Static analysis and simulation. A process model's static properties can be checked for consistency, completeness, internal correctness, and traceability. Processes modeled in a language-based notation such as PML can use a language compiler or interpreter to analyze static properties. In our experience, this analysis is perhaps the best source of high-value, short-term payoffs during process redesign because it clarifies where processes are and are not understood. PDIs can also provide simple mechanisms for producing intranet- or Web-based management reports or presentation materials suitable for routine use.

Simulation symbolically executes process models to determine the path and flow of intermediate state transitions in ways that can be made persistent and thus be replayed, queried, dynamically analyzed, and reconfigured into multiple alternative scenarios. We have used two kinds of process simulations:[6] *Knowledge-based* simulation enables capabilities such as query-driven and reverse simulations; *discrete-event* simulation offers visual simulations of processes, which make it easy to discover dynamic process bottlenecks and optimization opportunities.

Redesign. Process redesign focuses on reorganizing and transforming the structure of relationships within a modeled process and OPA to address weaknesses, such as too many steps, handoffs, or participants. Process redesign starts by applying a set of analysis routines and procedures we have developed for "measuring" and classifying process flow graphs or subgraph patterns. These measures help to surface process inefficiencies. Redesign heuristics, selected from a growing base of experience and published studies of successful tactics, are then applied to reveal efficient, transformed process representations.[10]

Figure 2 on the next page shows the structure of process flow steps for Funding Approval, a subprocess of ONR's research grants Preaward process. The levels of nesting indicate the chain of review and rework loops, which reveals potential weaknesses in the process, such as the apparent lack of parallelism in the process flow. This in turn suggests a process redesign heuristic that might call for joint collaborative reviews. With joint reviews, ONR could quickly separate the funding requests that are ready for approval or denial from those that require further attention or rework.

By using visualization aids like the graph in Figure 2 with their corresponding PML process descriptions, we identified several process inefficiencies for potential redesigns. For example,

- Consolidate and automate the *manual step sequence*.
- Parallelize *linear step sequences*.
- Collapse the *many work review steps* into joint collaborative review sessions.
- Move *many data validation steps* to earlier in the process.
- Use rule-based systems to automatically configure the *manual assembly of compound documents*.
- Eliminate the *extensive use of paper documents*, instead using electronic versions that can be stored in conventional or hypertext databases, accessible over wide-area networks.
- Use the intranet to consolidate *islands of automation*, integrating data transactions and the invocation of software

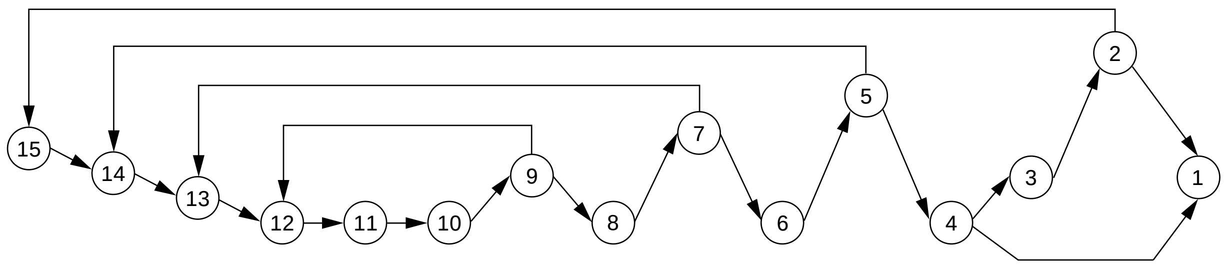


Figure 2. Chain of review approvals in the **Funding Approval** subprocess of ONR's research grants management. Graphs like this make it easier to visualize process weaknesses such as, in this case, a lack of parallelism in reviews.

applications or tools through client-side Web browsers.

- Use intranet-based process enactment mechanisms to better visualize *workflow across multiple organizational units* across the corporate network.

ONR is implementing all these alternatives, except the rule-based document-configuration system. The alternatives have reduced the steps in ONR's Grant Administration process from 31 to one, and for the Grant Award process from 34 to five (and possibly three). Similarly, using electronic grant documents eliminates a multimillion-dollar operational cost associated with the legacy processes that handle conventional paper-based research proposals, when considered with other process redesign alternatives.

Midstream Implementation Activities

Once processes and organizational webs have been redesigned, the next objective is to implement the redesigned alternatives. Planning and executing a transition process is central to the success of implementation. Like the implementation, the transition must engage process stakeholders in a way that lets them share ownership in the process redesign. PDIs help make process redesign alternatives easy to prototype and execute across a corporate network. Participants can thus see and experiment with process alternatives before committing to their final implementation and routinization. The aim of activities during this phase is to support an evolutionary refinement and implementation of the redesigned processes.

Visualization. Process stakeholders and senior managers need to see the structure and flow of process redesign. Graphical views frequently trigger an intuitive understanding of how a process works or doesn't work. Process stakeholders are often surprised by the difference in as-is and to-be process views, which helps to invoke additional feedback on process alternatives.

In Figure 2, for example, the visual display of a process structure helped its users and owners recognize that performing this process without getting caught in a rework loop was highly unlikely. In other words, if the probability of getting through the decision point in each loop is 0.8, with four nested loops, the probability of getting through the decision points of all four loops without performing a rework iteration is approximately 0.4 ($0.8 \times 0.8 \times 0.8 \times 0.8$), meaning most passes through the process will require rework. This process visualization helped the participants quickly recognize the previously unknown pathology of the as-is process structure. It was then easy to motivate process stakeholders to change the process structure.

Process visualizations can also be shared across an intranet to make it easier to support feedback on process redesign alternatives among geographically distant stakeholders.

Prototyping. One important way to visualize the dynamics of a process is through prototyping. Prototypes let users preview what will be seen and how it will work when processes are implemented and put into routine operation on an intranet. PDIs let stakeholders access and incrementally traverse prototypes at their desktops through a Web browser, providing a low-cost platform for refining organizational process redesigns.

Figure 3 shows one of many steps in the Grants Management prototype. In this step, which is part of the Preaward process, the principal investigator submits a research project proposal as an HTML document to ONR for consideration. The investigator then enters the data ONR needs to log and track the proposal. This includes identifying the proposal file to be uploaded or transferred to ONR. When the user clicks "done," the transfer is initiated. The user can then select Next Task, through which an electronic budget that accompanies the proposal is submitted. The frame in the bottom right of the figure displays the history of steps completed, serving to record the in-progress status of each step through the process. In this way, multiple concurrent process instances may be active at any time, and users can select the one they want to work on by selecting from available tasks for their selected role and user-ID. Basic access control and authentication mechanisms are applied along the way. Depending on auditing requirements, other attributes, such as date and time of step completion and PC network (IP) address, can be recorded and later replayed or analyzed for process improvement opportunities.[6]

Transition planning and administration. These activities involve assigning and scheduling specified users, tools, and process data objects to a plan for implementing the refined process alternative. PDIs serve as a vehicle for distributing and monitoring progress in fulfilling the plan.

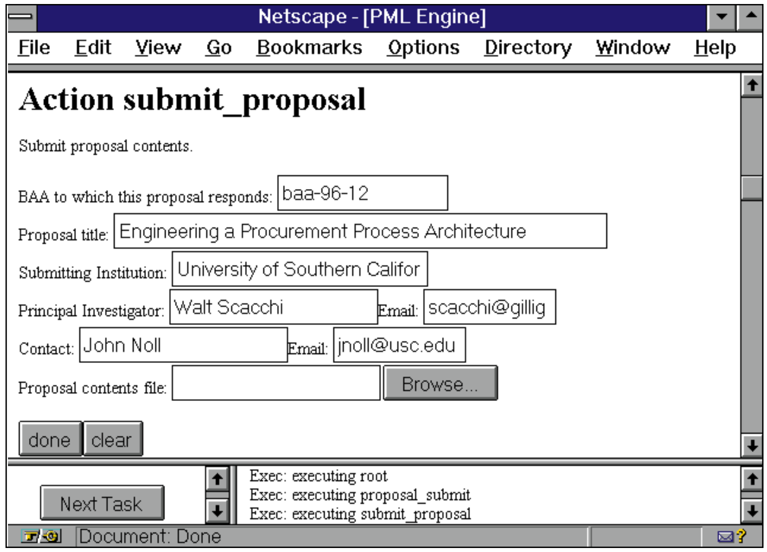


Figure 3. View displaying a process step in a prototype of the **Grants Management** process, which is displayed through a PDI-based browser.

Integration. On the technological side, systems integration entails encapsulating or wrapping selected legacy and new information systems, client-side tools, server-side repositories, and data objects that can be invoked or manipulated when enacting a process instance. Integration provides a computational workspace that binds user, organizational role, task, tools, and input and output resources into "semantic units of work." Scripting languages can be used, as can middleware interface mechanisms, to simplify the integration of distributed objects and services.[8,9]

Addressing the organizational integration of processes and systems may pose technical challenges, but, when those challenges are effectively addressed, there is a high payoff. For example, in the ONR case study, we identified a major integration problem in grants management. Many at ONR never knew with certainty the balance and distribution of funds as obligations, invoices, or disbursements. Obligations are budget allocation preferences that could be altered by top-level decision-makers at different times or under different conditions, invoices denote bills submitted for payment (subject to approval), and disbursements represent funds spent. The time between when ONR agreed to fund a project (obligation) and when it transmitted and authorized acceptance of invoices was typically seven to 10 weeks. The time from obligation to disbursement was often many months or longer. During these times, uncertainty about the balance of funds across each category or for different subsets of research grant awards led to conflicts and organizational inefficiency. Reducing these lead times to days (or better yet, hours) would alleviate these problems.

Complicating expenditure management even more was the structure of the information flow. The needed information was distributed across three distinct information systems and across four grants-management processes. Each system had its own administrative authority, organizational location, database, data model, and data format. We proposed a more integrated view, which Table 1 on the next page shows. Each of the last three columns represents data managed by a separate database management system and administrative authority. Integrating these databases into a single global database would be too expensive and time-consuming. A more realistic alternative was to use a form-based tool to provide read-only views to a data cache that periodically queried, downloaded, reformatted, and sorted (using award and request numbers as indices) the funding data from each database.

ONR personnel indicated that having on-demand access to an integrated view of funding actions over an intranet would directly address senior management's critical decision-making needs in ways that had not been available. Stressful ad hoc activities and workarounds for expenditure management could be eliminated. We also found that similar integrated data views could be created to display the dates associated with the performance of specified acquisition process steps, which would meet status reporting needs. Linking these tables with common PC tools like WordPerfect and Microsoft Access would automate and streamline management report preparation even more because the required reports and presentation graphics could then be produced on demand in redesigned grants management processes.

Environment generation. In this activity, a process model is automatically transformed into a PDI that selectively presents prototyped or integrated information systems to end users for process enactment. Our PML compiler and runtime environment make an effective application generator—in this case for generating PDIs that implement partially and fully specified models of complex organizational processes, user roles, tool or application activation, and input/output data mappings.

Downstream Routinization Activities

Once implemented, organizational processes must evolve into the background of the work setting so that competent process users are not constantly reminded what to do, but can rather

Table 1. A proposed integrated view of ONR funds across acquisition processes.

ONR Award Number	Request Number	Obligated Funding	Amount Invoiced	Amount Disbursed
N00014-95-1-0986	96PR03062-00	$100,000.00	$100,000.00	$100,000.00
N00014-94-2-0011	96PR04842-02	$606,317.00	$500,000.00	$350,000.00
N00014-95-2-0014	96PR05343-00	$381,562.00	$81,562.00	$81,562.00
…	…	…	…	…

seek help when something unexpected happens. In routinized processes, process structure and performance details often become tacit knowledge for those who execute them—until something occurs to disrupt the workflow. For core organizational processes, such occurrences are surprisingly frequent. Accordingly, processes must be continually adjusted and refined. PDIs provide an information infrastructure for the activities that support downstream routinization of work.

Instantiation and enactment. In this activity, we perform the modeled process using the PDI. Process enactment guides or enforces users or user roles to apply the process as specified. It looks and operates the same as a process prototype, except that now live applications and data are activated and accessed over the corporate intranet. Our experience is that guidance helps unfamiliar users as well as skilled users facing newly updated or improved processes. Further, PDI workspaces for process enactment are effective in tracking the state of progress made in each concurrent process instance.

Process monitoring, recording, and replay. These activities follow from the ability to collect and measure process enactment data. The history capture frame in Figure 3 is an example of how we might capture performance data for continuous process improvement, particularly when time stamps and other data are also recorded.[9] These activities also aid in documenting what process steps actually occurred in what order. Overall, we have found that such data will be most effective when collected and used primarily by process users, acting individually or in quality circles, rather than as personnel performance measures reported to senior management.

Articulation, evolution, and asset management. These activities shape the ongoing evolution of routinized processes. *Articulation* deals with the diagnosis, repair, and rescheduling of process enactments that have unexpectedly broken down or failed because of some unmet resource requirement.[9] While these problems have been investigated in traditional client-server settings, how to handle them in an intranet or wide-area network environment is still an open problem.

The formalized models and OPA representations for PDIs support process *evolution* for continuous improvement. In the computing world and in complex organizational webs, change is constant. We should expect to incrementally and iteratively enhance, migrate, or reengineer process models, OPAs, and process life-cycle activities to more effectively meet emerging user requirements, and to capitalize on the opportunities new tools and techniques introduce.

Finally, process models and OPAs require *asset management*. This stems in part from the need to systematically support process evolution and continuous improvement. Moreover, in large multisite organizations such as ONR, process assets should be centrally managed, but available on a distributed, wide-area basis. Furthermore, other organizations within an industry or within the federal government may benefit from the reuse, tailoring, or reenactment histories to help move their organizations toward reengineering their in-house processes using PDIs. Thus, PDIs will benefit from the use of database or hypertext repositories for managing and sharing PDI assets.

LESSONS LEARNED

In applying PDIs in the ONR case study, we learned several important lessons. Here we offer a sample of recommendations to those considering experimentation with PDIs.

Take time to understand existing processes. Most existing processes are neither well-defined nor well-understood. Yet, we have found that most process modeling or redesign efforts want to jump to new process alternatives without baselining the existing processes. Process redesign should be structured to encourage sharing and distributing information on proposed process improvements, and PDIs support this.

Make reengineering a collaborative, team-based endeavor. Having intranet tools like PDIs and techniques that support the participation of process stakeholders can help make redesign more successful. Those we developed address how to support process design, implementation, and routinization activities as well as collaboration concerns that other approaches ignore.

Consider multiple redesign alternatives. Some savings aren't possible unless you implement a set of interrelated alternatives.

Recognize that not all processes are suitable for reengineering. PDIs and OPAs may not be suitable for processes that are not large or routinely practiced. Understanding whether and how PDIs and OPAs might be applied to such processes requires additional research.

CONCLUSION

Redesigning an organization's web to more effectively streamline its processes and more efficiently use emerging information technologies is not a simple task. It cannot be finessed by installing a new technology that automates existing processes. A successful redesign requires understanding existing and alternative future forms of how people perform their work processes using available information technology, and engaging these people in transitioning the overall structure of their work.

We believe PDIs represent a new kind of intranet technology that provides a higher level of abstraction and interface for interconnecting distributed organizations through cross-integrated processes. The elements of our approach—PDIs, OPAs, and the design activities described—may form part of the foundation for a new generation of organizations that use process-driven, wide-area networked information systems in their core business processes. ONR is only one of the 10 or so federal agencies in the US government that manage and fund research grants. Collectively these grants total $15 billion annually. Improving and streamlining these processes benefits participants in the academic and industrial research communities. Large corporations also spend millions of dollars on back-office operations associated with funding, procuring, or purchasing goods and services from hundreds of trading partners. Thus, there are a substantial number of organizational settings that are amenable to streamlining and cost savings through reengineering with PDIs.

Our results are based on limited exploration and experimentation. New intranet and Web technologies will also enable others to experiment or apply PDIs, OPAs, and the process life cycle in their organizational web. Process-driven extranets that enable partnering business organizations to interconnect their external commerce processes with their internal operational processes seem to be a ripe area for investigation and commercial exploration. New BPR activities such as process prototyping, process environment generation, and automated process execution using intranets may help in implementing the redesign of business processes as intranet-based work environments. PDIs support incremental, continuous process improvements and enable rapid deployment of these improvements across multiple sites connected through a corporate intranet. ■

ACKNOWLEDGMENTS

We thank the many people at ONR who contributed to this study, particularly Jim Carbonara, Dennis Padilla, Jay Miller, Diana Nichols, Frank O'Day, Chuck Paoletti, John Starcher, Ann Stuart, Jim Valvocin, and Ralph Wachter. Cedric Knight of New Directions Technologies provided valuable field support and consulting services. Former ATRIUM Laboratory researchers Peiwei Mi, now at Intelligent Systems Technology, and Mark Nissen, now at the Naval Postgraduate School, contributed to the concepts and methods used in this investigation.

This research was supported by ONR grant N00014-94-1-0889. None of the material in this report should be construed as a statement of policy, procedure, or endorsement by ONR, the US Navy, or any other US government agency.

REFERENCES

1. B. Bashein, L.M. Markus, and P. Riley, "Preconditions for BPR Success and How to Prevent Failures," *Information Systems Management*, Vol. 11, No. 2, 1994, pp. 7-13.
2. P. Mi and W. Scacchi, "A Meta-Model for Formulating Knowledge-Based Models of Software Development," *Decision Support Systems*, Vol. 17, No. 3, 1996, pp. 313-330, http://www.usc.edu/dept/ATRIUM/Papers/Process_Meta_Model.ps.
3. J. Tenenbaum, T. Chowdhry, and K. Hughes, "Eco System: An Internet Commerce Architecture," *Computer*, May 1997, pp. 48-55.
4. R. Kling, "Organizational Analysis in Computer Science," *The Information Society*, Vol. 9, No. 2, 1993, pp. 71-87.
5. J. Noll and W. Scacchi, "Supporting Distributed Configuration Management in Virtual Enterprises," in *Software Configuration Management*, R. Conradi, ed., Lecture Notes in Computer Science, Vol. 1235, Springer-Verlag, New York, 1997, pp. 142-160, http://www.usc.edu/dept/ATRIUM/Papers/DHT-SCM7.ps.
6. W. Scacchi and P. Mi, "Process Life Cycle Engineering: A Knowledge-Based Approach and Environment," *Intelligent Systems in Accounting, Finance, and Management*, Vol. 6, 1997, pp.83-107 , http://www.usc.edu/dept/ATRIUM/Papers/Process_Life_Cycle.html.
7. F. Leymann and W. Altenhuber, "Managing Business Processes as an Information Resource," *IBM Systems J.*, Vol. 33, No. 2,:pp. 326-348.
8. J. Noll and W. Scacchi, "Supporting Software Development Projects in Virtual Enterprise," pending publication, original version on http://www.usc.edu/dept/ATRIUM/Papers/DHT-VE97.html.
9. J. Noll and W. Scacchi, "Integrating Heterogeneous Information Repositories: A Distributed Hypertext Approach," *Computer*, Dec. 1991, pp. 38-45, http://www.usc.edu/dept/ATRIUM/Papers/Distributed_Hypertext.ps
10. M. Nissen, "Knowledge-Based Organizational Process Redesign: Using Process Flow Measures to Transform Procurement," PhD dissertation, IOM Dept., USC School of Business Administration, Los Angeles, 1996, http://dubhe.cc.nps.navy.mil/~menissen/dissabst.

Walt Scacchi is a research professor and founding director of the ATRIUM Laboratory at the University of Southern California. He has directed more than a dozen research projects in process engineering over the past 10 years for both corporate and government sponsors. Scacchi received a PhD in information and computer science from the University of California, Irvine. He is a member of the AAAI, ACM, and IEEE Computer Society and an active industry consultant.

John Noll is a software engineer at Network Appliance, Inc., where his interests include wide-area hypertext environments, process enactment mechanisms, and Web caching techniques. During the work reported in this article, he was a research associate at the ATRIUM Laboratory. Noll received an MS and a PhD in computer science from the University of Southern California.

Contact Scacchi at ATRIUM Laboratory,USC, Los Angeles, CA 90089-1421, scacchi@gilligan.usc.edu; or Noll at NetApp, 2770 San Tomas Expwy., Santa Clara, CA 95051, jnoll@netapp.com.