

A service delivery platform for server management services

Computer server management is an important component of the global IT (information technology) services business. The providers of server management services face unrelenting efficiency challenges in order to remain competitive with other providers. Server system administrators (SAs) represent the majority of the workers in this industry, and their primary task is server management. Since system administration is a highly skilled position, the costs of employing such individuals are high, and thus, the challenge is to increase their efficiency so that a given SA can manage larger numbers of servers. In this paper, we describe a widely deployed Service Delivery Portal (SDP) in use throughout the Server Systems Operations business of IBM that provides a set of well-integrated technologies to help SAs perform their tasks more efficiently. The SDP is based on three simple design principles: 1) user interface aggregation, 2) data aggregation, and 3) knowledge centralization. This paper describes the development of the SDP from the vantage point of these three basic design principles along with lessons learned and the impact assessed from studying the behavior of SAs with and without the tool.

J. Lenchner
D. Rosu
N. F. Velasquez
S. Guo
K. Christiance
D. DeFelice
P. M. Deshpande
K. Kummamuru
N. Kraus
L. Z. Luan
D. Majumdar
M. McLaughlin
S. Ofek-Koifman
D. P
C.-S. Perng
H. Roitman
C. Ward
J. Young

Introduction

Current business infrastructures comprise multiple components (e.g., database management servers, application servers, and Web servers) residing on hundreds of servers distributed across many networks and running on multiple operating systems [1–3]. Due to the large scale and complexity of IT infrastructures, server system management has emerged as a major component of the IT (information technology) services business [4]. The ever-increasing demands for data availability and high-quality service have amplified the importance of continuously available service, and as a result, the industry has encountered server system management costs that exceed system component costs [5–7]. In fact, the human cost of system administration has grown since the 1990s to represent two-thirds of total system administration costs [7], requiring a business to achieve significant efficiency gains in order to be competitive in the services marketplace.

Our research focuses on designing and developing viable new technology to help improve efficiency of server

system management activities so that a given system administrator (SA) can manage larger numbers of servers, in order to reduce service management costs and improve business competitiveness.

In general, SAs are responsible for the installation, configuration, maintenance, and troubleshooting of servers and their hardware and software components [2]. System administration encompasses many domains including operating systems, storage, databases, and networks and involves multiple levels of technical expertise [2]. The work activities of SAs pertain mainly to the ITIL** (IT Infrastructure Library) processes of incident, problem, and change management [8] and focus on resolving the incident, problem, or change (IPC) tickets initiated by or on behalf of the customers. An SA's work can be labor intensive, involving extensive collaboration with peer SAs for troubleshooting and knowledge or tool sharing [2]. This often requires specialized analytical and pragmatic skills [9].

Our research effort began with the idea that an SA equipped with a single application, offering visibility to

©Copyright 2009 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 by any means or distributed royalty free without further permission by computer-based and other information-service systems. Permission to *republish* any other portion of this paper must be obtained from the Editor.

integrated data across managed systems, would be more efficient than an SA who had to manually synthesize this same data from many disparate systems and applications. Preliminary evidence in support of this idea came from the findings of two IBM internal studies, a Day in the Life (DITL)¹ study conducted in 2003 and an Activity-Based Costing (ABC)² study performed in 2005. The DITL study focused on daily work tasks performed by the participants to identify their repetitive tasks as targets for automation. Building on the DITL study, the ABC study had the following objectives: 1) to build a baseline labor-activity model of the current activities of SAs, 2) to identify opportunities for improvement, and 3) to develop an implementation plan to realize the improvement opportunity. Both of these studies found that SAs were frequently switching between various tasks in diverse contexts, which caused the SAs to pursue potential problems that had little or nothing to do with their primary tasks of solving server problems.

Based on the SA activity model developed by these studies, we conducted observations and informal interviews with operating system SAs from several groups in the IBM Server System Operations (SSO) business. Our goal was to understand what tools they use and how they integrate the information extracted from these tools in the context of the various types of activities pertaining to their roles.

We extracted several salient insights, some matching findings of other studies of SA activity. First, in performing their tasks, SAs use a large number of tools with minimal integration or standardization [2, 10, 11]. The basic suite of management tools includes tools for secure access to managed servers and applications, as well as tickets, server configuration, and real-time and historical system vitals [i.e., critical information relating to aspects of system functioning, such as memory and CPU (central processing unit) utilization]. Second, SAs spend a significant amount of time performing data integration and synthesis to support their basic tasks. Examples of such integration and synthesis tasks include assessing the current state of the managed servers, performing troubleshooting [2], or prioritizing among tickets sourced from different tools or customers. SAs use a diverse set of knowledge sources to help their change-planning and troubleshooting efforts, including external search engines [2] and historical tickets [9]. Finally, SAs' work involves significant risk, given that problems left unresolved may have a significant negative impact on customer business operations [4, 12]. The analyses of these instances of SAs' work in the context of

their daily operational environment (i.e., situated work) were used to guide our tool design effort, as illustrated by previous studies [13–15].

More specifically, in this research, we propose and develop a context-aware environment that supports SAs' activities in the IT services environment. This context-aware environment is based on three simple principles: 1) user interface (UI) aggregation, 2) data aggregation, and 3) knowledge centralization. The proposed design is realized in the Service Delivery Portal (SDP), which provides a set of well-integrated technologies to help SAs efficiently perform their tasks.

UI aggregation facilitates the integration of commonly required functionality into a single operational console capable of executing a wide variety of tasks needed for server management. UI aggregation builds upon technology that enables secure access to tools and applications. The immediate benefit of this design principle is the elimination of overhead related to the secure access to individual tools. Further, UI aggregation provides a standardized view across tools with similar functionality but with different data models, such as models employing problem-ticket databases and server-configuration databases. UI aggregation is especially valuable for teams of SAs that work on multiple customer accounts supported by different tool suites, because it facilitates account cross-training and thus enables efficiency gains within the team.

Data aggregation facilitates integration across disparate data sources in a meaningful way such that its application provides the SA with the complete information required to perform particular tasks. Such aggregation reduces the SAs' overhead related to data collection and synthesis. One example of data aggregation is the summary of ticket and server status, which helps an SA to assess easily his pending workload and priorities, thus reducing the risk of SLA violations and improving operational efficiency. Another example of data aggregation, valuable to the SAs in the process of defect prevention, is the identification of change operations that are likely triggers of a given incident or problem ticket, which requires aggregation across ticket and server-configuration data sources.

Knowledge centralization facilitates the pooling of knowledge that individual SAs have accumulated over time in various content types into an easily searchable, centralized repository. Sample content types include internal content such as historical tickets, best-practice process descriptions and account-specific knowledge items, and external content such as Web feeds and content indexed by popular search engines. Valuable capabilities provided by knowledge centralization include full-text search and context-sensitive extraction of knowledge relevant for SAs' troubleshooting and other

¹IBM Global Services, "System Administrators—A Day in the Life," *IBM Internal Report*, 2003.

²IBM Integrated Technology Delivery and IBM Research, "SO Activity-Based Cost Analysis," *IBM Internal Report*, 2005.

activities. For instance, access to best practices and process knowledge helps SAs to deliver consistent operational support across the organization. Similarly, an SA working on an incident ticket is interested in finding similar tickets in order to guide his own problem-resolution efforts. Overall, knowledge centralization enables SAs to obtain knowledge, within appropriate privacy limits, from content shared by SAs directly or indirectly (e.g., in ticket resolutions) at the global enterprise level, across teams, and across accounts. Knowledge centralization improves business efficiency through more effective actions during incident and problem resolution as well as prevention of defects triggered by change operations.

While each of the three principles has an individual impact on the efficiency of SA activities, their integration in a context-aware [16] work environment brings further benefits with respect to reducing navigation overhead. In particular, the design of the graphical UI (GUI) guides the user through the likely use cases based on the observed context, such as the sequence of clicks that preceded the current state, without losing the thread of the user's investigation. For instance, the GUI provides links for jumping seamlessly from information about an open ticket to the list of similar tickets, thus providing a shortcut for troubleshooting. Similarly, as tickets and servers are always interrelated in an SA's work, the GUI provides links for jumping from information about servers to information about tickets and vice versa. Further, navigation is optimized by providing role-based shortcuts in order to reduce the user's time spent diagnosing and resolving problems.

The SDP instantiates the above design principles with the goal of improving SAs' effectiveness by making available to them data and knowledge acquired through integration, consolidation, and correlation of data from disparate server and service management tools and applications. The SDP implementation addresses key requirements for system administration tools, including extensive configurability to user and customer account specifics [10], tools for collaboration and communication [10], easy-to-use interfaces [10], scalability [10, 17] across servers and managed customer accounts, and currency of information [17] through user-controlled information refresh rates. The mapping of the design principles to the actual SDP implementation is guided by a set of use cases collected from narratives provided by SAs in a focus group, including a detailed breakdown in steps and sources of information for their most prevalent activities.

The paper proceeds with a discussion of related work, followed by a more detailed presentation of the design principles and their implementation in the SDP, and an evaluation of the SDP with respect to actual use and

perceived usefulness, two factors of information systems success. The paper concludes with a discussion of our findings and implications for future work.

Related work

Previous research has involved the study of the activities of computer SAs and IT support staff with a focus on understanding the specific nature of their work and the factors that affect their performance. In particular, research has focused on the types of activities performed by SAs [2, 18, 19], tool effectiveness [4, 10, 11, 17], approaches to diagnosis and problem solving [9, 13, 15], social aspects and collaboration [2, 20], and time management [1, 21]. This research is guided by the design of tools that aimed at improving the efficiency of various components of SA activity [10, 11]. This includes tools for direct interaction with the managed systems, tools for troubleshooting based on effective visualization [22], tools for personal activity management [23, 24], and tools for management of service delivery organized around the concept of activity [1].

In addition, several commercial products are used to improve the effectiveness of SA activities. For instance, the IBM Integrated Solutions Console (ISC) [25] allows the integration of multiple applications within a single Web interface. Similarly, the IBM Virtualization Engine (IBM VE) [26], based on ISC, integrates system-configuration, system-vitals, and task-execution tools across all servers in the environment.

Past research also identifies the need and benefits of centralized knowledge access in SA activity [2]. Relevant and timely information facilitates effective task execution and problem solving in daily work. Knowledge sharing promotes learning, collaboration, innovation, and expertise recognition [27].

The research presented in this paper builds on the findings reported in previous research related to the nature of SA work and tool requirements. As mentioned, our proposal improves the efficiency of SA activity through faster access to the necessary data, knowledge, and system management UIs. At the core of our approach is the elimination of overheads through automation of one or more actions in the path of frequently performed activities, as well as the facilitation of a faster comprehension and resolution of work items. While as a whole, the overall set of design principles is novel in the space of tools directed to support SA activity, the individual principles in our proposal relate to previous work. For instance, the IBM VE [26] employs UI and data integration to provide SAs with a centralized tool for control of resource states and resource relationships. Similarly, Reference [1] proposes the use of data aggregation for presentation of a consistent view of the pending structured and

unstructured work activities for improved activity management. However, none of these tools also addresses the principle of centralized knowledge across all types of content relevant for SAs' work.

User studies in the area of task management [23, 24] demonstrate that users integrate multiple sources of information in order to create a "one-glance," comprehensive view of all of the items of information involved in an instance of activity planning. As mentioned, the data aggregation principle in SDP is driven by a similar purpose, which is to provide the SAs with a comprehensive view of the state of tickets and servers so that they can promptly begin diagnosis and perform services.

The tool design proposed in this paper works with the same set of abstractions that the SAs currently handle in the context of their work, rather than introducing new abstractions, such as the activity abstraction in Reference [1]. We suggest that this approach facilitates adoption and, thus, the early realization of efficiency improvements within the organization.

User interface and data aggregation

The aggregation of UIs and data across multiple tools is an important design principle of the SDP. Our observations of SAs' activities, along with related studies [2, 10], show that the SAs relied heavily on functionality and information provided by multiple tools. They interpreted the data from each tool and then correlated the information to determine whether their action was required on an account level or a server level. In performing this correlation, the SAs spent much of their time interpreting data while switching between application sources. This observation suggests UI and data aggregation can lead to significant productivity gains.

The UI integration provided by SDP includes both explicit and implicit connections to target applications and managed systems. In an explicit connection, the GUI or console of an application or server is launched in context, through automated negotiation of secure access; example uses of explicit connections in the SDP include the accesses to the consoles of managed servers and related tools, as well as to ticketing systems and system-vitals repositories. With implicit connection, information extracted from a particular UI is presented in an SDP page, integrated with information from other sources. For instance, real-time and historical system vitals are extracted through implicit connection to monitoring tools such as IBM Tivoli* Monitoring (ITM) and presented on the SDP pages related to servers or problems, along with information from server-configuration databases. The implicit connection model provides the user with fast

access to information of interest as well as an integrated view with other sources of information.

As mentioned, the type of data aggregated by the SDP includes server vitals, server-configuration parameters, and ticketing data. Server-vitals data includes real-time and historical statistics on server performance metrics such as CPU utilization and file system utilization. Server-configuration data includes information regarding the system setup, such as machine hostname, IP (Internet Protocol) address, physical location, amount of installed RAM (random access memory), hard disk storage capacity, and installed software. *Ticketing data* refers to the details of incident, problem, or change (IPC) tickets.

The data-aggregation models provided by the SDP include both data federation and data correlation. *Data federation* refers to the assimilation of similar data from multiple sources such as IPC tickets, for the accounts managed by a user, coming from multiple ticketing sources, or configuration information coming from different configuration management systems. This model of data aggregation provides the SAs with a single view across all of the management components, thus improving the effectiveness of their activity planning [23].

Data correlation refers to the correlation of related data from disparate sources, a step that was performed manually prior to the SDP. Through the SDP, correlated data from the multiple types of sources is presented to the user without the need for the user's explicit login to each of the sources or manual extraction and correlation of data across sources. This aggregation saves time, improves problem determination, and results in a more stable and robust customer environment. **Figure 1** shows this type of UI and data consolidation. The left side of the screen displays data from IBM parity monitoring tool, and the right side shows data from a configuration management tool. Information pertaining to the servers is collected from these disparate tools and joined in the SDP, resulting in a single interface that provides a consolidated view of all managed servers.

Another instance of data aggregation in the SDP is the correlation of IPC ticket and real-time server status summaries into a view that highlights the status of the managed systems on a customer-account basis. This account dashboard, illustrated in **Figure 2** (upper table), supports the SA's effort to learn about his current work items and determine an activity plan [23]. In particular, the dashboard presents the number of servers down for each account, as well as the ticket queue, which involves the number of pending tickets for the individual SA and his team. A "click" (i.e., mouse-click selection) of the account name presents the SA with the view of the severities of the pending tickets, as seen in

Server list for: US - HIBRANDAND				
<< first < Showing servers 1 - 20 of 48 > last >>				
Server	State	OS	Type	Admin
aix20.hibrandand.com	Up	IBM AIX	Print Server	John Smith
aix34.hibrandand.com	Up	IBM AIX	Web Server	James Muller
aix37.hibrandand.com	Up	IBM AIX	Application Server	John Smith
aix38.hibrandand.com	Up	IBM AIX	Application Server	Kumar Rajes
aix40.hibrandand.com	Up	IBM AIX	Application Server	Adams Lee
aix44.hibrandand.com	Up	IBM AIX	Application Server	Kumar Rajes
aix45.hibrandand.com	Up	IBM AIX	Application Server	Jassica Chen
aix47.hibrandand.com	Up	IBM AIX	Application Server	James Muller
aix48.hibrandand.com	Up	IBM AIX	Application Server	Jassica Chen
IBM Parity Vitals		CMI (Configuration Management Interface)		

Figure 1

Account server summary.

Figure 2 (lower table). A subsequent click on the number of tickets for a given severity presents the list of tickets for the particular account and severity. As this example shows, the SDP provides access to data from multiple customer accounts with the data aggregated from several sources and presented to the user in one meaningful view.

The implementation of UI integration in the SDP is based on automated login to the various data sources, applications, or servers using the individual SA's credentials, which are stored in a password vault. A key enabler for UI and data aggregation is the extensive configurability of the SDP [10]. Users can configure the list of managed accounts (referred to as *my accounts*), credentials for a variety of applications, refresh period for data extracted from remote sources, and account-specific interpretation of content in select ticket fields.

Knowledge centralization

In the current service delivery environment, knowledge originates from different sources, in different formats, and with separate access credentials. The requirements addressed by the knowledge centralization principle in the design of SDP draw from our own observations and related studies [2, 27]. First, relevant knowledge spans a wide variety of content types from internal and external sources, including IPC ticket descriptions, problem resolutions, best practices, operational procedures, work team document repositories, wikis (e.g., wiki Web pages), forums, RSS (Really Simple Syndication) feeds, and

personal documentation. Integration with external sources is essential in order to make use of the vast space of relevant knowledge and to accommodate the most popular knowledge sources among SAs [2]. Second, ease of access and information quality facilitate effective task execution and problem solving in daily work. Finally, integration of search efforts across multiple sources speeds the access to relevant information across relevant content types.

The remainder of this section describes additional aspects of content collection and access within the SDP and presents details of the SDP functions that involve text search and content correlation.

Content collection and access

The SDP architecture, illustrated in **Figure 3**, supports knowledge management through automated content extraction from various repositories and content indexing within the SDP search engine. In addition, on-demand indexing is performed upon end-user upload or creation of documents within the SDP.

The internal knowledge sources integrated in the SDP include repositories of IPC tickets, operation "handover notes," operations-support knowledge bases (KBs), repositories of diagnostic procedures, best practices, and shared account team documents. Operation handover notes include details about server status, insights about association between tickets and servers, or other types of information that SAs deem necessary to provide to their team members in follow-on shifts for smooth

Ticket Summary						
Problem details						
Change details						
Incident details						
Ticket summary for account FR - ABCCORP						
Data source	Server: SESDR Ticket: eESM Ticketing system					
	Problems	Changes	Incidents	My problems	My Changes	My Incidents
new Open	3	5	n/a	0	0	n/a
Target due today	0	1	n/a	0	0	n/a
Target Overdue	1	3	n/a	0	0	n/a

Ticket List						
<< first < Showing tickets 1 - 3 of 3 > last >>						
Queue	Assigned to user	Open date	Due date	Severity	Problem id	Abstract
FLQ-IFRAHSUNI	PIERRE COIRA	03/09/09 15:01:03	-	3	IBM-18889646	KA3_sasprd02:GENDSKLGE001E Disk space used %: /department/st
FLQ-IFRAHSUNI	BECHIR BELGHI	03/07/09 02:24:27	03/09/09 01:02:31	2	IBM-18863163	ZA3_zs3sr3q02:ERRPTLGE001E: PERM_H: Nb Total Alerts :1 - Nb
FLQ-IFRAHSUNI	VINCENT LOURULT	02/14/09	-	3	IBM-18518979	AST: KA3_sivtm01:

Figure 2

Sample account dashboard and ticket details. The column labeled *Abstract* refers to ticket description abstracts and includes brief descriptions of the particular issues. *SESDR* refers to the name of a server-configuration database, and *eESM* is the name of a ticketing system. *FLQ* is an account symbol. The color-coding helps users easily identify the level of problem severity.

delivery of operations. These notes support collaborative work, as well as troubleshooting efforts through ticket and server correlation.

External knowledge sources include search engines, such as the standalone Google search engine, or engines that are integrated with support Web sites, such as is the case for the Microsoft Help and Support Web site.

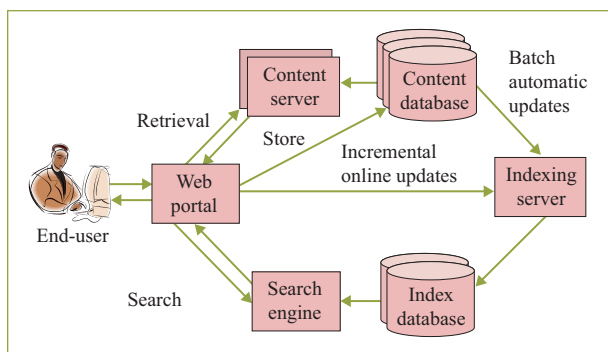


Figure 3

SDP knowledge portal architecture. The indexing server, at right, performs indexing of the content in order to facilitate rapid searching of the content.

Additionally, external sources include Web feeds, such as provided by *Red Hat Magazine*, VMware, and Slashdot. Each of these content types has specific procedures for document collection and access.

The SDP text search represents a unifying, seamless access paradigm over all of the aforementioned data domains in the KB. In order to avoid an overwhelmingly complicated search interface, the search is presented in two separate pages (**Figure 4**). One search page, labeled *Knowledge Base*, is relevant to searches normally done in the context of ticket resolution, giving access to historical tickets, Web feeds, handover notes, and external search repositories. A second page, labeled *Libraries*, provides access to documents normally accessed when performing changes and other support procedures. These content sources include account team libraries, best-practices documentation, and IMKD (information management knowledge database).

IPC ticket content is collected daily from multiple ticketing systems. The various pieces of information pertaining to a ticket are filtered and aggregated into a record stored within the SDP KB, or knowledge database. The SDP ticket record model is specialized for IPC tickets and is used to represent content from diverse IPC ticket sources. The content of an SDP ticket record supports the many tasks performed by the SDP target

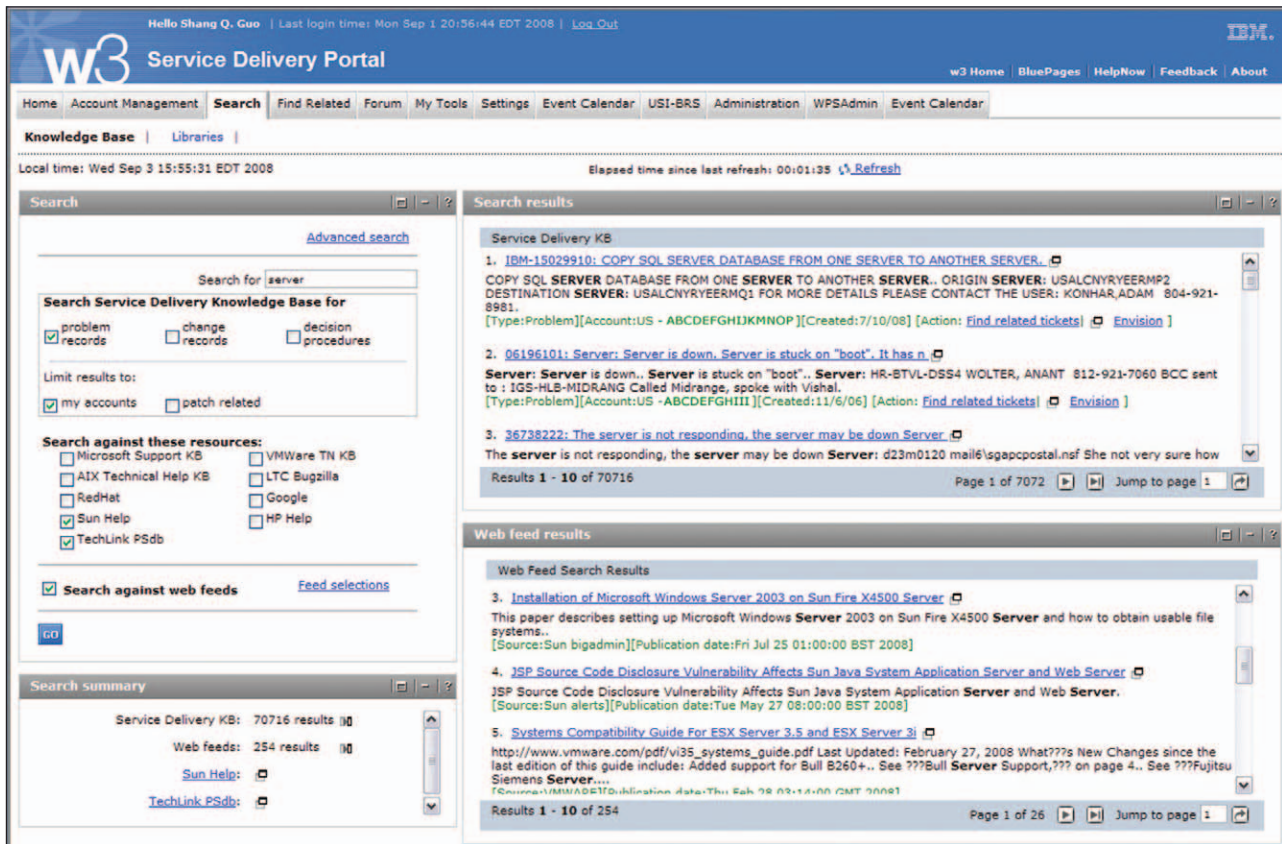


Figure 4

Knowledge-base search page, aimed at supporting problem-resolution activity.

user population, including solution reuse [9], ticket correlation within troubleshooting efforts, expert identification for initiation of collaborative work [2], and change risk assessment for defect prevention. From the search results, users can access the SDP ticket record or full details from the related ticket management system. Aside from SDP search, IPC tickets are accessible through content correlation functions, such as represented by *find-related* tickets and servers.

Content from external search sources is not indexed in the SDP KB, because of scalability and access limitations. Rather, our approach is to provide links to external result pages. More specifically, when the user submits an SDP query involving an external site, the SDP presents a link with the corresponding query directed to the external site. Upon clicking on the link, the user is transferred to the results page for that site where he can inspect the results and, if necessary, continue to refine the search. This approach provides a better quality of user experience compared to providing links to external sites.

Web feeds contain up-to-date, condensed, and an informative summary of content coming from Web resources and content providers. Web-feed content from select providers is collected and indexed in the SDP KB. Scalability derives from the typically small size of Web-feed content. SDP users can configure the list of Web feeds of interest for searching. Also, SDP administrators can configure the list of Web-feed providers from which to collect content to serve the entire SDP user population. From the search-result item related to a portion of Web-feed content, a user can access the text of the Web feed and can follow a link to the referenced Web page if the user has a deeper interest in the specific topic.

Decision procedures are in the form of executable flowchart-style documents with step-by-step instructions for effective diagnoses of well-understood problems. The text descriptions are indexed in the SDP KB as part of the decision-procedure authoring process. In the context of troubleshooting activity, the SDP users can search across decision procedures using standard text

search, and with a single click, they can launch the flowchart-execution engine for sequential execution of the flowchart.

The IMKD is a Lotus Notes*-based [28] document management system for knowledge documents and task instructions specifically authored to support the daily operations of IT service delivery personnel. Due to the relatively small volume of the IMKD content, the IMKD knowledge content is extracted and indexed in SDP, rather than searched via IMKD APIs (application programming interfaces). From the SDP search results, the user can open relevant IMKD documents in their own Notes client, or for faster access, the user can retrieve the document content stored in the SDP.

The best-practices library [29] is an application for authoring, management, and presentation of documented best practices covering all areas of IT service delivery. The best practices are described as flowcharts. The textual content of these documents is indexed in the SDP, thus providing for full-text search capabilities in addition to the original application features. SDP users locate relevant best-practices records of interest in the SDP and, with a click, or mouse-click selection, access the full description in the managing application.

The document library provides a single access point for uploading and sharing team documents of various types, such as Word**, Excel**, and PDF** (Portable Document Format). Users can use text searches to find documents and can view them in IMKD GUIs. Documents can be designated private to the account team or public for the entire SDP user population.

The performance impact of data collection on each of the SDP content sources is minimal. For the most critical sources, such as the IPC ticket databases, a combination of batch processing and off-peak access minimizes the impact.

Knowledge-base text search

The SDP text search is at the base of all KB interactions. Search is invoked either directly by user actions in the two SDP GUI search pages or indirectly from within the components that implement knowledge correlation functions.

The search GUI allows the user to control the scope of the operation by selecting the target data domains and accounts. The data domains are selected through the search page (e.g., a focus on ticket-resolution or steady-state activity) and page-specific checkboxes. For example, from the search page catering to ticket resolution, one can search over various types of tickets, external sources, and Web feeds. Account selection has two options: the accounts for which the user is assigned (labeled *my accounts*) or all accounts. The search across all accounts is limited by account-specific privacy settings. Within

a data domain, the user can make further scope selections, such as the specific external sites to include in an instance of external search. The scope selections are stored in the user's profile and used as the default setting for the next access to the page.

A search result summary presents the number of results found in the internal KB and links for queries over the selected external search sites, as seen in Figure 4. On the result list, each item has a hyperlinked title, a snippet of text from the neighborhood of the (highlighted) search terms, and details about the document, such as account, type, and timestamp, which help the user to identify with ease the most relevant item. Links specific to document type, such as links to functions for viewing the document in the GUI of the related ticket management application or links to related SDP functions such as "find related" speed the user's access to more details.

The implementation of text search in the SDP is based on the IBM eSearch search engine [30] with the IBM Juru search library [31]. eSearch is a Web application that provides for federated search through a set of locally stored indexes, each index representing a collection of knowledge referred to as a *knowledge base*. eSearch supports a rich search syntax allowing free text search based on keywords, as well as advanced search options such as fielded search, synonym expansion, access control, and time-based search.

Figure 5 illustrates the integration of eSearch with the other SDP components for the provision of search functionality. The content stored in the SDP KB is partitioned across multiple eSearch KBs (i.e., indexes), based on data domain, related accounts, and record types. For instance, Web feeds, best practices, and IMKD content are each mapped to dedicated KBs, because of the low volume of content. In contrast, IPC tickets are partitioned by accounts and ticket type, which ensures better relevance-based ranking in searches across multiple domains and record types and more scalable access times. For further scalability, the indexes are partitioned across multiple servers and eSearch instances.

Aside from eSearch, the SDP KB is supported by the *syndication hub* Web application and the feed-update agents. The hub monitors the Web on a regular basis, polling registered Web feeds, detecting feed changes, and notifying subscribing agents. The agents extract the updates and prepare the content for indexing into eSearch.

Content correlation: Find related

Observations suggest that SAs often solve technical problems by comparing and contrasting current server problems to those that have been solved in the past. Similarly, SAs determine the risk associated with change

tickets by analyzing the problems triggered by similar changes on other servers. The SDP find-related functionality correlates data across systems and accounts, a task that was not easily accomplished prior to the advent of the SDP. The functionality is organized into two separate subpages: Find Related Tickets and Find Related Servers.

The find-related-tickets function allows users to find tickets that are similar (i.e., correlated) to an input ticket and, for each of the similar tickets, to determine their associated servers, possible triggering and triggered tickets, and the SAs that resolved it. The function uses textual similarity criteria applied to a ticket description. The result is a list of similar tickets ordered by relevance and a list of people believed to have the best expertise in solving the problem, as illustrated in **Figure 6**. With a single click, the user can begin chatting with one of the experts (lower left of Figure 6), regardless of whether the expert was previously on the user's "buddy list" of the user's chat program.

For a change ticket, the function can provide a list of problems that could arise as result of the change. The statistical likelihood that a problem is caused by a given change is computed using a combination of textual similarity among change tickets, direct mentions of server names in problems triggered by a change, and temporal proximity of a triggered problem to the related change. These likelihood estimates provide a numerical score of the degree to which a similar problem could arise as a result of the current change ticket.

The find-related-servers function allows users to find servers similar to a given server based on one or more user-specified similarity criteria, including hardware, operating system, running applications, open software alerts, and the related customer account. The user can compare side-by-side the source and the similar servers with respect to configuration data, recently opened tickets, and real-time and historical traces of system vitals including CPU, memory, hard disk, and swap space utilization.

The challenge is that the functions estimating the similarity criteria do not satisfy the metric property and, specifically, a triangle inequality [32]. This makes it difficult to index the servers so that we can efficiently identify, upon user request, the most similar servers (i.e., the top- k servers) related to a given server [32]. Our solution for criteria with nonmetric value spaces is to precompute a criterion-specific similarity score between each pair of servers. At runtime, the precomputed similarities are combined according to user selection of similarity criteria in order to determine the overall similarity score.

The enabling element for the design of find-related functions is the availability of information about the

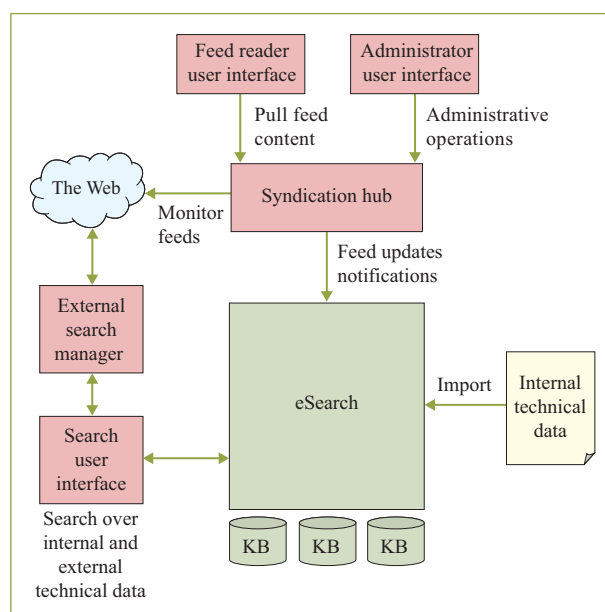


Figure 5

Service Delivery Portal search interactions (KB: knowledge base).

relationship between tickets and servers. Given that an explicit specification of this information is missing from most ticket records, the SDP uses ticket annotation techniques to produce it. In particular, ticket descriptions are analyzed and server names are identified based on a dictionary created from information in server-configuration databases. Ticket annotation is performed off-line, when tickets are loaded in the KB. The quality of the resulting annotations is relatively high: No false negatives exist (i.e., no relationship is missed), and only approximately 2% of false positives occur because of server short-name duplication across domains or matching to a common vocabulary word or product name.

SDP evaluation

Aligned with previous research regarding information system and KB success [33–36], we have studied the success of the SDP along two lines: actual use of the system and perceived usefulness of the system. The study is focused on the following topics: usage patterns of the various SDP functions, expected overhead savings due to UI and data integration, and perceived usefulness of SDP.

SDP usage patterns

As a step toward evaluating the benefits and usability of each of the various SDP functions, we analyzed the related usage and access patterns. For the various SDP

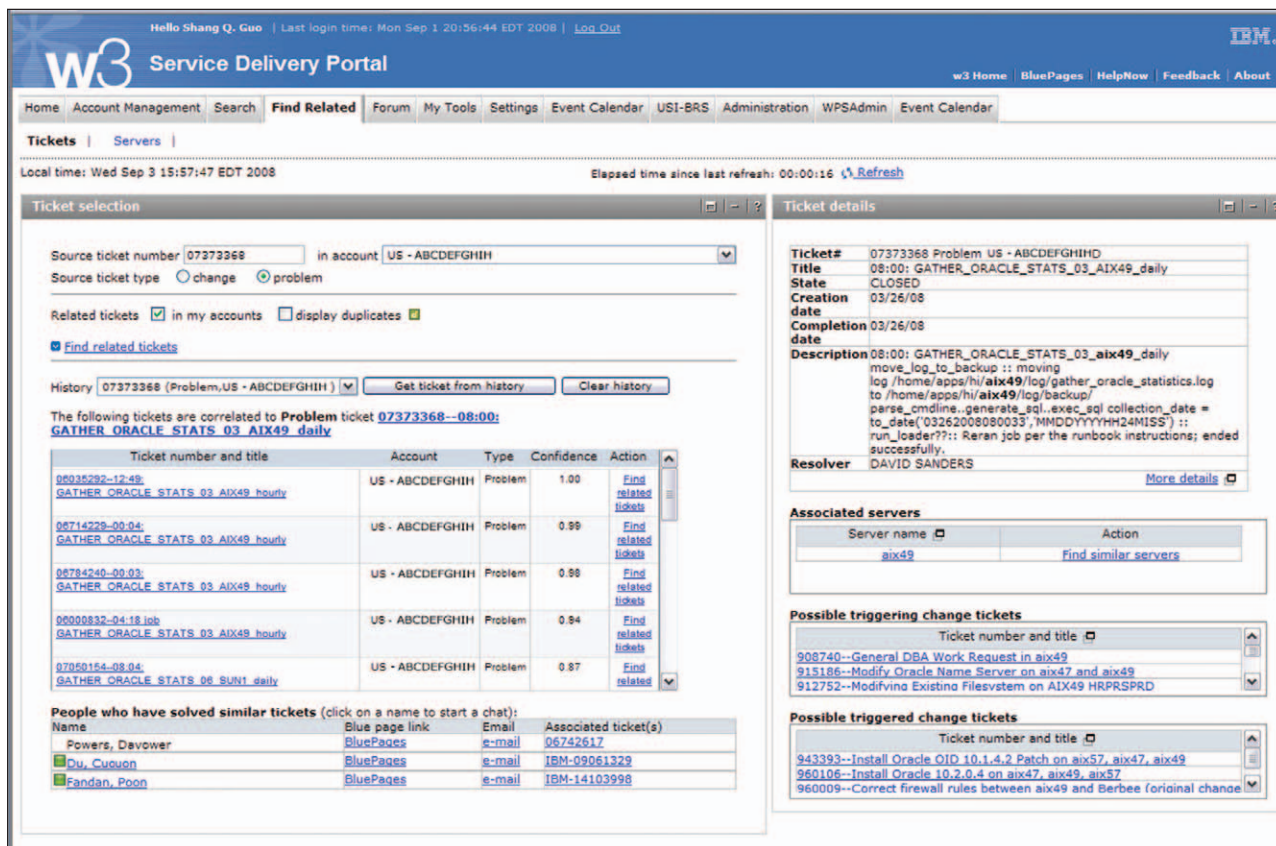


Figure 6

Find-related-tickets page, providing results from knowledge-base content correlation functions.

functions, our goal is to assess usage frequency and task-completion efficiency, the latter measured in terms of number of mouse-clicks employed to achieve a given task (i.e., the number of clicks required to reach and interact with the related GUI page). We acknowledge that this metric is limited in reflecting the context of the related interactions, such as the specific user goals of resolving a ticket or working on defect prevention. The lack of observations of users' situated work to match the observed sequences of SDP activity prevents us from a more insightful analysis using sequential analysis methods [37].

Our analysis is based on the log of user actions in the SDP, such as switching tabs on the GUI or entering a search query. A user action corresponds to a request form submission, an explicit page navigation action, and an explicit page refresh action. The log of user actions is derived by appropriately filtering the SDP portlet-invocation log. A notable limitation, to be addressed in future research, is the lack of references to actions implemented as direct JSP (JavaServer** Page)

invocations, including many data-integration functions such as the presentation of ticket details and system vitals.

The log includes traces from 716 users, with different experience levels. In particular, 153 users had more than 60 days of experience, 183 users had between 14 and 60 days of experience, and so forth, as depicted in **Table 1**. This assortment of experience levels is consistent with the fact that the SDP has a staged account-based deployment.

Across all users, the log comprised about 61,000 user actions, of which approximately 25,700 were form submissions. Over the final 2.5 months of the study, the log registered an average of 265 form submissions by an average of 69 users per day. Table 1 shows statistics of the logged usage for different levels of user experience with the SDP. The data shows that as the duration of users' experience increases, the ratio of form submissions as a percentage of total refreshes increases slightly, perhaps indicating that over time, the user's actions become more directed toward task completion.

Table 1 Usage statistics from the user-action log. User actions include form submissions, explicit page navigation, and page refreshes.

<i>Experience with application</i>	<i>User count</i>	<i>Total user actions</i>	<i>Form submissions</i>	<i>Forms as % of total actions</i>
Experience < 14 days	380	6,282	2,302	39.0%
14 ≤ experience < 60 days	183	31,791	9,138	40.3%
Experience > 60 days	153	32,716	14,328	43.8%
Total	716	61,268	25,768	42.0%

In order to evaluate function popularity and efficiency of access, we targeted the users that were reasonably familiar with SDP functions. As a result, we selected the traces of users with more than 60 days of experience, and the actions in their last 30 days of activity, thereby avoiding their trial and training period.

We extracted the frequency of the various actions considered individually and in sequences (i.e., a series of actions without repetitive subsequences). From the occurrences of the function accessible in the SDP home page, Account Dashboard, we remove the instance related to login, thus reporting only the explicit user actions. **Table 2** presents the most popular functions based on the related action clicks. The data is consistent with the fundamental SDP design assumptions. In particular, the most popular SDP functions are those related to the data integration across ticket and server-configuration domains (90% of accesses), and the UI integration for access to application and server consoles (8% of accesses). The SDP KB functions (ticket search and find related tickets and servers) (2% of accesses) are used significantly less frequently than account-management functions. This observation is not completely surprising since the KB functionality services fewer use cases. However, these functions do have the potential to yield greater value to the SA per time invested in the application compared to account-management functions because they support the efficient completion of problem determination and resolution activities, which are a significant portion of the SAs' time [2].

Within the KB functions, knowledge correlation across tickets and servers (i.e., the find-related functions) is more popular than search (1.3% for tickets and servers versus 0.6% for search). Details from invocation parameters reveal that searches through external sites, such as the AIX* (Advanced Interactive eXecutive) Technical Forum, are more popular than Web-feed searches. Within internal sources, ticket content search is more popular than searches of document libraries, best practices, and handover notes.

Users accessed the SDP with diverse access rates and patterns. For instance, the trace included 46 users with

100 to 1,084 actions, 45 users with 50 to 100 actions, and 81 users with fewer than 50 actions. On average, each user session involved 17 actions.

By far the two most frequent action sequences (>97%) were related to account dashboard updates and the retrieval of account status details. We infer that it is typical for users to stay connected to the portal, updating the view in the dashboard repeatedly (or letting it refresh automatically) in order to learn about account status updates. When updates occur, users look for ticket and server details in order to resolve the particular situation. The find-related server or ticket and search appear sporadically in the access sequence, only when the user sought additional insights, when starting on the problem or server management pages. When search appeared in an action sequence, often users repeated the function and occasionally moved to and from the problem description pages.

This analysis reveals that users access the SDP with very diverse frequency models; further research is needed to understand the triggering factors. Also, the analysis reveals that the SDP design enables efficient user interactions, but further improvements are identified.

Table 2 Service Delivery Portal (SDP) function popularity across users with more than 60 days of usage in the latest 30 days.

<i>SDP function</i>	<i>% utilization</i>
Account dashboard	77.3%
Single-click access to applications and servers	7.6%
Server vitals and configuration	6.2%
Problem ticket summary and detail	4.5%
Server summary for account	1.3%
Find related tickets	0.7%
Server compare selection	0.6%
Knowledge-base search	0.6%
Find related servers	0.3%
Frequently asked questions (FAQs)	0.2%
Other	0.5%

Table 3 Observed sequence of events involved in an incident-resolution session.

1. Open PuTTY session to the problem server. Fail because of mistyping server name.
2. Open PuTTY session, successfully.
3. Open ticketing system, and log on.
4. Lose connectivity to ticketing system, followed by multi-person chat session with several people to determine cause of connectivity loss.
5. Reconnect to ticketing system.
6. Determine that a new ticket is assigned to the system administrator.
7. Chat session with colleague on current configuration setup of the problem server.
8. Reconnect to ticketing system after session has timed out.
9. Create ticket note regarding the current configuration of the problem server.
10. Access server console to change configuration settings on the server.
11. Chat session to explain the problem to team member.
12. Update ticket with root-cause details.
13. Close ticket and shut down the system.

For instance, a new alarm feature is developed to eliminate the need for refresh clicks related to account-status updated. Finally, the SDP KB functions are used predominantly in the context of problem resolution, validating the relevance of the SDP context-aware design in transitioning from account-management and KB functions.

Time overhead savings due to data and UI aggregation

This section evaluates the expected time overhead savings in the incident-resolution process due to the SDP design principles of UI and data aggregation. The time overhead is estimated by the number of application accesses, as each access involves overheads related to application launch and secure login. This measure is an underestimation of the actual time-saving because it does not include the savings due to more effective sense-making enabled by the integrated presentation of data [22]. The evaluation is based on videos (~13 hours) of on-screen ticket-resolution activities before deployment of the SDP, using the screen-casting application Camtasia Studio** developed by the TechSmith Corporation.

From among the captured videos, we analyzed six representative incident-resolution sessions. For each session, we identify the accesses to various tools and

applications, accounting for every distinct access, even if an application is accessed multiple times in a session. The most frequently accessed tools are the ticket management system, chat application, and server console via a secure shell client. **Table 3** illustrates a sample incident-resolution session. Consistent with previous research [2], our observations show that communication with peer SAs is an important component of the ticket-resolution process, often perceived as speeding up task completion. For instance, in the sample session presented in Table 3, the SA found it more convenient to ask a colleague about the configuration setup of the server rather than check by himself by accessing the server or a server-configuration system.

We flagged each activity as to whether it could be handled within the SDP or not. For instance, within the SDP, logging into a managed server can be performed with a single-click launch. Overall, the SDP does not eliminate the need for any of the most frequently used tools, but it reduced the number of explicit accesses to the tools and the access overhead.

We found that a user performed an average of 12 application accesses per ticket, of which an average of 4.33 accesses could be handled from within the SDP, representing 36% of the total incident-resolution activity. This large percentage illustrates that the SDP makes a significant step toward providing the user with a self-contained environment for his daily activities. Further, we note that in the observed scenarios, none of the SAs have accessed a knowledge repository or searched historical tickets in the ticketing systems. Yet, they acquired equivalent information from peers through direct communication. In light of previous studies [20], the easy access to server-configuration details that is enabled by the SDP could help limit interruptions observed by SAs due to activity outside their immediate tasks and thus further improve SA productivity.

Perceived usefulness

In order to assess the perceived usefulness of the SDP, we use an unrefined survey instrument, distributed to 19 trial users. Aside from operating system SAs, which is the target SDP user type, the set of participants includes other roles such as command-center operator, team lead, application administrator, and technical customer manager. The survey evaluated the impact of various SDP services on the users' job performance and efficiency, two factors of perceived usefulness identified in the literature [38]. The survey comprised 73 questions with answers on a 1–5 scale. The questions addressed qualitative aspects, technical features, and quality and efficiency gains generally related to the SDP and also to each of its main functions.

The survey found a relevantly high level of perceived usefulness of the SDP in general, drawing from productivity and service quality improvements. Considering individual functions, the survey found that users in all roles had a very positive opinion on the effect of UI aggregation on the quality and efficiency of their work. The function ranked first with respect to perceived usefulness was the single-click access to applications and servers via a password vault. Another insight is that users that manage multiple accounts find the SDP significantly more useful than users that manage a single account. Further, the users that solve IPC tickets on a regular basis found relatively larger benefits from KB search centralization than users (such as command center operators) who do not handle tickets.

The results of this survey-based study were in accord with the insights from the study of the SDP usage patterns, in showing that users perceive higher benefits from the efficiencies that the SDP provides related to account-management activities than to those related to problem resolution and knowledge sourcing.

Discussion of findings and limitations

Overall, the evaluation of the SDP presented in this paper—addressing the factors of actual use, predicted and perceived usefulness—indicates that the SDP can foster relevant efficiencies, mainly drawing from the application of the SDP design principles of UI and data integration to the activities related to account and server management. However, some of the findings challenged our expectations. For instance, the average number of user actions in the SDP is lower than expected. While the results can be biased by the actual volume of activity during the observed period, the user survey reveals that the context of an SA's work is also a relevant factor. For instance, when assigned to working on a single account, the SA does not perceive that the existing set of tools has sufficiently high efficiencies to motivate the SA's daily use of the SDP.

Further, the evaluation reveals a significantly low level of the actual use and perceived usefulness of the SDP functions that support problem resolution through exploitation of the SDP design principle of knowledge centralization. This is different from our expectations based on the positive feedback on the related design and implementation received from a diverse set of SAs. Multiple factors can contribute to this situation. First, the quality of the current tool implementation or the integrated knowledge sources could limit the efficient access to the needed information [27]. Second, the nature of the system administration work and workloads could indicate that after a period of training, an SA's need for knowledge sourcing diminishes as the SA accumulates a relevant level of knowledge that enables

efficient resolution for a very large share of his daily tasks. Third, cultural and environment factors such as the availability of group-specific or account-specific documentation for problem resolution, or the easy access to colleagues [2], could provide knowledge sources considered more trusted and efficient [27]. Finally, a limited level of SDP training could prevent the users from exploiting the full capabilities of the SDP environment [39].

Our evaluation is limited by the lack of extensive observations of situated work, which prevents us from using validated methods for modeling user activity [39, 40] and analyzing it in context using sequential analysis [37]. The sequential analysis method would allow us to identify the relevant factors and develop new use-case scenarios, which could drive our actions toward maximizing the efficiencies that the SDP brings to the organization, such as the development of new functions, or specialization of training with respect to a user's work context.

Another limitation of our evaluation involves the low number of samples related to the use of KB functions in the study of actual use patterns. As a result, the insights on the relative use of these functions have a low confidence level. However, we are confident about the comparison between account-management and KB functions; similar relative values are observed when extending the observed window of activity beyond 60 days. For our future analysis of SDP features, we hope to have access to a larger base of active SDP users, which would enable us to reach appropriate confidence levels.

Conclusion and future work

This paper presented a novel set of design principles that underlie the architecture and implementation of the SDP, an integrated tool environment to help server SAs to more efficiently perform their daily tasks.

The insights and feedback gained from our global SDP deployment demonstrate that the design principles proposed by this research, namely UI and data aggregation, and knowledge centralization within a context-aware environment, as well as their instantiations in the current SDP architecture, are driving efficiency improvements beyond the scope of server system administration, which was the initial target of our efforts.

The SDP implementation is guided by studies related to the success of IS (information system), and, more specifically, of knowledge repository systems. In particular, the SDP implementation addresses the key IS success factor of system quality [17, 39] by targeting features such as reliability, ease of use, and fast response time. The SDP fosters user participation in the tool development, which affects both system quality and perceived usefulness [39] through integration of tools for

direct input of feedback and for user forums. The SDP deployment experience informally verifies the relevance of IS success factors that relate to training and support of upper-level management personnel [39].

Related to knowledge repository success [27], the SDP addresses the key requirement of fast access to knowledge of interest by providing specialized solutions to some frequent use cases. However, the study of usage patterns and the user survey reveal that more should be done for improving the use of the KB features. We plan to gain insight on the factors that affect the use of knowledge such as GUI usability, data quality, and actual knowledge needs and exploit these insights in future SDP releases.

The main goal of our future work is to understand the contribution of the factor groups mentioned above by examining in deeper details the work practices and situated work of SAs both before and after their becoming SDP users. We plan to use a diverse set of instruments, including interviews, observations, video capture, and surveys and employ validated methods for modeling and analyzing user activity [37, 40].

Further, our goal is to expand the reach of the SDP to a wider user base, beyond the domain of server system administration. Related studies [1, 2] demonstrate that SAs in a variety of areas, such as database, network, and application management, are faced with similar challenges as server SAs with respect to diversity of tools and data sources required for performing their activities, as well as the wealth of knowledge sources from which they can benefit. Further, our own observation of SDP activities reveals that SAs in a variety of areas (as well as other classes of users in the service management business, such as team leaders, dispatchers, and quality analysts) benefit from the principles underlying the SDP architecture and the current set of SDP features. As the target range of SDP users expands, specialized SDP extensions should be investigated, such as the integration of additional UIs and internal knowledge sources, or new instances of data aggregation, such as the association of tickets with related business applications, and new functions for specialized knowledge extraction across multiple sources.

*Trademark, service mark, or registered trademark of International Business Machines Corporation in the United States, other countries, or both.

**Trademark, service mark, or registered trademark of the Office of Government Commerce, Microsoft Corporation, Adobe Systems Inc., Sun Microsystems, Inc., or TechSmith Corporation in the United States, other countries, or both.

References

1. J. Bailey, E. Kandogan, E. Haber, and P. Maglio, "Activity-Based Management of IT Service Delivery," *Proceedings of the Symposium on Computer-Human Interaction of the Management of Information Technology (CHIMIT)*, 2007.
2. R. Barrett, E. Kandogan, P. Maglio, E. Haber, L. Takayama, and M. Prabaker, "Field Studies of Computer System Administrators: Analysis of System Management Tools and Practices," *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW)*. Chicago, IL, 2004, pp. 388–395.
3. D. Patterson, A. Brown, P. Broadwell, G. Candea, M. Chen, J. Cutler, P. Enriquez, et al., "Recovery-Oriented Computing (ROC): Motivation, Definition, Techniques, and Case Studies," *Technical Report CSD-02-1175*, University of California, Berkeley, 2002.
4. E. Anderson, "Researching System Administration," Ph.D. dissertation, Computer Science, University of California at Berkeley, 2002.
5. P. Horn, *Autonomic Computing: IBM's Perspective on the State of Information Technology*; see http://www.research.ibm.com/autonomic/manifesto/autonomic_computing.pdf.
6. J. Kephart and D. Chess, "The Vision of Autonomic Computing," *IEEE Computer* **36**, No. 1, 41–51 (2003).
7. D. Patterson, "A Simple Way to Estimate the Cost of Downtime," *Proceedings of the Large Installation System Administrator's Conference (LISA)*, Philadelphia, PA, 2002, pp. 185–188.
8. Office of Government Commerce, United Kingdom, *ITIL Service Operations Book*, The Stationery Office, 2007.
9. B. T. Pentland, "Bleeding Edge Epistemology: Practical Problem Solving in Software Support Hot Lines," *S.R. Barley and J.E. Orr Between Craft and Science: Technical Work in U.S. Settings*, ILR Press, New York, 1997.
10. E. Haber and J. Bailey, "Design Guidelines for System Administration Tools Developed through Ethnographic Field Studies," *Proceedings of the Symposium on Computer-Human Interaction for the Management of Information Technology (CHIMIT)*, Cambridge, MA, 2007.
11. N. F. Velasquez and S. P. Weisband, "Work Practices of System Administrators: Implications for Tool Design," *Proceedings of the Symposium on Computer-Human Interaction for the Management of Information Technology (CHIMIT)*, San Diego, CA, 2008.
12. G. Halprin, "The Workflow of System Administration," *Proceedings of the Conference of the System Administrators Guild of Australia*, Canberra, Australia, 1998.
13. J. E. Orr, *Talking About Machines: An Ethnography of a Modern Job*, Cornell University Press, Ithaca, New York, 1996.
14. L. A. Suchman, *Plans and Situated Actions: The Problem of Human Machine Communication*, Cambridge University Press, New York, 1987.
15. L. A. Suchman and E. Wynn, "Procedures and Problems in the Office," *Office: Technology and People* **2**, 133–154 (1984).
16. N. Hanssens, A. Kulkarni, R. Tuchida, and T. Horton, "Building Agent-Based Intelligent Workspaces," *Proceedings of Agents for Business Automation (ABA)*, 2002.
17. N. F. Velasquez, S. Weisband, and A. Durcikova, "Designing Tools for System Administrators: An Empirical Test of Integrated User Satisfaction Model," *Proceedings of the 22nd Large Installation System Administration Conference (LISA '08)*, San Diego, CA, November 9–14, 2008, pp. 1–8.
18. E. Haber and E. Kandogan, "Security Administrators. A Breed Apart," *Proceedings of Workshop on Usable IT Security Management at the Symposium on Usable Privacy and Security (SOUPS)*, 2007.
19. R. Barrett, P. P. Maglio, E. Kandogan, and J. Bailey, "Usable Autonomic Computing Systems: The Administrator's Perspective," *Proceedings of the International Conference on Autonomic Computing (ICAC)*, New York, 2004.
20. L. A. Perlow, "The Time Famine: Toward a Sociology of Work Time," *Administrative Sci. Q.* **44**, No. 1, 57–81 (1999).
21. V. M. Gonzales and G. Mark, "Managing Currents of Work: Multi-tasking among Multiple collaborations," *Proceedings of*

- European Conference on Computer-Supported Cooperative Work (ECSCW), Paris, France, 2005, pp. 143–162.
22. D. M. Russell, “Learning to See, Seeing to Learn: Visual Aspects of Sensemaking,” *Proceedings of the SPIE*, Santa Clara, CA, 2003, pp. 8–21.
 23. V. Bellotti, B. Dalal, N. Good, D. G. Bobrow, and N. Ducheneaut, “What a To-Do: Studies of Task Management Towards the Design of a Personal Task List Manager,” *Proceedings of Conference on Computer-Human Interaction (CHI)*, Vienna, Austria, 2004, pp. 735–742.
 24. V. M. Gonzalez, L. Galicia, and J. Favela, “Understanding and Supporting Personal Activity Management By IT Service Workers,” *Proceedings of the 2nd ACM Symposium on Computer Human Interaction for Management of Information Technology (CHIMIT)*, San Diego, CA, 2008.
 25. IBM, IBM Integrated Solutions Console; see <http://www.ibm.com/developerworks/autonomic/books/isc-readme.htm>.
 26. IBM, IBM Virtualization Engine; see <http://publib.boulder.ibm.com/infocenter/eserver/v1r2/topic/esmcfinfo/eicac.pdf>.
 27. P. H. Gray and A. Durcikova, “The Role of Knowledge Repositories in Technical Support Environments: Speed Versus Learning in User Performance” in *Management Information Systems* **22**, No. 3, 159–190 (2006).
 28. IBM, Lotus Notes and Domino 8.5; see <http://www-01.ibm.com/software/lotus/notesanddomino/nd85.html>.
 29. H. Qu, J. Sun, and H. Jamjoom, “SCOOP: Automated Social Recommendation in Enterprise Process,” *Proceedings of the IEEE International Conference on Services Computing (SCC)*, Honolulu, Hawaii, 2008, pp. 101–108.
 30. C. David, M. Shtalhaim, and A. Soffer, “eResponder: Electronic Question Responder,” *Proceedings of International Conference on Cooperative Information Systems (CoopIS)*, 2000, pp. 150–161.
 31. C. David, E. Amitay, M. Herscovici, Y. Maarek, Y. Petruschka, and A. Soffer, “Juru at TREC 10—Experiments with Index Pruning,” *Proceedings of Text Retrieval Conference (TREC)*, 2001.
 32. P. Zezula, G. Amato, V. Dohnal, and M. Batko, *Similarity Search—The Metric Space Approach*, Springer, New York, 2006.
 33. W. DeLone and E. McLean, “Information Systems Success: The Quest for the Dependent Variable,” *Information Syst. Res.* **3**, No. 1, 60–95 (1992).
 34. A. Kankanhalli and B. Tan, “Knowledge Management Metrics: A Review and Directions for Future Research,” *Int. J. Knowledge Mgt.* **1**, No. 2, 2–32 (2005).
 35. A. Rai, S. Lang, and R. Welker, “Assessing the Validity of IS Success Models: An Empirical Test and Theoretical Analysis,” *Information Syst. Res.* **13**, No. 1, 50–69 (2002).
 36. P. Seddon, “A Respecification and Extension of the Delone and Mclean Model of IS Success,” *Information Syst. Res.* **8**, No. 3, 240–153 (1997).
 37. R. Bakeman and J. M. Gottman, *Observing Interaction: An Introduction to Sequential Analysis*, 2nd Edition, Cambridge University Press, New York, 2008.
 38. F. D. Davis, “Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology,” *Mgt. Information Syst. Quart.* **13**, No. 3, 319–340 (1989).
 39. R. Subherwal, A. Jeyaraj, and C. Chowa, “Information System Success: Individual and Organizational Determinants,” *Mgt. Sci.* **52**, No. 12, 1849–1864 (2006).
 40. B. A. Nardi, *Context and Consciousness: Activity Theory and Human-Computer Interaction*, The MIT Press, Cambridge, MA, 1996.

Received October 31, 2008; accepted for publication February 1, 2009

Jonathan Lenchner IBM Research Division, Thomas J. Watson Research Center, 19 Skyline Drive, Hawthorne, New York 10532 (lenchner@us.ibm.com). Dr. Lenchner is a Senior Technical Staff Member and former manager in the Services Research department at the IBM T. J. Watson Research Center, where his work focused on making technical support more efficient within IBM. His current work focuses on minimizing energy consumption of computers and data centers. He earned a B.A. degree from Dartmouth College, a CPGS (Certificate of Postgraduate Study) from Cambridge University, and a Ph.D. degree from the Polytechnic Institute of New York University, all in mathematics.

Daniela Rosu IBM Research Division, Thomas J. Watson Research Center, 19 Skyline Drive, Hawthorne, New York 10532 (drosu@us.ibm.com). Dr. Rosu is a Research Staff Member in the Service Delivery department of the IBM T. J. Watson Research Center. Her current research interests include IT service operations management and support for effective problem resolution. In the past, Dr. Rosu worked in multiple areas including business goal-driven resource management in complex IT environments, operating systems support for high-performance Web servers, distributed Web-caching infrastructures, and real-time operating systems. She received a Ph.D. degree in computer science from the Georgia Institute of Technology in 1999, with a dissertation in the area of adaptation in complex real-time systems. She also holds an M.S. degree in computer science from Georgia Institute of Technology (1995) and an M.S. degree in theoretical computer science from the Faculty of Mathematics, University of Bucharest, Romania (1987).

Nicole F. Velasquez University of Arizona, P.O. Box 210108, Tucson, Arizona 85721 (nicolefv@gmail.com). Dr. Velasquez is a postdoctoral research associate in the Management Information Systems department at the University of Arizona. She received a B.S. degree in business information systems from Utah State University in 2001 and a Ph.D. degree in management information systems from the University of Arizona in 2008. She has worked with the IBM Systems and Technology Group in the past as a performance analyst and hardware tester for enterprise storage, and with IBM Research as an intern. Dr. Velasquez is author or coauthor of many refereed conference and journal papers, two technical papers, and one patent (application pending). Her research interests include human-computer interaction, technology impacts, and knowledge management. Dr. Velasquez is a member of the Association of Information Systems, the Association of Computing Machinery, and the League of Professional System Administrators.

Shang Guo IBM Research Division, Thomas J. Watson Research Center, 19 Skyline Drive, Hawthorne, New York 10532 (sguo@us.ibm.com). Dr. Guo is a Research Scientist in the Service Research department at the Thomas J. Watson Research Center. She received B.S. and M.S. degrees in computer science from Zhejiang University, China, and a Ph.D. degree in computer science from Illinois Institute of Technology. She joined the IBM Software Group in 1999 and subsequently joined the IBM T. J. Watson Research Center in 2004, where she worked in the area of Services Research. Her research interests include graphics, visualization, intelligent GUIs, and distributed computing.

Ken Christiance IBM Research Division, 3039 Cornwallis Road, Research Triangle Park, North Carolina 27709 (kdchris@us.ibm.com). Mr. Christiance is a Distinguished Engineer within the IT (information technology) Delivery organization, responsible for enterprise management architecture. He has been with IBM for 21 years and has 15 years of deep experience working on IPC (incident, problem, or change) management systems with a breadth that includes operational management products covering systems management and business

management disciplines. His published and pending patents in this area address enterprise architecture for systems management, productivity solutions for distributed server management, and methods for utilizing system data to support business processes such as billing and charge-back, data normalization across disparate environments, and automation solutions that provide efficiencies for distributed system management.

Don DeFelice *IBM Integrated Technology Delivery Division, 2455 South Road, Poughkeepsie, New York 12601 (donaldd@us.ibm.com).* Mr. DeFelice is an IT Specialist and Software Engineer in the e-Business Configuration and Management Development department. He specializes in WebSphere* portal development utilizing Web 2.0 technologies including AJAX (asynchronous JavaScript** and extensible markup language), DHTML (dynamic hypertext markup language), J2EE (Java** 2 Platform Enterprise Edition) and advanced user-interface design. Mr. DeFelice has been associated with the Service Delivery Portal project since its inception. He received a B.S. degree in computer science from SUNY New Paltz in 1993 and has worked at IBM since November 2000. In 2006, he published an IBM internal white paper on transforming Web applications into portlets.

Prasad M. Deshpande *IBM Research Division, India Research Laboratory, Embassy Golf Links Business Park, Block D, Bangalore, India 560071 (prasadesh@in.ibm.com).* Dr. Deshpande is a Research Staff Member at IBM India Research Laboratory in Bangalore. His areas of expertise include data and knowledge management, data mining, OLAP (online analytical processing), and related technologies. He received a B.Tech. degree in computer science and engineering from the Indian Institute of Technology, Bombay and M.S. and Ph.D. degrees in database systems from the University of Wisconsin, Madison. After graduation, he worked in various startup companies for four years before joining IBM Almaden Research Center in San Jose. He subsequently moved to IBM India Research Laboratory, to the knowledge management group. He has authored or coauthored more than 20 publications in conferences and journals, served on program committees, and has a number of patents granted and applied for. Dr. Deshpande is a member of the Association for Computing Machinery.

Krishna Kummamuru *IBM India Research Laboratory, Indian School of Business Gachibowli, Hyderabad, India 500032 (kummamuru@in.ibm.com).* Dr. Kummamuru received the B.Sc. degree in physics from Nagarjuna University in 1986. He received an M.E. degree and a Ph.D. degree in electrical engineering from Indian Institute of Science, Bangalore, India, in 1993 and 1999, respectively. He has been a Research Staff Member in IBM India Research Laboratory since December 1998. He is presently managing IBM collaboration on service science with the Indian Business School, Hyderabad, India. He received the Alfred Hay Medal for the best graduating student in electrical engineering in 1993 from the Indian Institute of Science. He has five patents granted by the U.S. Patent and Trademark Office (USPTO) and about 20 publications in refereed conferences and journals. He has roughly ten invention disclosures under evaluation by the USPTO. Dr. Kummamuru's research interests are in service science, machine learning, and data and text mining.

Naama Kraus *IBM Research Division, Haifa Research Laboratory, University Campus, Mount Carmel, Haifa, 31905, Israel (naamah@il.ibm.com).* Ms. Kraus is a Research Staff Member in the Information Retrieval department at the Haifa Research Center. She received a B.Sc. degree in computer science and mathematics from Bar-Ilan University in 1996. In 1997, she joined IBM at the Haifa Research Center. She worked in the area of parallel and distributed computing, and in the area of information retrieval. In 2006, she received an IBM Research

Division Award for her contribution to parallel checkpoint/restart. In 2008, she received an IBM Research Division Award for her contribution to the Information Technology Delivery Service Delivery Portal. She is an author or coauthor of four patents and three technical papers.

Laura Z. Luan *IBM Research Division, Thomas J. Watson Research Center, 19 Skyline Drive, Hawthorne, New York 10532 (luan@us.ibm.com).* Dr. Luan received her Ph.D. in electrical engineering from Keio University in Japan. She joined IBM in late 1996 and is an Advisory Engineer in the Services Research department. Her current research interests are in the areas of problem management and knowledge discovery in IT service delivery. She has worked in a variety of areas including service-level management in utility computing infrastructure, automation of composite software installation, and distributed command and control for real-time high-bandwidth multimedia streaming. Prior to joining IBM, she was a scientific staff member at Nortel Research Division from 1990 to 1996, working on communications network synchronization and digital transmission systems.

Debapriyo Majumdar *IBM Research Division, India Research Laboratory, Embassy Golf Links Business Park, Block D, Bangalore, India 560071 (debapriyo@in.ibm.com).* Dr. Majumdar is a researcher in the Services Information Analytics department at the IBM India Research Laboratory, Bangalore, India. He received a B.Sc. degree in mathematical sciences from Chennai Mathematical Institute, Chennai, India, in 2001, and M.Sc. and Ph.D. degrees in computer science from Max-Planck-Institute for Informatics, University of Saarland, Germany, in 2003 and 2007, respectively. In 2007, he joined the IBM India Research Laboratory. Dr. Majumdar's primary research interest is in the areas of text retrieval and analysis, as well as data engineering. He is author or coauthor of nine research papers and three patents.

Martin McLaughlin *IBM Integrated Technology Delivery Division, Glasgow, United Kingdom (mmclaugh@uk.ibm.com).* Mr. McLaughlin is currently the Global IT Architect on the Service Delivery Portal project, which is an internal ITD (integrated technology delivery) Web portal project. Mr. McLaughlin has been an IT Architect for the last 10 years of his 18-year IBM career and has worked in numerous Information Technology Delivery projects during that time. He received a B.Sc. degree in computer information systems from Glasgow Caledonian University in 1991.

Shila Ofek-Koifman *IBM Research Division, Haifa Research Laboratory, University Campus, Mount Carmel, Haifa, 31905, Israel (shila@il.ibm.com).* Ms. Ofek-Koifman has been working for IBM since 2004 at the Haifa Research Laboratory in the area of information retrieval and has been managing the Information Retrieval Systems group since 2006. She received her B.Sc. degree in information systems engineering and her M.B.A. from the Technion-Israel Institute of Technology, Haifa, Israel in 1997 and 1999, respectively. Her research interests include enterprise search, social search, and Web-feeds search and discovery. Ms. Ofek-Koifman is the coauthor of several conference papers and patents in the area of information retrieval. Prior to her work in IBM, she worked in several Israeli technology companies.

Deepak P *IBM Research Division, India Research Laboratory, Embassy Golf Links Business Park, Block D, Bangalore, India 560071 (deepak.s.p@in.ibm.com).* Mr. P works in the area of information management at the IBM India Research Laboratory, Bangalore, India. He has a B.Tech. degree in computer science from Cochin University, Kerala, India and an M.Tech. degree in computer science from Indian Institute of Technology, Madras, India, and is currently pursuing a Ph.D. degree from Indian Institute of Technology, Madras, India. More information can be found at <http://www.research.ibm.com/people/d/deepaksp>.

Chang-shing Perng *IBM Research Division, Thomas J. Watson Research Center, 19 Skyline Drive, Hawthorne, New York 10532 (perng@us.ibm.com).* Dr. Perng is a Research Staff Member at the IBM T. J. Watson Research Center. He received his Ph.D. degree in computer science in 2000 from the University of California, Los Angeles, and has been at IBM since that time. His research theme is applying data-mining technology to system and service management. His current research interests include temporal data mining, autonomic computing, and intelligent system management design.

Haggai Roitman *IBM Research Division, Haifa Research Laboratory, University Campus, Mount Carmel, Haifa, 31905, Israel (haggai@il.ibm.com).* Dr. Roitman is a Research Staff Member at the IBM Haifa Research Laboratory. He works in the Information Retrieval Systems group. He received a B.Sc. degree in information systems engineering in 2004 and a Ph.D. degree in information management engineering in 2009, both from the Technion-Israel Institute of Technology. His main research interests are at the boundary between dynamic data management (e.g., Web monitoring) and content management (e.g., content analysis and content dissemination networks), and in Web 2.0 data management (e.g., data gathering, tagging, and performance). He is also an adjunct lecturer in the Faculty of Industrial Engineering and Management, Technion.

Christopher Ward *IBM Research Division, Thomas J. Watson Research Center, 19 Skyline Drive, Hawthorne, New York 10532 (cw1@us.ibm.com).* Dr. Ward is a Research Staff Member and Manager in the Service Delivery department at the IBM T. J. Watson Research Center. He joined IBM in 2000 and is most recently responsible for innovative functionality in service management with an emphasis on applicability to enterprise environments. Since joining IBM, he has received various achievement awards, has chaired selected standards committees, and has published several technical papers. Prior to joining IBM, Dr. Ward was head of the Advanced Networking and Distributed Control Group and a Senior Member of the Technical Staff at the Sarnoff Corporation. Prior to his employment at corporate research facilities, he held academic appointments at City University of New York and Auburn University. Dr. Ward has published more than 60 papers addressing a variety of computer science problems, is author or coauthor of numerous patents, and is a Senior Member of the IEEE. He received a Ph.D. degree in computer science from the University of Florida in 1988.

James Young *IBM Global Technology Integration & Management Competency, Toronto, Canada (youngj@ca.ibm.com).* Mr. Young is a technical team lead for global Web portal projects at IBM. He has worked in North America and Europe on large IT infrastructure projects and has been involved with the WebSphere portal since its inception. He focuses on architectural issues and development challenges and has authored many components of the Service Delivery Portal. His technical focus is on Web 2.0, J2EE, and advancing the user experience for portal users. He holds a U.S. patent for a model-driven portlet development methodology.