IT Autopilot: A flexible IT service management and delivery platform for small and medium business

S. Mastrianni
D. F. Bantz
K. A. Beaty
T. Chefalas
S. Jalan
G. Kar
A. Kochut
D. J. Lan
L. O'Connell
A. Sailer
G. Wang
Q. B. Wang
D. G. Shea

IT Autopilot is a flexible architecture to support the delivery of information technology (IT) systems management services. Complex services that involve several tools require integration between the tools and automated processes that can invoke multiple tools. Designed primarily for the small and mid-sized enterprise, the architecture of IT Autopilot allows it to be deployed as a set of local and remote services delivered by the enterprise or by service providers as a flexible and extensible service offering. The IT Autopilot integrated IT service management platform is able to combine different tools and services to create specific, customized IT service solutions. Using the analogy of an autopilot on an airplane, the pilot first performs a set of manual operations to get the airplane off the ground and flying. Next, the autopilot is engaged to carry on normal flight operations. In our vision, there is an initial manual configuration step before IT Autopilot is enabled to take over and maintain the customer's normal IT operational state. In this paper we explain our vision and describe the prototype system we have implemented.

INTRODUCTION

In today's economy, the majority of professions require extensive use of computer resources. Whether the computer is used for simple data entry or complex analysis, computers are essential to support productivity and provide services both inside and outside an enterprise. A mid-sized or large enterprise may support hundreds or even thousands of computer systems. To manage these systems, a proper infrastructure architecture with well-defined policies and processes for systems management is required, as are the proper level of security and disaster recovery systems. Systems management in these environments requires a staff of well-trained information technology (IT) professionals with the proper skills and tools, who adhere to corporate policies and implement best practices.

©Copyright 2007 by International Business Machines Corporation. Copying in printed form for private use is permitted without payment of royalty provided that (1) each reproduction is done without alteration and (2) the Journal reference and IBM copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free without further permission by computer-based and other information-service systems. Permission to republish any other portion of the paper must be obtained from the Editor. 0018-8670/07/$5.00 © 2007 IBM

MICROSOFT CORP.
EXHIBIT 1011

Authorized licensed use limited to: Klarquist Sparkman LLP. Downloaded on November 24,2025 at 18:11:11 UTC from IEEE Xplore. Restrictions apply.

Successful systems management

Key to a successful systems management implementation is the maintenance of a consistent software and hardware configuration throughout all deployed systems and strict adherence to a set of policies and procedures. There is a strong contrast between large and mid-sized enterprises when it comes to implementing systems management practices, policies, and procedures. Computer systems in large-enterprise environments are usually tightly controlled to prevent the installation of programs or components that might affect the proper operation of the systems and to minimize the chances of a conflict with any future application or with operating-system updates. The IT department of larger enterprises is of sufficient size and skill to establish policies for computer access after conferring with management and can deploy these policies effectively throughout the company's network. In addition to locking down its computer systems, the company might also attempt to limit potential problems by limiting users' access to certain Web sites or external networks.

In the larger enterprise, when a problem occurs, users are directed to submit a problem report or trouble ticket to a support system. Once in the support system, the problem is logged and tracked to ensure that it gets resolved. This usually involves a workflow engine configured to reflect the practices of the company, often involving a tiered help-desk support system whose higher-level support professionals are trained by vendors. Large-enterprise customer support teams collaborate with vendors to troubleshoot and develop software patches or administrative fixes. In doing so, the large enterprise is indirectly accessing the deep knowledge bases of the vendor's technical support organization. In the case of an update or fix, the IT organization first examines the fix for applicability in the company's environment. The fix is manually tested on a few noncritical machines to make sure it does not cause any compatibility problems or change the behavior of the target system in some unwanted way. The fix may then be deployed electronically to several test machines on the network, and the IT staff will observe the results. After they are convinced that the deployment is necessary and that it will not cause any other problems or side effects, they may then schedule the fix to be delivered during nonworking hours to provide the least amount of disruption.

The IT organizations in large enterprises often do not want anything to be done to their managed computer systems without their express approval. They want to maintain positive control over the configuration of their systems. Such organizations prefer that a responsible person approve configuration changes, even if a systems management tool could accomplish the changes automatically.

Larger enterprises also differ from small and medium businesses (SMBs) in that their IT organizations can choose systems management tools individually through an orderly evaluation process. Implementing their strategy may require the purchase and installation of various tools and components from different software vendors to provide the optimal solution. The emphasis is on the suitability of each tool rather than on its potential for integration.

SMB IT management—Problem statement

The typical SMB usually supports from 50 to 2500 users. Yet unlike the large enterprise, an SMB may not have a separate IT department. The company may be so conscious of costs that the systems management tasks are performed by the owner or principal, or perhaps by a friend or acquaintance. As the company becomes larger, these positions are usually filled by someone else in the company who has some computer skills and does it part-time, in addition to his or her other responsibilities. If the company continues to grow, the tasks eventually become substantial enough to require dedicated support staff, often augmented with expensive contract personnel or by service firms. The head of the IT department or group may not have formal training in systems management and ends up being in charge by virtue of having been there the longest.

The computer systems in this environment usually consist of a mixed collection of off-the-shelf and custom-built systems of various manufacture and age, often with different operating systems and applications. Replacing or adding computers is expensive, and many business owners are reluctant to purchase something they might view in the same way that they view a broom or a fax machine. In this environment, systems are often not up-to-date and may contain viruses, worms, key loggers, and spyware. They may be running obsolete software, which not only compromises the integrity of the system, but the integrity of the entire network and

Authorized licensed use limited to: Klarquist Sparkman LLP. Downloaded on November 24,2025 at 18:11:11 UTC from IEEE Xplore. Restrictions apply.

every system attached to it. In this environment, where no consistent system configuration is maintained, establishing an infrastructure suitable to install a systems management solution can be difficult. Therefore, just the discipline of establishing consistent hardware and software standards may provide significant benefit to the organization.

In the SMB environment, there are usually no formal procedures to review, test, or install updates. Users are encouraged to update their systems whenever updates become available. They are usually encouraged to update their virus and spyware signatures and to run their antivirus and spyware-detection programs at least once a week. In larger companies, these tools are often set up to automatically download their updates and run them at scheduled times, but even in these circumstances, when a problem is detected, the user is usually required to take actions to resolve it.

As in any commercial computing environment, the proper infrastructure and procedures must be in place to provide for the recovery of data in the event of a catastrophic failure. This should include regular automatic backups and archival services to ensure the integrity of the backups.

A key factor in maintaining the integrity of any systems management infrastructure is the employment of best practices. Although the IT organization in a large enterprise is intimately aware of best practices and how to implement them, the physicians in a 150-person medical practice are much less likely to be concerned with best practices for IT. It is important, therefore, that the systems management tools and components deployed in an SMB environment embody best practices and do not require IT administrators to manually install, configure, or maintain systems. In certain mid-sized business environments, these best practices should be implemented and enforced without human intervention. In these environments, it is beneficial for system managers to interact with their tools as if they were engaged in a joint activity with the system.[1]

Some mid-sized businesses have an IT organization responsible for systems management. The IT staff understands their company's business and how IT affects their company's "bottom line." In this situation, systems management policies and procedures are implemented in such a way as to provide the least amount of disruption to the business. Introducing a level of systems management automation is likely to generate a great deal of skepticism among IT professionals, many of whom have "been there, done that." Because many have seen even the most innocuous change cause major problems, they are not comfortable with allowing systems management to be automated. They do want their systems to be monitored, and they want to be informed if any problems are detected, but they also want the ability to review the potential solutions or actions before these solutions or actions are implemented.

SMBs that do not have the benefit of a skilled or knowledgeable IT staff prefer to automate as many operations as possible so that they can concentrate on business instead of IT administration. Unlike large enterprises, most mid-sized businesses view autonomic systems management as a requirement, provided the automated actions do not cause problems or business disruption. They do not have the resources or skills to understand the details of operating individual tools and maintaining their IT infrastructure, nor do they appreciate the potential for best practices to keep their environment running smoothly. As a result, these businesses usually opt for total solutions—complete packages that come ready to plug in and have the ability to be configured and operate in an integrated manner.

Systems management tools and processes for the SMB market should therefore accommodate a broad range of options for how they are deployed and configured. The degree of integration and automation, the selection of individual systems management tools, and the choice of best practices should all be systematized and selectable by the customer. Our research in this area has led us to conclude that existing systems management tools, suites, and packages do not meet these needs.

Goal

IT Autopilot is a unified architecture for IT services specific to systems management. Its purpose is to enable a higher level of systems management process complexity, especially those processes involving the use of multiple systems management tools. These partially or fully automated processes can be invoked by users and administrators without the need for them to use the systems management

Authorized licensed use limited to: Klarquist Sparkman LLP. Downloaded on November 24,2025 at 18:11:11 UTC from IEEE Xplore. Restrictions apply.

tools directly. IT Autopilot can manage the flow of data among the various tools and include provision for exceptional conditions. IT Autopilot promises integrated and reusable processes, policies, and knowledge.

As mentioned earlier, SMBs often do not have the proper staffing or expertise to develop and implement systems management policies and procedures. They generally do not have time to learn complex tools for managing their systems, nor do they have the knowledge to implement and enforce best practices for systems management. IT Autopilot scans the customer's environment to determine the types of systems and applications on the network and assists with the creation and implementation of systems management policies based on best practices. Thus, IT Autopilot analyzes the systems and provides a set of procedures to manage the infrastructure.

Feedback from our customers indicates that they want total solutions that can be configured and maintained by individuals who are not IT specialists. Rather than purchasing separate tools from different vendors, these customers have told us that they would prefer to purchase a solution from a single vendor that presents a common, integrated interface and abstracts the underlying tools used to perform the systems management functions. IT Autopilot provides an architecture that abstracts the underlying systems management tools. Built primarily on existing management tools and technologies, IT Autopilot integrates these technologies into a single, cohesive platform capable of delivering a wide variety of IT services and scenarios. Based on the autonomic computing MAPE (monitoring, analyzing, planning, and execution) model,[2] IT Autopilot instantiates policies that are customized for the customer's environment and that adhere to best practices. Customers do not have to understand the various tools and technologies required to perform the systems management tasks, as IT Autopilot assists them by following well-defined processes and policies.

In the remainder of this paper, we continue with a structural and functional description of IT Autopilot, its integration architecture, and its services. We then discuss potential services that cover IT requirements specific to SMB in greater detail. We describe an initial prototype and potential scenarios that the prototype is intended to address. We investigate one of those scenarios, discovery of the environment and essential IT service deployment, to show how the integration architecture and services work together to implement an end-to-end solution. We discuss different forms of service delivery based on IT Autopilot, including the use of a virtual-machine-based appliance. Finally, we discuss related work and topics for future research.

INTEGRATION ARCHITECTURE FOR SERVICE DELIVERY

IT Autopilot consists of an integration architecture and an open-ended set of IT Autopilot services. The integration architecture must allow its components to execute and interact in ways required to support complex systems management tasks. It must be possible to initiate, terminate, and control the execution of a component, and there should be no constraints on the platform (hardware or operating system) on which a component runs. It is very desirable to minimize the effort required to add a new component or replace an existing one without recycling the system as a whole. Components may be remote, and processes may require tasks that can only be performed manually. These requirements are typical of those addressed by service-oriented architectures (SOAs).[3]

IT Autopilot architecture

An SOA has been described as a standards-based, loosely coupled, stateless, coarse-grained system, often adhering to the Web Services (WS) standards.[3] This approach is ideal for IT Autopilot, with the addition of a workflow and policy engine, component adapters, stateful services (those that maintain data about past actions and results), and knowledge repositories. Conceptually, IT Autopilot is separate from the IT systems being managed and is connected to the managed environment by means of its component tools. In practice this separation may only be conceptual because of the requirements of the individual tools.

Figure 1 shows the IT Autopilot integration architecture and services and the systems management tools that perform management actions on the managed system. The adapter library, one of the IT Autopilot services, is shown as being hosted remotely. The non-WS-enabled systems management tool in the middle is shown as wrapped by an adapter to achieve the necessary compatibility with

Authorized licensed use limited to: Klarquist Sparkman LLP. Downloaded on November 24,2025 at 18:11:11 UTC from IEEE Xplore. Restrictions apply.

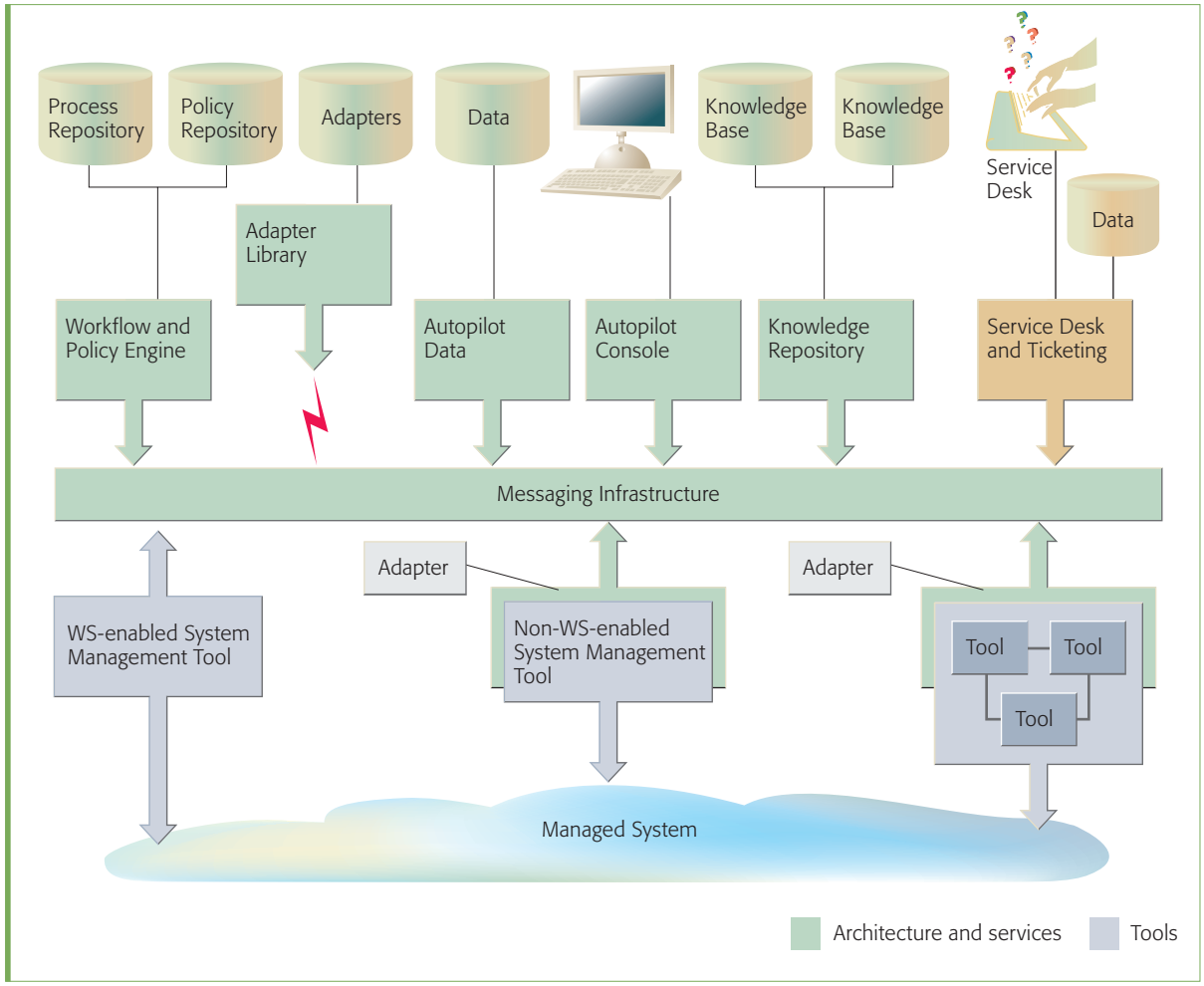

Figure 1
IT Autopilot, tools and the managed system

IT Autopilot. This tool was not designed initially to make its interfaces accessible to control and data as WS. The rightmost gray box represents a tool suite, where a proprietary integration architecture connects the three tools of the suite. The suite as a whole is wrapped by an adapter, which is required for inclusion in the Autopilot SOA.

This is an SOA, specifically, an Enterprise Service Bus,[4] based on WS standards. The messaging infrastructure in the figure may be based on something as simple as direct local HTTP (Hypertext Transfer Protocol) or as sophisticated as secure messaging and queuing, depending on the packaging of tools and IT Autopilot services. Due to loose coupling and no requirements for a common platform for the tools, adding or substituting a tool is simpler than in tool suites where all tools share a common platform. The adapter library houses prebuilt adapters for commonly available tools and suites.

The IT Autopilot SOA is similar to SOAs typically used for enterprise application integration. Its primary goal is the integration of loosely coupled services. Data does flow between services to enable the implementation of complex cross-tool processes. IT Autopilot also operates under implicit and sometimes explicit time constraints because the processes it automates affect the availability and integrity of the enterprise IT infrastructure. When reacting to a security threat or to a failure, IT Autopilot must act quickly to contain damage and to restore the IT infrastructure to health.

Authorized licensed use limited to: Klarquist Sparkman LLP. Downloaded on November 24,2025 at 18:11:11 UTC from IEEE Xplore. Restrictions apply.

There are three new contributions made by IT Autopilot. The first is its workflow-based control and the orchestration of its services so that changes can be effected in a managed system. The second is its automation of certain IT management services based on the autonomic computing MAPE model. The third is the customization of its default policies to match specific SMB requirements.

The choice of systems management tools for an automated systems management solution such as Autopilot is an important one. Many smaller businesses are unable to invest in the level of training required for their IT staff to master a complex, capable tool. They turn to simpler, more easily understood tools with less capability and simpler user interfaces. IT Autopilot favors tools with complete functionality. It hides the complexity of the tool by invoking it programmatically as a process step. The details of how the tool is controlled can be captured either in the process definition or in an associated knowledge base. Complex, powerful tools do come at a price: They may not integrate well with other tools, they may be difficult to configure to a particular environment, and their initial purchase costs may be high.

IT Autopilot services

In addition to its integration architecture, IT Autopilot includes services for knowledge maintenance, processes and policies, a console, IT Autopilot data, and the service desk. In each case, these services are specific to the provision of complex cross-tool systems management tasks. We expect that the set of services will broaden in the future.

Knowledge maintenance

IT Autopilot can provide assistance to solve problems and respond to service requests by using data contained in the knowledge repository, which stores solutions generated by users such as engineers and administrators. There are multiple potential solutions offering this service that could be integrated in our platform; for example, the IBM Tivoli* Monitoring[5] problem determination workflow functionality.

Processes and policies

Systems management tasks in the IT Autopilot architecture are represented as process definitions, modified by policies and executed by a workflow engine. This way of representing a task is very flexible, maintainable, and extensible. Process definitions represent the business processes of IT management; that is, if you think of IT as a business within a business, its business processes are IT Autopilot processes.

For our purposes, a *process* represents a set of systems management actions that, taken together, perform a systems management task. A typical task might be to add a new user to an IT environment or to replace a given server with a spare. Processes represent a change in the configuration of an IT infrastructure and thus must be undertaken with care. Whenever possible, processes must allow the configuration of the IT infrastructure to be restored to what it was before the process was performed.

A process consists of a set of steps that can be performed sequentially or simultaneously. Steps themselves can be processes, and their results can be used in decisions concerning which step is to be performed next. In IT Autopilot, steps that are not defined as processes are performed either by Web Services (WS) or by purpose-written code fragments.

In the IT Autopilot architecture, processes are represented as executable code (*hard-coded processes*) or as described in the WS-BPEL (WS Business Process Execution Language) standard[6] as augmented for the IBM WebSphere* Process Server.[7] A visual representation of a process can be constructed by using various tools, including IBM WebSphere Business Modeler.[8] WebSphere Business Integration Collaborations[9] provides prebuilt templates for business processes that can be customized in controlled ways. Templates of this kind are an ideal representation for IT Autopilot generic systems management processes. Templates can be helpful to enforce adherence to best practices. One source of best practices is the IT Infrastructure Library** (ITIL**),[10] whose section, *Information and Communications Technology Infrastructure Management*,[11] is the subject area relevant to IT Autopilot.

For our purposes, a *policy* is a customer-specific guideline that is consulted by a process. Computer-based representations of policies have great potential to improve the customizability of services delivered with IT Autopilot. (This is an aggressive goal, requiring comprehensive and general process

Authorized licensed use limited to: Klarquist Sparkman LLP. Downloaded on November 24,2025 at 18:11:11 UTC from IEEE Xplore. Restrictions apply.

definitions, discovery of all managed resources and their dependencies, a comprehensive knowledge base, and the like.) Relevant computer-interpretable policy representations include Ponder[12] and Policy Management for Autonomic Computing (PMAC),[13] among others. The Web Services Policy 1.2—Framework (WS-Policy) standard[14] relates much more to the specification of the nonfunctional attributes of WS than to guidelines. The natural site for policy residence is the workflow engine, but unfortunately, no known workflow engine incorporates this capability.

Policy consultation can be done by including an additional step in the workflow at each point where a policy applies. Alternatively, policies can be consulted by embedding "decision points" in the code of a tool or a hard-coded process.

A process instance represents a systems management task, created by instantiating a process definition. The IT Autopilot architecture selects a process definition to be instantiated in two ways: either through an interaction with a system administrator using the IT Autopilot console or programmatically, through process invocation as a step in another workflow. This programmatic selection can be quite involved, as in the case of problem determination, where a process definition is selected on the basis of its being the best remediation of a problem.

IT Autopilot console

The IT Autopilot console aggregates information from the constituent tools and enables the system administrator to interact with IT Autopilot and the underlying tools. It allows the administrator to view and change the processes and policies that describe how the systems management tasks are performed, and it provides a display of the systems management tasks that have been performed along with their effect on the system. The IT Autopilot console is developed by using the Tivoli Enterprise Portal Server (TEPS), a component of IBM Tivoli Monitoring.[5] TEPS is a Web-based technology.

IT Autopilot data

IT Autopilot relies primarily on data maintained and managed within its constituent tools. This is both an advantage and a weakness: It is an advantage because data maintained within a tool is accurate; the tools take care of the data maintenance (e.g., caching, updating, and replication). The weakness is that the tool may not make all needed data accessible at the IT Autopilot level, and it may create only internal repositories. Also, the tools may store overlapping data without the capability to synchronize it or structure it with consistent schemas. A prime example of tool-maintained data that is accessible to IT Autopilot is the IBM Tivoli Change and Configuration Management Database (CCMDB),[8] used and updated by several Tivoli tools. As IT Autopilot evolves, we expect its data needs to increase, and our dependence on tool-managed private data may prove problematic. In this case, an additional layer on top of existing data repositories could provide the necessary data federation and management.

Integrating the service desk

Service desks are generally a combination of manual and automated services where end users and service desk personnel enter and track IT incidents. This service also plays an important role in IT Autopilot by maintaining historical customer incident data. The interaction between this service and the other IT Autopilot services is two-way: IT Autopilot shares the environment-monitoring and policy repository, and the service desk component shares whether an issue is a consequence of a policy deviation.

Issues may be created in three ways: through a direct end-user interaction with the service desk portal, through direct end-user interaction with service desk personnel, or through direct detection by IT Autopilot.

So far, we have identified potential players for the basic services that can be integrated in our IT management platform. In the next section we describe the details of the implementation of our prototype.

INITIAL EXPERIENCES

Our first step was building a prototype of the IT Autopilot. The objective of the prototype was to demonstrate the benefits of functional integration of two management tools—IBM Tivoli Monitoring (ITM)[5] and Tivoli Application Dependency Discovery Manager (TADDM)[15]—and to create the basis to implement more advanced services in the future. The Tivoli Enterprise Portal (TEP) is used as the

Authorized licensed use limited to: Klarquist Sparkman LLP. Downloaded on November 24,2025 at 18:11:11 UTC from IEEE Xplore. Restrictions apply.

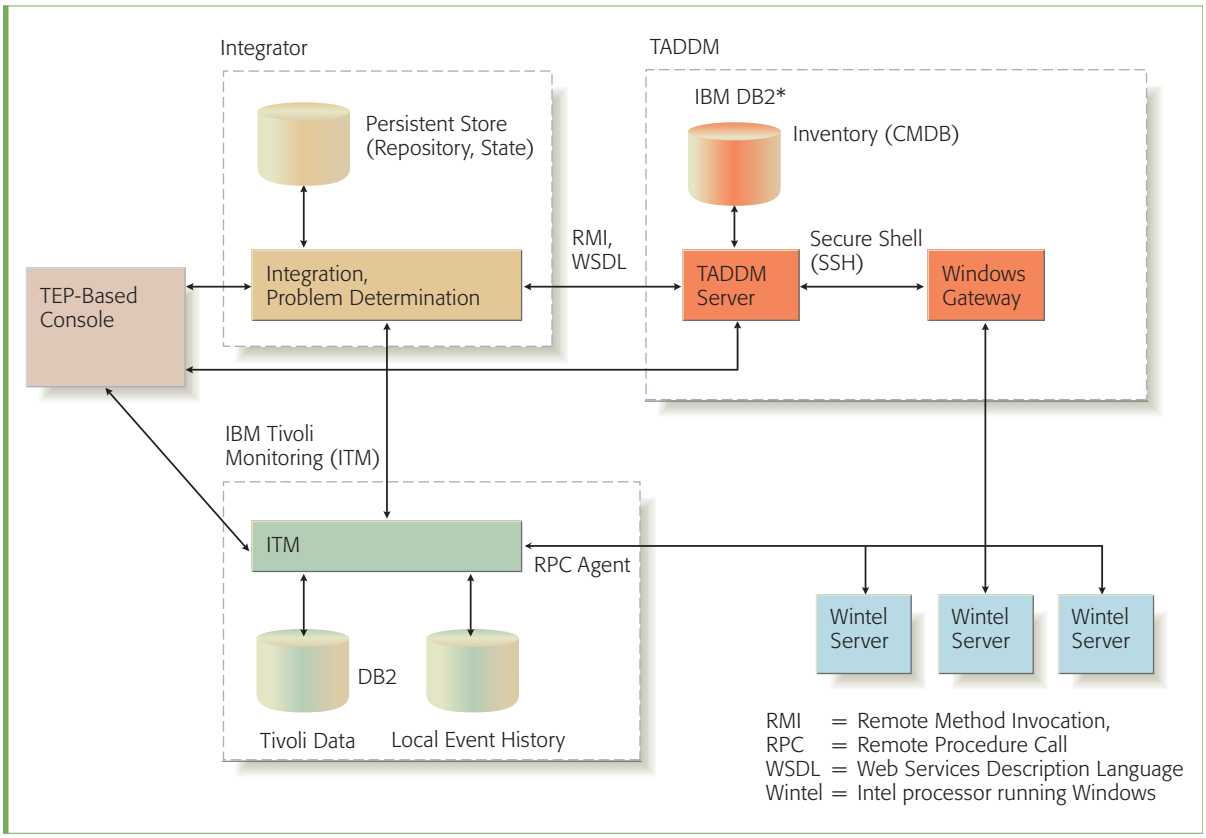

Figure 2
Initial IT Autopilot configuration

integrated console for both data and control of the system. At the same time, a self-service portal is provided to the customer. In this portal, the customer can view the description of the process of the controller and can enable or disable a specific process within the controller. Each of the processes corresponds to specific systems management scenarios. In this paper, there are two scenarios selected to be implemented. The first concerns automated ITM agent deployment and configuration for the customer's initial environment and for newly introduced resources. The second concerns policy-based recommendations for the provisioning of essential IT services, such as security and backup, based on the initial business-critical application discovery.

The initial prototype integrates the two tools mentioned earlier by using a WS-based approach. Both ITM and TADDM make programmatic control and monitoring available as WS, but not all of the required functions are available through the WS interfaces. Thus, the tools were extended to enable the required functionality.

Figure 2 is an overview of the initial IT Autopilot configuration. The Integrator is the core of the system. It uses WS to communicate with TADDM and ITM and makes a control interface accessible to the graphical user interface (GUI) of the console. TADDM consists of the main server, the CMDB (configuration management database), and the Microsoft Windows** gateway server enabling discovery of Windows systems through the Windows gateway. ITM consists of the main server, unified console, distributed agents, and several databases (including Tivoli Data Warehouse and Enterprise Portal Database). The current version of IT Autopilot uses TEP as the presentation layer. *Figure 3* shows the main components of IT Autopilot and the interaction among them.

In the scenario, suppose the system is connected to the customer's network ("managed system"), as in

Authorized licensed use limited to: Klarquist Sparkman LLP. Downloaded on November 24,2025 at 18:11:11 UTC from IEEE Xplore. Restrictions apply.

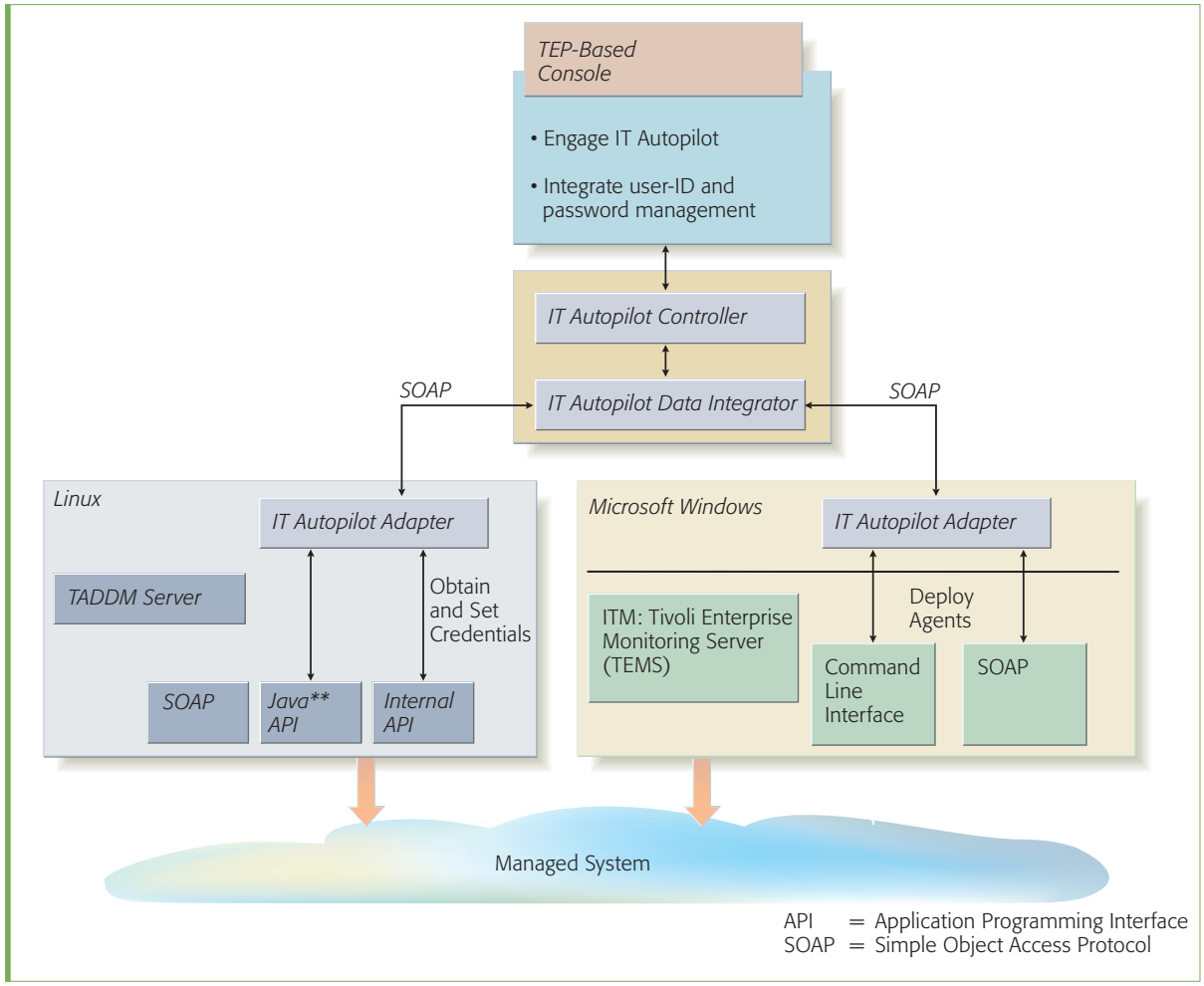

Figure 3
Interaction among IT Autopilot components

Figure 3. The IT administrator supplies the Internet Protocol (IP) configuration parameters of the customer's network and the credentials required to access the systems (to enable automated discovery and agent deployment). The administrator then defines the scope (in terms of the range of IP addresses or subnet mask) that IT Autopilot should cover. Next, IT Autopilot is started, and the IT Autopilot Integrator triggers the TADDM discovery process over the defined scope. As a result, the CMDB is populated with the details of the customer's environment. The IT Autopilot Integrator analyzes the discovery results and extracts the information concerning the physical infrastructure and installed software in the environment. ITM agents are then deployed based on the discovered device types, operating systems, and software components. Additionally, based on the results of the business-critical application discovery, IT Autopilot provides recommendations on security and backup services, basing its suggestions on IT best practices. Depending on the customer selections, the newly recommended services (e.g., Tivoli Storage Manager) are deployed. Finally, the IT Autopilot Integrator initiates the ITM monitoring process.

The IT Autopilot Integrator periodically triggers TADDM discovery. When a new resource is introduced in the environment, it will be discovered; the Integrator will learn of its presence and will initiate the ITM agent deployment to that newly discovered resource. As a result, the monitoring components are automatically deployed and configured without further intervention from the oper-

Authorized licensed use limited to: Klarquist Sparkman LLP. Downloaded on November 24,2025 at 18:11:11 UTC from IEEE Xplore. Restrictions apply.

ator. If needed, the IT management services update waits for operator approval before deployment.

IT AUTOPILOT SCENARIOS

To fully exercise the potential of the IT Autopilot prototype, several other scenarios, discussed in the next subsections, can be developed. The criterion for selecting the scenarios was to demonstrate customer value by exploiting the integration made possible by IT Autopilot and the ability to define complex, cross-tool management processes. The Integrator can bring together disparate data from multiple sources, and it can form hypotheses and draw conclusions based on known, observed, or calculated results.

Configuration error

Through monitoring, IT Autopilot detects that a system cannot connect to the Internet. Using the TADDM inventory, the configuration of the disconnected machine is compared with a valid configuration, represented by a template. If differences are found, a process is dispatched to make the change, and the problem machine is updated to the proper configuration. If there is no difference, the system searches for a machine that is on the same subnet as the problem machine. If one is found, the system runs a test on it to see if it can access the Internet. If the test machine also cannot connect to the Internet, then IT Autopilot verifies that the configuration of the test machine is also correct. If the settings are correct, IT Autopilot determines that the problem exists at some point on the network and generates a network problem event. The network problem event is detected and routed to the proper network problem resolution module, or optionally, logged on the IT Autopilot console as a problem that requires human intervention to resolve.

Transaction error

IBM Tivoli Composite Application Manager (ITCAM)[16,17] detects an issue with a transaction. IT Autopilot uses the TADDM inventory and dependencies data to detect possible configuration changes and reports them to the Integrator. In addition, IT Autopilot collects data from each individual resource participating in the transaction by using the ITM non-operating system (non-OS) agents (ITCAM for HTTP,[16] ITCAM for WebSphere,[16] and IBM DB2* agent[18]). IT Autopilot uses the TADDM dependency data together with the ITM non-OS agent monitoring data and identifies the resource contributing to the transaction failure.

Security

An optional network monitor module that monitors and tracks network traffic on the local network detects that a system has been infected with a virus. The network monitor uses a combination of packet inspection, correlation, and pattern analysis to determine the type of infection. Using the connectivity and security information from TADDM, IT Autopilot connects to the infected machine, forces an update of the antivirus and spyware signatures, and then forces an antivirus and spyware scan of that machine to remove or repair the offending threat. IT Autopilot then, as a precautionary measure, pushes those updates to all of the systems on the local network and launches scans on those systems.

Policy and security

IT Autopilot uses ITM, other means, or both to detect that sharing restrictions have been violated or that there are other permissions which violate policy on a machine. Using the TADDM inventory, the configuration of the machine is compared with the authorized configuration. If differences are found, then the appropriate changes are made to ensure compliance with the policy.

DELIVERING SERVICES WITH IT AUTOPILOT

Ultimately, the purpose of IT Autopilot is the provision of services. Its flexibility permits these services to be delivered in various ways. This section discusses some of those ways and what must be done to enable them.

Service offerings

IT Autopilot is designed to be installed, configured, and operated in several different ways: as an appliance, as a virtual appliance packaged in a virtual machine, or as software that runs on hardware provided by the customer or a business partner. In the first case, IT Autopilot is installed as a stand-alone appliance with all of its services provided locally. (*Appliance* means that the software is delivered on a self-contained hardware platform, preconfigured and self-configuring to the customer's managed system.) In this case, IT Autopilot relies on preconfigured templates for problem detection and resolution. The knowledge base is provided at the time of installation. IT

Authorized licensed use limited to: Klarquist Sparkman LLP. Downloaded on November 24,2025 at 18:11:11 UTC from IEEE Xplore. Restrictions apply.

Autopilot can be updated by downloading an updated problem detection and resolution database.

In one potential business model using an appliance model of service delivery, the appliance is offered at a nominal cost, and the user pays a subscription fee that covers periodic updates. The updates can be made available for download and installation by the customer or automatically installed when they become available. This is analogous to the model adopted by cellular telephone companies, which have opted to focus on profits from providing a service.

Although IT Autopilot focuses on IT systems management, the infrastructure is designed to permit other services to be offered as well. The IT Autopilot plug-in architecture allows for various compatible tools, components, and services to be added or aggregated by the customer, business partner, or provider. This makes it possible for the customer to add value from a trusted partner of choice. The plug-in architecture also makes it possible for a user to select and deploy new services from a catalog of available services. (These services may be limited or constrained by local regulation or governance and by resource availability, such as broadband access.)

Dynamic service delivery

IT Autopilot delivers services that are managed by business policies and business objectives. If a company requires maximum productivity during the hours of 8:00 a.m. to 5:00 p.m., IT Autopilot postpones updates and other disruptive activity until a more convenient time. IT Autopilot can be configured to provide a more aggressive set of actions by checking systems each hour if, for example, the customer has been experiencing frequent virus outbreaks or spyware infections. A tax accountant business with a backup-and-restore service installed might specify that all systems related to accounting tasks and critical to the business must be backed up each hour, whereas other less critical systems can be backed up only once per week.

IT Autopilot as an appliance

Software appliances are common to the industry today. The idea of having a turnkey solution that is fully configured and needs only to be plugged in (e.g., to electricity and the network) in order to provide a stated service is commonplace. The ideal software appliance should be as easy to use as plugging in a toaster at home.

It is a desirable goal to build IT Autopilot to be a self-sufficient appliance. The intent is to have a self-contained solution by integrating those IT tools that provide valuable services with a minimum of human intervention. In the prototype work, we have kept the appliance model in mind and have therefore used automation to limit the points where manual configuration and decision making are required.

Appliances, whether hardware or software, often provide the means to be upgraded in the field, as it is much easier to do this than to have a product recall when a major defect is found. For IT Autopilot, we envision upgrading being done as a subscription model. Enhancements to a variety of components, such as newer versions of IT tools and services or knowledge data for problem determination, could be provided by IBM by means of a subscription-based update facility.

To implement the model of IT Autopilot as a hardware-based appliance, one approach is to preconfigure its services and tools on the blades of an IBM BladeCenter*. Installation at a customer's premises consists of connecting the various systems management tools to the customer's network and configuring them appropriately. The internal configuration of IT Autopilot can be performed beforehand.

Compared with traditional customized methods of application configuration and deployment, an appliance model is simpler, but it may provide less flexibility. Although the appliance model provides the simplicity of preconfiguration, there is still flexibility during the creation of the preconfigured appliance. This flexibility is supported through a repository of preconfigured appliance templates and images.

The appliances are designed to make a limited set of configuration points accessible for customization. These characteristics allow the virtual appliances to be easily cloned from a repository and customized, providing a radically simplified installation in the customer environment. The customization opera-

Authorized licensed use limited to: Klarquist Sparkman LLP. Downloaded on November 24,2025 at 18:11:11 UTC from IEEE Xplore. Restrictions apply.

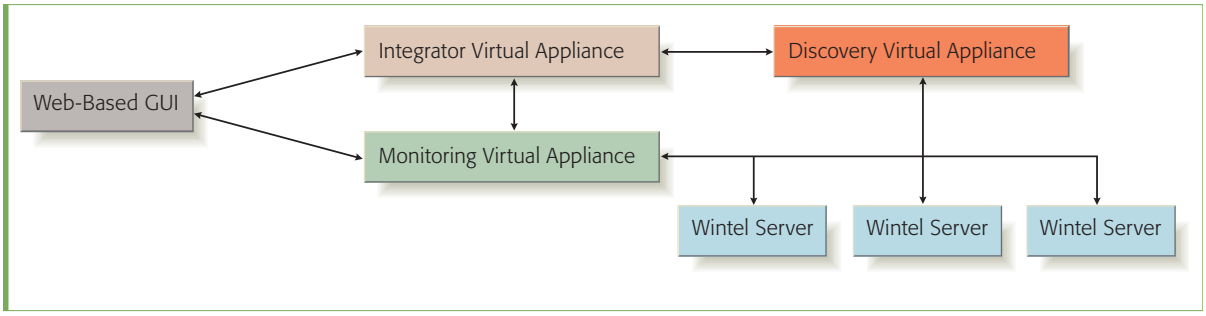

Figure 4
IT Autopilot deployment model as a VM appliance

tions can be fully automated by using a software product or manually invoked through simple scripts.

Note that the integration of multiple IT tools may require different platforms, such as Microsoft Windows or Linux**. For example to simplify the installation of the IT Autopilot appliance we can make use of virtualization technology to co-host IBM Tivoli TADDM (which requires Linux) and IBM Tivoli ITM (which requires Windows) on a single physical server. Because we are installing virtual machines, not software packages, the configuration effort is minimized. We expand on this option below.

IT Autopilot as a virtual machine-based appliance

A virtual machine (VM) appliance is a set of VM images preconfigured to specific requirements by domain experts. Such an appliance includes the operating system with both middleware and applications already installed, configured, and tuned. As virtualization technologies mask the actual hardware, the VM appliances are easy to initiate and run and require no installation beyond that of the virtualization environment. Once that environment is in place, the appliance is started by downloading the VM image and starting the VM in the new hosting environment.

Delivering applications and solutions as preconfigured VM appliance images eliminates most of the error-prone manual steps during software installation and customization, radically simplifying the customer experience. To use VM appliances in different customer environments, some level of configuration and customization needs to be provided by making a limited number of configuration points accessible. This section discusses a very simple process and corresponding tools to enable delivery and customization of middleware and applications as virtual appliances. The objective for virtual appliance customization is to handle 90 percent of the customer requirements by conducting very simple configuration operations.

There are two phases to the creation and instantiation of a VM appliance. The first phase, *appliance creation*, creates the virtual appliance with the requisite customization capabilities. The second phase, *appliance activation*, uses the metadata and scripts in the virtual appliance package created to customize and activate a VM appliance for a new environment.

Figure 4 shows how to use the VM-based appliance model in IT Autopilot. ITM and TADDM are instantiated as VM images for customer-side delivery and customization (refer to Figure 2 for comparison).

RELATED WORK

Integrated, automated delivery of complex IT services for the SMB market is a new area of exploration. To the best of our knowledge, only point solutions have been proposed. Systems management vendors targeting large enterprises are beginning to derive new tools for mid-sized enterprises from their existing tools, but as we observed earlier, such enterprises look for turnkey solutions with much improved ease-of-use—even autonomics.

The goal of IT Autopilot is the integration of complex cross-tool systems management processes. Naik et al.[19] present an extensive review of

Authorized licensed use limited to: Klarquist Sparkman LLP. Downloaded on November 24,2025 at 18:11:11 UTC from IEEE Xplore. Restrictions apply.

approaches to the coordination of independent systems management tools.

A commonly used system design methodology suitable for system integration is SOA.[3,4] It defines a flexible architecture that is the foundation of IT Autopilot architecture. One of the possible technologies used to implement SOA is WS.[20] We follow the best practices of leveraging this technology in IT Autopilot.

SUGGESTIONS FOR FURTHER RESEARCH

The key issues in IT Autopilot concern autonomics, implementation of processes and policy, federation of tools, the appliance model, and provisions for third-party providers. There remain many questions to be answered and areas to be explored.

IT Autopilot makes use of a data repository that contains problem-detection information, weighted possible and recommended solutions, and weighted outcomes. It is possible that while attempting to fix a problem, IT Autopilot may cause another problem or set of problems to surface. These problems may be minor and perhaps can be easily fixed, but what if the solution that was applied creates an even more serious problem, one that cannot be solved by IT Autopilot and that requires human intervention? Should IT Autopilot perform the problem resolution operation in spite of the questionable outcome? If problems do occur after the solution is applied, should IT Autopilot roll back the changes just implemented? Can the changes be rolled back? And should there be policies in place that govern how and when the problem resolution should take place? Should the resolution be deferred to after-hours, or is it critical enough that current operations should be suspended?

The data repository represents knowledge. It would be desirable to share that knowledge with others so that they could benefit. The knowledge base should be capable of being updated with new scenarios and resolution data. The data could be published by the IT organization, provided by a business partner or reseller, or provided by the hosting service.

As problems are detected and fixed, the repository is updated with the latest detection and resolution data. If a resolution scenario fails, the resolution is downgraded, which allows a more favorable solution to take its place as the preferred solution. Over time, only the best solutions remain viable, in effect allowing IT Autopilot to learn from its mistakes.

To take full advantage of this feature, tools must be provided to allow the data in the repository to be edited and published in a form that a customer or IT administrator can understand. A simulator should be able to display a solution and data flow so that the solution can be modeled and tested before deployment. The simulator must be capable of doing this based not on a static template but by using the current system configuration.

The introduction of policy decision points into IT Autopilot processes makes the control of policy checking a responsibility of the process designer. This may be too burdensome in practice. If the notion of policy is found useful, then research into the automatic introduction of policy decision points into all process definitions could simplify process design.

ITIL has a CMDB as the central repository for all configuration and management information. Because IT Autopilot is a tool integration platform, different applications such as ITM, TADDM, and ITCAM all have different databases to store specific information required for their operation. Some of this information is replicated unnecessarily. Some is relative, has dependencies, and should be linked together. If it were possible to combine this information to provide an integrated view, processes could have more consistent and complete information on which to base decisions. In our future work, we intend to address the formation of a systems view of application data, integrate information and databases, build up the linkage and dependency between information providers (tools), and present a consistent view to facilitate ITSM.

Our interest in appliance models of service delivery clashes with our choice of tools, some of which are resource-intensive. These resource needs make it difficult both to supply software for customer installation and to supply hardware for a turnkey solution. We need capable, lighter-weight tools for the reduced performance needs of mid-sized businesses. We also need tools that automatically discover the customer's configuration so that the manual configuration steps can be minimized.

Third-party providers should, in theory, be able to participate in an IT Autopilot-based systems man-

Authorized licensed use limited to: Klarquist Sparkman LLP. Downloaded on November 24,2025 at 18:11:11 UTC from IEEE Xplore. Restrictions apply.

agement solution. The sticky problems of assigning responsibility, limiting resource usage, and proper billing remain to be solved, but the open integration architecture promises to enable the third-party provider.

SUMMARY

We have presented an open integration architecture to coordinate disparate systems management tools in the provisioning of complex, cross-tool systems management processes. Although valuable for many reasons, this architecture requires tools with the appropriate interfaces made programmatically accessible as WS. SOA gives us the flexibility to choose tools on the basis of their capabilities rather than have the tools dictated by the platform. The integration of the low-level components at the tools level provides an architecture that is "tool agnostic" and allows Autopilot to manage tasks according to the needs of the business rather than according to the capabilities of the tools.

This architecture includes a knowledge repository that provides expert help to detect and fix problems. The flexibility and openness of SOA makes it easy for customers, providers, and partners to contribute their unique knowledge and expertise to this knowledge base, making it possible for others to benefit from their experiences.

Having worked with both small and large companies, it has become evident to us that a systems management solution needs to support different models of automation depending on the size of the company. Smaller companies tend to require and expect more automation. They may or may not have an IT staff, and thus may not be experienced in managing their systems. The automation provided by IT Autopilot allows them to concentrate on their core business instead of on systems management tasks. Larger companies generally have an IT organization that deals with systems management issues on a regular basis. For these companies, it is important to maintain strict control of their IT environment. To ensure compliance, they often do not allow users to install applications or to change system settings. Before any changes are deployed, the proposed changes must be reviewed, tested, and approved by an IT administrator. In this case, IT Autopilot can be configured to make recommendations without providing any automation and perform selected systems management tasks only when approved. To accommodate the wide range of automation, IT Autopilot uses a combination of WS-BPEL workflows, policies, and hard-coded fragments to provide the flexibility to radically change behavior without radically changing the system.

ACKNOWLEDGMENTS

We express our thanks and appreciation to Luis Casco-Arias, Jim Crosskey, and Jim Fletcher from IBM Tivoli Division for their expertise on IT systems management and tooling for small to medium business. We also thank and acknowledge Manoj Kumar from IBM Research for his support of this effort and Pei Sun, Yu Fei Ma, and Zhong Bo Jiang from IBM China Research Laboratory for their early contributions to this project.

*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 United Kingdom Office of Government Commerce, Microsoft Corporation, Linus Torvalds, or Sun Microsystems Inc. in the United States, other countries, or both.

CITED REFERENCES

1. D. M. Russell, P. P. Maglio, R. Dordick, and C. Neti, "Dealing with Ghosts: Managing the User Experience of Autonomic Computing," *IBM Systems Journal* **42**, No. 1, 177–188 (2003).
2. *Practical Autonomic Computing: Roadmap to Self Managing Technology*, White Paper prepared for IBM Corporation, Enterprise Management Associates, Boulder, CO (January 2006), http://www-03.ibm.com/autonomic/pdfs/AC_PracticalRoadmapWhitepaper_051906.pdf.
3. M. Colan, "Service-Oriented Architecture Expands the Vision of Web Services, Part 1," developerWorks, IBM Corporation, www.ibm.com/developerworks/webservices/library/ws-soaintro.html.
4. M. Keen, S. Bishop, A. Hopkins, S. Milinski, C. Nott, R. Robinson, J. Adams, P. Verschueren, and A. Acharya, *Patterns: Implementing an SOA Using an Enterprise Service Bus*, IBM Redbooks (2004), http://www.redbooks.ibm.com/abstracts/sg246346.html.
5. IBM Tivoli Monitoring, IBM Corporation, www.ibm.com/software/tivoli/products/monitor/.
6. J. O. Kephart and D. M. Chess, "The Vision of Autonomic Computing," *IEEE Computer* **36**, No. 1, 41–50 (2003), http://www-03.ibm.com/autonomic/pdfs/AC_Vision_Computer_Jan_2003.pdf.
7. IBM WebSphere Process Server, IBM Corporation, http://www-306.ibm.com/software/integration/wps/.
8. IBM Tivoli Change and Configuration Management Database (CCMDB), IBM Corporation, http://www.ibm.com/software/tivoli/products/ccmdb/.

Authorized licensed use limited to: Klarquist Sparkman LLP. Downloaded on November 24,2025 at 18:11:11 UTC from IEEE Xplore. Restrictions apply.

9. IBM WebSphere Business Integration Collaborations, IBM Corporation, www.ibm.com/software/integration/wbicollaborations/.
10. IT Infrastructure Library, U.K. Office of Government Commerce, http://www.itil.co.uk.
11. *ICT Infrastructure Management*, IT Infrastructure Library Services, U.K. Office of Government Commerce, Her Majesty's Stationery Office, Norwich, United Kingdom (May 2002), ISBN-10:0113308655.
12. N. Damianou, N. Dulay, E. Lupu, and M. Sloman, "The Ponder Policy Specification Language," *Proceedings of the International Workshop on Policies for Distributed Systems and Networks*, Bristol, U.K. (2001), pp. 18–39.
13. IBM Policy Management for Autonomic Computing, alphaWorks, IBM Corporation, www.alphaworks.ibm.com/tech/pmac.
14. *Web Services Policy 1.2—Framework (WS-Policy)*, S. Bajaj, D. Box, D. Chappell, F. Curbera, G. Daniels, P. Hallam-Baker, M. Hondo, et al., W3C Member Submission (April 25, 2006), http://www.w3.org/Submission/2006/SUBM-WS-Policy-20060425/.
15. IBM Tivoli Application Dependency Discovery Manager, IBM Corporation, www.ibm.com/software/tivoli/products/taddm/.
16. IBM Tivoli Composite Application Manager for Internet Service Monitoring, IBM Corporation, www.ibm.com/software/tivoli/products/composite-application-mgr-ism/.
17. IBM Tivoli Composite Application Manager for WebSphere, IBM Corporation, http://www-306.ibm.com/software/tivoli/products/composite-application-mgr-websphere/.
18. IBM Tivoli Monitoring for Databases, IBM Corporation, http://www-306.ibm.com/software/tivoli/products/monitor-db/.
19. V. K. Naik, A. Mohindra, and D. F. Bantz, "An Architecture for the Coordination of System Management Services," *IBM Systems Journal* **43**, No. 1, 78–96 (2004).
20. Web Services Architecture Working Group, World Wide Web Consortium, http://www.w3.org/2002/ws/arch/.

Accepted for publication February 6, 2007.
Published online July 13, 2007.

Steve Mastrianni
IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598 (stevemas@us.ibm.com). Dr. Mastrianni is a senior software engineer currently writing autonomic and services software. He has a B.S. degree in business administration and M.S. and Ph.D. degrees in computer science. He joined IBM Research after running a software development and consulting company for a number of years where he specialized in operating systems development, device drivers, real-time applications, data acquisition, and computer 3D graphics. Dr. Mastrianni is the author of three programming books, numerous conference papers, and over 100 articles for industry trade publications. He has 20 issued patents.

David F. Bantz
IBM Research Division, Thomas J. Watson Research Center, 19 Skyline Drive, Hawthorne, New York 10532 (dfbantz@us.ibm.com). Dr. Bantz was a research staff member from 1972 to 2005 after a short stint with a startup. He rejoined IBM in 2006 as a part-time employee. He graduated from Columbia University School of Engineering and Applied Science in 1970 with an Eng.Sc.D. degree and taught there and at the University of Southern Maine as an adjunct professor for over 30 years. He has 29 issued patents. His technical interests have always been in personal computing applications and technology, and he is currently working on automated systems management.

Kirk A. Beaty
IBM Research Division, Thomas J. Watson Research Center, 19 Skyline Drive, Hawthorne, New York 10532 (kirkbeaty@us.ibm.com). Mr. Beaty is a senior software engineer. His research and interests include systems and network management, database management systems, storage-area-network storage systems, and Lotus Notes® scalability. He joined IBM in 1981 as a systems programmer and has been developing systems management software for over a decade at both the IBM Cambridge Scientific Center and the Watson Research Center. Mr. Beaty has a B.S. degree in mathematics and computer science from Manchester College, Indiana, and a Certificate of Advanced Study in software engineering from Harvard University.

Tom Chefalas
IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598 (chefalas@us.ibm.com). Mr. Chefalas is a Senior Technical Staff Member in services technologies. He received a B.S. degree in computer science in 1988 and an M.S. degree in computer science in 1992 from New York University. He joined IBM in 1987. Mr. Chefalas is currently working on desktop management, autonomic computing, and end-user services.

Srikant Jalan
IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598 (sjalan@us.ibm.com). Mr. Jalan received a B.S. degree in engineering from the Indian Institute of Technology in 1989. He joined IBM in 1994 and is currently developing systems management software for desktops, servers, distributed systems, and grid-based systems. Mr. Jalan's interests include desktop management, end-user services, and life-cycle management.

Gautam Kar
IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598 (gkar@us.ibm.com). Dr. Kar is the manager of the Systems and Network Management Group. He has a Ph.D. degree in computer and information science from Ohio State University. He joined the Thomas J. Watson Research Center in 1983. From 1989 to 1993, he was the manager of the Distributed Applications Research Group at the IBM European Networking Center in Heidelberg, Germany. From 1994 to 1999, he was the senior manager of Network and Systems Management in the IBM Global Network Services Division. Dr. Kar's research interests include performance and availability management of distributed systems, and services.

Andrzej Kochut
IBM Research Division, Thomas J. Watson Research Center, 19 Skyline Drive, Hawthorne, New York 10532 (akoshut@us.ibm.com). Dr. Kochut is a research staff member. He has an M.S. degree in computer science from the University of Warsaw and a Ph.D. degree in computer science from the University of Maryland. Dr. Kochut's research interests include design and performance evaluation of computer systems, network protocols, systems modeling, problem determination, and distributed systems.

Dong Jun Lan
IBM Research Division, China Research Laboratory, Building 19, Zhongguancun Software Park, 8 Dongbeiwang West Road,

Authorized licensed use limited to: Klarquist Sparkman LLP. Downloaded on November 24,2025 at 18:11:11 UTC from IEEE Xplore. Restrictions apply.

Haidan District., Beijing 100094, People's Republic of China (landj@cn.ibm.com). Mr. Lan is a researcher in the High Performance Solution team at the IBM China Research Laboratory. He has an M.S. degree in electrical engineering from Tsinghua University, Beijing, China. Mr. Lan has published more than 10 papers in international conferences and holds five patents.

Larry O'Connell
IBM Global Small and Medium Business, 1551 So. Washington Avenue, Piscataway, New Jersey 08854 (loconnel@us.ibm.com). Mr. O'Connell is the director of a global team of IT architects focused on growth markets. He has an M.B.A. degree from the Stern School of Business of New York University and undergraduate and M.S. degrees in mechanical engineering from Manhattan College. He is a licensed professional engineer in New York State. Prior to joining IBM in 1998, Mr. O'Connell led the Lucent/Bell Lab Multimedia Communications team in introducing IP-based voice, video, and data collaboration products to the marketplace.

Anca Sailer
IBM Research Division, Thomas J. Watson Research Center, 19 Skyline Drive, Hawthorne, New York 10532 (ancas@us.ibm.com). Dr. Sailer joined IBM in 2003 where she is currently a research staff member in the Systems and Network Management department. She received a Ph.D. degree in computer science from Pierre and Marie Curie University of Paris in 2000. She was a research member at the Networking Research Laboratory at Bell Labs, Lucent Technology from 2001 to 2003, where she specialized in internet services traffic engineering and monitoring. Dr. Sailer's interests include Web Services architecture, system monitoring, problem determination, and self-healing technologies.

Gang Wang
IBM Research Division, China Research Laboratory, Building 19, Zhongguancun Software Park, 8 Dongbeiwang West Road, Haidan District, Beijing 100094, People's Republic of China (gwang@cn.ibm.com). Mr. Wang is a research and development engineer in the High Performance Solution team at the IBM China Research Laboratory. He received B.S. and M.S degrees in computer science from Peking University, Beijing, China in 2003 and 2006, respectively. Mr. Wang joined IBM in 2006.

Qing Bo Wang
IBM Research Division, China Research Laboratory, Building 19, Zhongguancun Software Park, 8 Dongbeiwang West Road, Haidan District, Beijing 100094, People's Republic of China (wangqbo@cn.ibm.com). Mr. Wang is a staff research and development engineer in the Distributed Infrastructure Group at the IBM China Research Laboratory. He received B.S. and M.S degrees in computer science from Harbin Institute of Technology, Harbin, China in 2000 and 2002, respectively. Mr. Wang is currently researching systems management simplification by applying virtual appliance technology to enterprise IT environments.

Dennis G. Shea
IBM Research Division, Thomas J. Watson Research Center, 19 Skyline Drive, Hawthorne, New York 10532 (dgshea@us.ibm.com). Dr. Shea is a senior manager in the Services Research department. He has B.E. and M.E. degrees in electrical engineering from Rensselaer Polytechnic Institute, an M.A.S. degree from Florida Atlantic University, and a Ph.D. degree in computer science from the University of Pennsylvania. He began his career at IBM Boca Raton working on small systems, and his technical interests have revolved around connectivity, mobile solutions, and client management services. Dr. Shea's current interest is in the area of service sciences research and its application to solving customer problems. ■

Authorized licensed use limited to: Klarquist Sparkman LLP. Downloaded on November 24,2025 at 18:11:11 UTC from IEEE Xplore. Restrictions apply.