

WebSphere Portal & Rational HATS: A SOA Starting Point

by Zachary Hanlon



IBM, the IBM logo, ibm.com, CICS, IMS, WebSphere, Rational, DB2, and Lotus are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol ([®] or [™]), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at “[Copyright and trademark information](http://www.ibm.com/legal/copytrade.shtml)” at www.ibm.com/legal/copytrade.shtml

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java is a trademark of Sun Microsystems, Inc. in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Table of Contents

About This Paper	4
Purpose and Goals.....	4
Author	4
Special Thanks	4
Executive Summary	5
Service-Oriented Architecture	5
WebSphere Portal, Rational HATS & the SOA Strategy	5
Introduction to Service Oriented Architecture.....	7
Defining SOA	7
Entry Points.....	8
WebSphere Portal on System z.....	10
What is Portal?.....	10
Portal and SOA	12
Why Portal on System z?.....	13
Rational Host Access Transformation Service	14
What is HATS?.....	14
HATS and SOA	15
Why Portal and HATS Together?.....	16
Risk and ROI.....	16
Composite Applications and Cooperative Portlets	16
Complementing SOA Entry Points	16
Improved Efficiency	17
Conclusion	19

About This Paper

Purpose and Goals

The purpose of this paper is to act as a reference for those involved in the IT architecture strategic decision making process. It will guide understanding of how WebSphere® Portal and Rational® Host Access Transformation Services (HATS) operate together as a safe and effective service-oriented architecture (SOA) entry point with quick results for IBM clients.

For those considering adopting a SOA strategy for your IT infrastructure this paper offers a basic overview of SOA, its values to the company as well as a safe entry point into that architecture through two IBM products. For those who have already decided on the SOA approach for your IT, this paper also offers a good entry point or next step for your SOA through the combination of WebSphere Portal and Rational HATS.

There are no necessary requirements to understand this paper other than a working knowledge of information technology in the business world. After reading this paper, the reader should have a basic understanding of SOA, WebSphere Portal, Rational HATS, and how they fit together as a starting point for the implementation of a SOA. This basic understanding should be enough to facilitate productive discussions and help one determine if further inquiry into these products is beneficial.

Author

Zachary Hanlon is an Americas level System z Software IT Specialist focusing on WebSphere Portal. Zachary joined IBM after graduating from North Carolina State University with a Bachelors of Science in Computer Science.

Special Thanks

Special thanks are in order for several individuals who contributed to this publication on matters of style, structure, and substance.

Don Bagwell – Sr. Certified Technical Sales Specialist, ATS
Meshach Baptiste – Technical Sales Specialist – Rational & System z Tools
Togara Dinga – System z Software IT Specialist
Paul Houde – System z Software IT Specialist, ATS
Scott Peters – Lotus® Technical Sales Manager
Rod Riley, PhD – Software IT Architect
Will Smythe – Product Line Manager – HATS
Erasmus Tapera – Software IT Architect

Executive Summary

Service-Oriented Architecture

In the ever changing industries of today coupled with the race to remain competitive, companies can no longer be burdened and constrained with the rigid IT structures that have been productive for them in the past. As businesses and their goals change so should the supporting services and infrastructure. Historically, shifts in strategy and goals have been slow, expensive and occasionally prevented because there was no affordable way to change tightly coupled applications, data, and infrastructure. Today this is unacceptable.

The solution to these problems is commonly known as a service-oriented architecture (SOA). SOA is an approach to designing IT architecture and business services based on the principles of flexibility and reuse under strict governance. A SOA is implemented by creating business services that are well defined and loosely coupled to work together to meet the company's goals. The static piecemeal architectures of the past were effective at accomplishing the goals they were designed for, but have failed to produce the same results when companies change their strategic direction.

The implementation of a SOA provides a number of benefits to the business including:

- Flexibility to reuse existing technology and services in new ways that are still compatible with a defined governance structure.
- The ability to add to or expand the current infrastructure and services within the bounds of the same governing.
- The freedom to create or change business processes for increased responsiveness to customer needs.

Each of these benefits contributes to raising the company's bottom line. More and more we see that a client's potential for success is greatly increased when business processes and IT services are aligned; a SOA helps facilitate this possibility by providing a dynamic infrastructure designed for change.

WebSphere Portal, Rational HATS & the SOA Strategy

Whether a business is evaluating the benefits of SOA as their IT strategy or has already started the implementation they should evaluate the benefits of combining WebSphere Portal and Rational HATS as either a SOA entry point or next step. This combination is both a safe starting point and next step in the implementation of a SOA and has proven to have quick results.

WebSphere Portal has frequently been a starting point for a SOA because it can utilize existing assets to be the aggregation point for repeatable business services. WebSphere Portal portlets are stand alone applications that execute a business function inside of the Portal framework. There are multiple portlets per page, each of which can execute independent of the others and can functionally be web services, applets, calendaring, or any other application the user chooses. The Portal framework is also

designed to be a collaborative tool that bridges the communication gap to make people and resources available regardless of location.

Rational HATS is an Eclipse based development environment plug-in that allows developers to modernize legacy applications by exposing them to users through the Web. HATS effectively transforms “Green Screen” applications to Web based applications with graphical user interfaces (GUIs). This is done by developing an HTML presentation layer that has direct host access. This HATS application is then deployed to the WebSphere Application Server framework for Web exposure and effectively removes the need for traditional 3270 emulators.

This functionality is useful in several ways, particularly when used in conjunction with WebSphere Portal technology. First, companies can continue to utilize essential mainframe applications (e.g. CICS[®] and IMS[®]) without having to train new staff on mainframe commands, navigation, and application functions which reduce training expenses and increases adjustment speed to business changes. Secondly, when HATS applications are deployed as portlets, they can be used to create composite applications and cooperative portlets for new business process and increased productivity. For further information on how Portal and HATS can help in a SOA strategy please examine the information in the following sections.

Introduction to Service Oriented Architecture

Defining SOA

Service-oriented architecture is an IT architecture designed around the idea of creating business *services*. These *services* are components of the larger operations that run the business and can extend all the way from the user to the application development level. IBM has defined the *services* in a SOA as being “well-defined and loosely coupled interfaces which promote interoperability and reuse with other *services*.” The primary function of these services is to provide some business capability to any user or service requiring only knowledge of a standardized implementation. It is important to understand that SOA is more than just UML diagrams and Web services. SOA is a method of design that, at its fullest implementation, includes:

- Run-time deployed applications similar to transactional system based on business process modeling
- Middleware implementations for service connectivity
- Clearly defined, well governed, repeatable business services that are supported by a modular, loosely coupled, and reusable IT infrastructure.

Beyond this, a SOA is a changing implementation of a unique paradigm. The SOA lifecycle shows us that as business needs and goals change so does the supporting architecture. Input from our infrastructure management feeds into new infrastructure modeling which in turn changes the services we need to assemble and deploy, all of which remain transparent to end users and inside of the governance policies established. This is all possible because of the defining characteristics of a service-oriented architecture. Below are brief descriptions of the fundamental characteristics of a service-oriented architecture.

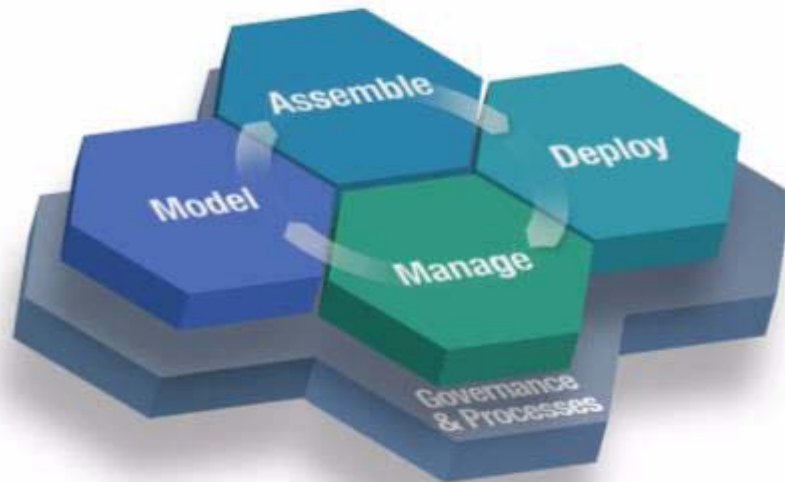


Figure 1

Flexibility

Flexibility, or loose coupling, means that two particular business services have the ability to interact with each other through standardized and commonly understood interface requirements. Loose coupling also attempts to hide service dependencies to maximize reusability and modularity. This flexibility allows for most service types to interact with a second service type in the same way it would interact with a third service type.

Reusability

Reusability is the construction of new composite business operations by choreographing current services within the bounds of the governing structure. Reuse is one of the most important sources of SOA business value because it reduces development and testing time. Reusability is tangentially tied with flexibility in a SOA because the construction of new business services is made possible by loosely coupled existing services. The key focus here is to avoid having to incur the costs and overhead associated with buying or developing new services and infrastructure.

Scalability

Scalability as it relates to SOA means that the infrastructure and services that are set up must be able to grow and shrink with the business. This characteristic draws heavily upon the previous two because in order to support more customers, for example, a user interface must be reusable and loosely coupled with other systems so that it may be deployed and customized on another server. The alternative is that a new interface will have to be developed for the influx in business. This would result in additional development, testing, and support cost.

Governing

A SOA must have a well defined governance structure in place or it will lose its ability to support business needs efficiently and provide unique value. The governing structure is what defines business processes as well as establishes guidelines which define what services are and how different services may be used, or reused, to perform new processes. As the business and its supporting IT grow, structure and organization are increasingly important in ensuring a SOA remains flexible, reusable and scalable.

Entry Points

Often clients see the benefits and are interested in a service-oriented architecture but have difficulty understanding where to begin. It appears to some that they need to restructure their entire IT department or replace existing business processes; this is when understanding SOA entry points becomes valuable. The entry points below are areas in which customer may choose to focus during a SOA implementation.

People

Throughout the course of conducting business, people need information. They collaborate with other people inside and outside the enterprise to make decisions. The information associated with business processes are delivered by linked and reusable services through SOA applications. To gain an edge in a competitive world, enterprises must find ways to improve people productivity and promote collaboration. This entry point focuses on helping people be more responsive by conducting business in real-time.

Process

Business processes define how a business works. IBM has defined a Process Entry Point which provides a starting point for deploying SOA enabled business process modeling solutions. This entry point focuses on specific tools and consulting services to help streamline and improve processes across the enterprise.

Connectivity

True connectivity is the ability to link people, processes and information in your business with a seamless flow of information from virtually anywhere at anytime. The Connectivity Entry Point can help focus your implementation to create a more secure, reliable and scaleable infrastructure to connect within and beyond your enterprise. Message flow and enterprise wide communication is at the heart of the Connectivity Entry Point.

Information

When architecting SOA-based solutions it is often important to consider how you will make information, which may be from many disparate sources, readily available throughout and beyond the enterprise. While the Connectivity Entry Point can help you provide access to this data, the Information Entry Point focuses on enabling you to obtain maximum value from the data by deploying information integration, master data management and analysis tools to ensure users have the data they need, when they need it, and in a form that they can quickly digest and use.

Reuse

The Reuse entry point focuses you implementation on maximizing existing resources by reusing applications and services. This increase business flexibility and responsiveness by reducing development time and helping eliminate of duplication of processes and systems.

WebSphere Portal on System z

What is Portal?

WebSphere Portal is an extensible framework for a consistent, integrated, and personalized point of access to different business resources. These resources can be in almost any form:

- Legacy Application
- Web Applications
- Static or Dynamic Content
- Applets
- Web Based Forms
- Documents
- Many more...

Foundational to the understand Portal as a product is an understanding of the concept of portlets. WebSphere Portal aggregates disparate resources into a Portal page(s) through independent applications called portlets. Each of these portlets are independently developed, can contain any supported content, and can be placed at any location on a given Portal page. After a portlet is deployed to the Portal page a theme is applied to each portlet and a skin is placed over the entire page to give a clean consistent presentation.

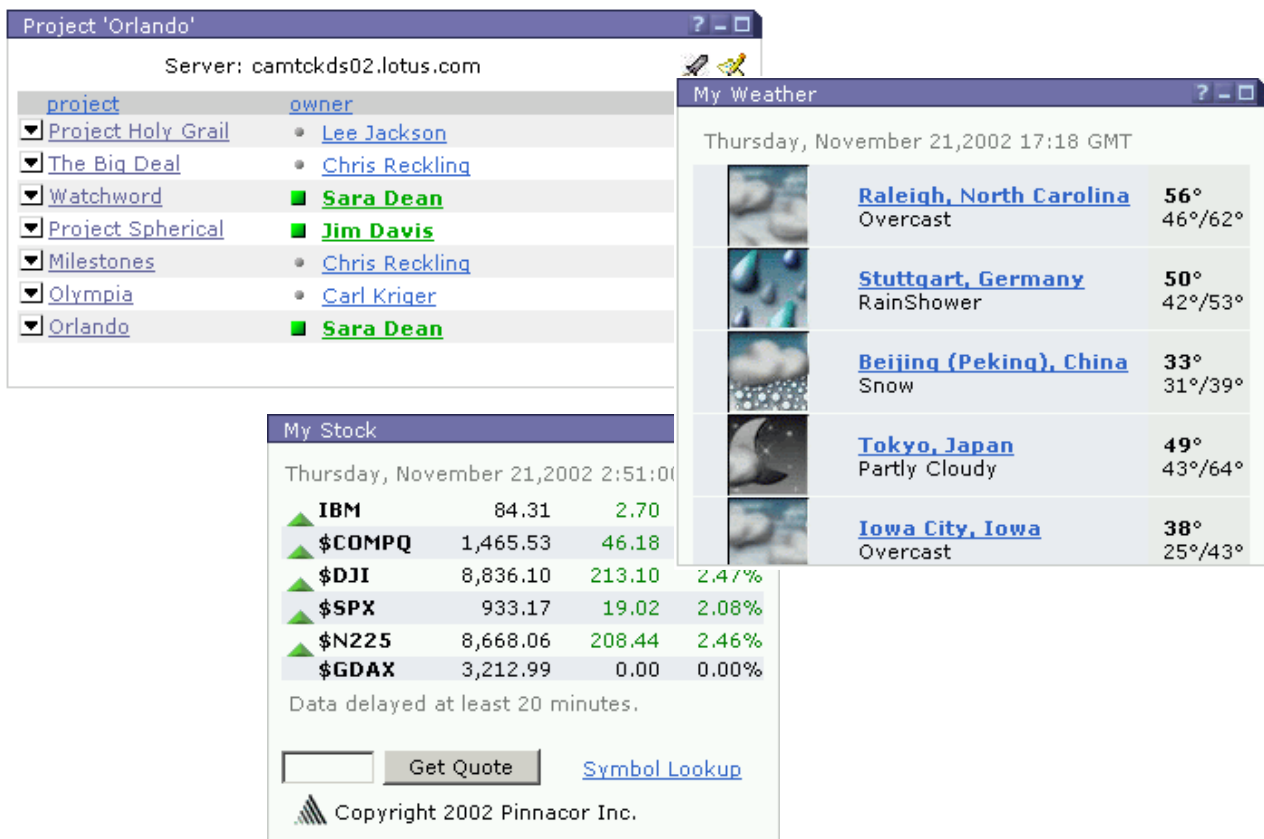


Figure 2

The portlet concept can be understood from several different perspectives. To the end user a portlet is simply a window which displays some content they need. To the developer a portlet is merely a Java application that gets managed by the container in which it resides. To the Portal server a portlet is just another component that is to be aggregated and rendered on one of its pages.

The basics of WebSphere Portal are easily understandable concepts to anyone with servlet or WebSphere Application Server (WAS) experience. The purpose of an application server, in very simplified terms, is to provide an environment for application code to execute. WAS as a product, provides, manages, and services multiple application servers that are capable of executing application code. Portal was designed to run on top of the WAS environment. More specifically WebSphere Portal is a specialized application server that runs inside of WebSphere Application Server.

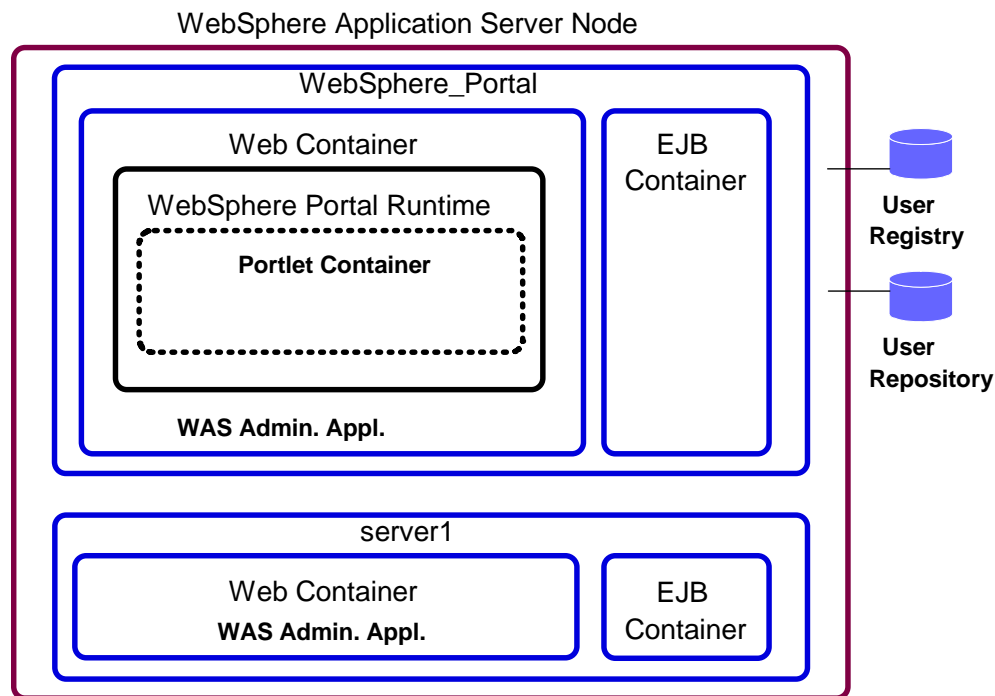


Figure 3

Within WAS, an application server defines the implementation of both a Web and EJB container. Figure 3 above shows that the WebSphere Portal server resides in a WAS application server node like any other application server. The Portal server however, is configured with several specialized *.ear* files and its web container also includes a portlet container. From figure 4, below, you can see a conceptual flow of how a portlet executes within the Portal runtime. Step 1, a client clicks on some portlet on the Portal page. That request is sent to an action dispatcher function which in turn sends input to the specific portlet inside the container (step 2). In step 3, the portal server is communicating with a back end application, basically like a servlet. This communication can be in the form of a SOAP or JDBC connection or even by sending a request to the EJB container displayed in figure 3. The results of the portlet execution are returned to the portlet container for markup then passed to a page aggregation sub-system which renders the portlet to the

page. This conceptual process executes for each independent portlet which means the entire page does not have to be refreshed for the results of one to be displayed. Fundamentally a portlet is a servlet that runs inside of a special framework that allows *seemingly concurrent* execution on a screen the user can customize.

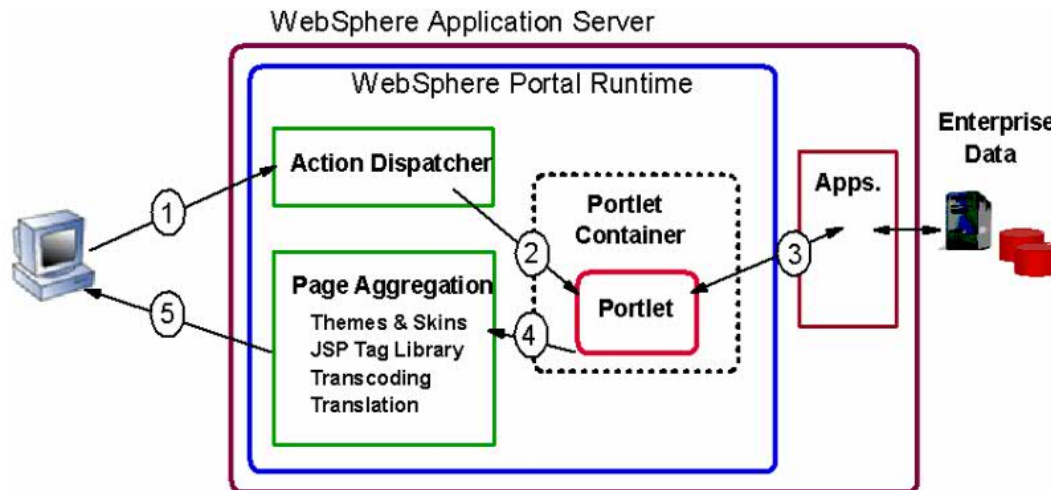


Figure 4

Portal and SOA

WebSphere Portal has the potential to play a major part in any SOA because unlike most other architectural components, Portal creates tangible benefits that the user can experience. Often times when discussing SOA the conversation is abstract and refers to previous or possible future financial and performance issues associated with a static infrastructure. WebSphere Portal addresses real issues at the user level where changes have the most immediate impact. One of the biggest benefits of WebSphere Portal is its ability to integrate other applications to facilitate collaboration among employees. Portal user can experience enhanced collaboration abilities in the form of e-mail, calendaring, people awareness, document management, desktop integration, and Workplace forms. This integration can span the breadth of a global a corporation to increase productivity by making people and valuable internal resources available in a single location.

This front end approach to a SOA is what IBM calls the People Entry Point. This has traditionally been a starting point for SOA implementations because it allows businesses to utilize existing infrastructure in new way thus realizing quick results while deriving continued value from existing assets. Portal is also a great starting point for a SOA because of its ability to simplify business processes through cooperative portlets and composite applications. This Portal function allows businesses to design portlets that span multiple applications or services thus reducing process complexity.

WebSphere Portal is more than just an amalgamation of Web services at the presentation level; it is a collaboration service and deployment platform. A major benefit of Portal is the Eclipse based plug-in, WebSphere Portlet Factory. Portlet Factory is specially designed for rapid application development in the Portal framework through IBM development tools such as Rational Application Developer (RAD) and Rational Developer for System z (RDz). This rapid development allows for quick responses at the application level to changes in architecture or business processes. IBM also has

developed the Portlet Repository which is generally available to Portal customers. This repository contains well over 600 portlets and composite applications that are both general in nature as well as industry specific. Having these resources available facilitates the reuse concept in service-orientated architecture.

Why Portal on System z?

Though there are many benefits of running WebSphere Portal on z, there are two that are most obvious: protection of your asset and cost of operation. For over 40 years IBM has been at the forefront of computing technology and the mainframe has been the foundation of this presence. A market presence as powerful as System z comes only through a history of providing a server that is as reliable, available, and scalable as possible. Portal on System z provides a valuable user experience that companies can depend on to meet their user's needs. Consider, would your business be noticeably affected by a disruption in your Portal service? Would revenue be lost? Would a decrease in employee productivity hamper day-to-day operations? If the answer to any of these were yes then you can appreciate the quality of service provided by running applications on System z.

The two operating systems customers can chose to run Portal on each have a unique value proposition. For z/OS customers there is unparalleled quality of service (QoS) through hardware redundancy, Portal ZFS support, DB2 performance enhancements, capacity on demand, and work load management. Linux for System z allows customers to passively exploit the hardware QoS inherent in System z while providing an open platform that is easy to manage and can optimize existing infrastructure through server consolidation and management.

System z offers many benefits that are focused on protecting the assets and investments you have on System z but there are also substantial financial savings associated with running applications on the mainframe. Today hardware costs are lower than ever but that is being met with higher labor costs; System z technology has the ability to substantially reduce both administrative and hardware costs associated with creating production, test, and development environments in a distributed computing scenario. Further, IBM offers specialty processor engines for System z which drastically reduce MIPS (millions of instructions per second) cost for executing Linux, Java, or DB2 work loads.

For customers who have adopted a Linux strategy IBM offers the IFL (Integrated Facility for Linux) processor which allows System z customers to execute Linux workloads on these processors free of z/OS related software charges. Also important to Portal customers is the IBM zAAP (System z Application Assist Processor). zAAPs, similarly to IFLs, run asynchronously to general purpose processors and allow z/OS customers to offload Java work loads. This dedicated Java execution engine reduces the general processors work load and therefore providing more execution time for other non-Java applications all while reducing mainframe costs.

Rational Host Access Transformation Service

What is HATS?

Host Access Transformation Service (HATS) is an Eclipse based plug-in that allows developers to transform legacy (3270 or 5250) application interfaces from traditional green screens to more user friendly HTML pages. HATS effectiveness comes through its ability to accurately recognize these host screen components and render them in HTML, in real time, to a browser. Through a set of predefined rules, HATS can also add drop down menus, links, tables, labels, buttons, value lists, tabs, and even graphics to the Web interface. It is important to state that the HATS application to host screen mapping is not one-to-one. HATS is not another screen-scraping technology. Screen-scraping merely provides a one-to-one HTML representation of a green screen while HATS technology is a rules based transformation engine that provides dynamic translation of any page. A single HATS application can provide programmed support for navigation through multiple host screens as well as the ability to skip and combine pages, pre-fill or verify entry field values, and retrieve global variables based on previous answers or user preferences. Screen-scraping cannot dynamically handle any changes.

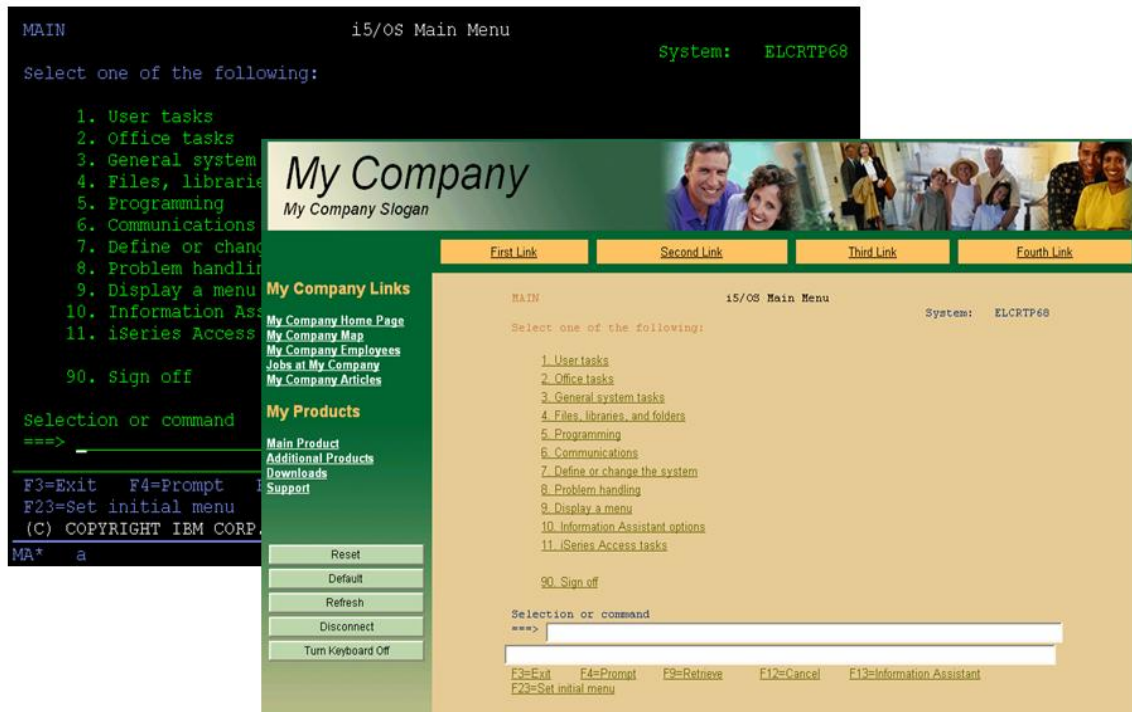


Figure 5

After a basic understanding of HATS capabilities the next question is: How does HATS work? HATS applications interpret the data stream bound for a user terminal, and dynamically change the 3270 data stream into HTML. During the HATS application development process, a Java EE application file is created so that the application can be deployed to the WebSphere Application Server. It is also possible for the user to create

a .war file so the application can be deployed as a WebSphere Portal portlet. Within either of these files is the HATS application runtime code, the host components that will be displayed to the end user's screen, as well as any enterprise Java beans that which can provide host access to other applications. The created .ear or .war files are then deployed to either WAS or Portal, respectively, and are ready for use. Once deployed a HATS application, just like portlets, act fundamentally like servlets on an application server. The deployed HATS runtime code and EJBs facilitate GUI based client-host communication thus reducing the requirement for traditional 3270 and 5250 emulator skills. Further, these HATS portlets are fully capable of being part of a composite application or cooperative portlet scenario that utilizes WebSphere Portal click-to-action (C2A) and single-sign-on (SSO) capabilities.

HATS and SOA

In today's mainframe, applications and processes built on traditional languages, such as COBOL and PL/1, and runtimes, such as CICS and IMS, continue to deliver a wealth of mission-critical benefits to enterprises of all sizes. Many of these existing applications run on the System z platform and are accessible through either 3270 terminals or terminal emulation programs. A foundational principle of SOA is reuse, these existing green screen applications don't lend themselves to that principle very easily. The new workforce has less and less mainframe experience while the businesses dependency on it remains constant. It is at that junction where HATS can play a critical role in facilitating the continued realization of value from existing technology.

The SOA Reuse entry point is more than just using existing resources in new locations, it is about using them in new ways. Client experience has demonstrated that users can see real results by using HATS as a green screen modernization tool. Application modernization is a worthy investment for both an SOA implementation and the company's bottom line. Below are a few of the results HATS can bring:

- Increases in efficiency through modernized application interfaces
- Increased productivity by aggregating legacy application screens
- Increased productivity by utilizing HATS functions (e.g. entry field pre-population and validation)
- Decreased training costs by application modernization to offset the lack of mainframe experience
- Continued value from existing IT assets

This product also has the ability to play a key role in a businesses SOA strategy because it facilitates rapid response to changing business needs and processes. Once the application is installed, new HATS applications can be developed in a matter of hours rather than day or weeks. This speed, and associated cost savings, is a product of asset reuse.

Why Portal and HATS Together?

Risk and ROI

An inherent characteristic of HATS applications is safety thus allowing for quick deployment. By safety, we mean, rapid deployment of new interfaces that provide customers the same level of performance from legacy applications themselves while increasing employees ability to use them. By reusing thoroughly tested applications rather than developing new ones, HATS is minimal invasive to the IT department and provides what could be the first steps to a service-orientated architecture. This is a good SOA starting point for any company, not just because of the infrastructure reuse, but also because of the speed of return on the investment. Once HATS is properly installed into the appropriate development environment, legacy applications can be transformed into HATS applications and deployed very quickly.

The deployment of HATS applications to the Portal framework also allows the users to utilize legacy applications but the return to the customer has potential to be much greater. In addition to application modernization users have a centralized location for applications, including HATS, which streamlines business processes and increase end user efficiency beyond that of simple modernization. This benefit is further realized when HATS applications are used in a cooperative portlet or composite application scenario.

Composite Applications and Cooperative Portlets

A major advantage of WebSphere Portal is its support for the creation of composite applications and cooperative portlets. A composite application is a portlet that has condensed multiple applications functionality into a single portlet. Cooperative portlets are specially designed to communicate with other portlets within the same portal environment with features like wiring. These features are particular helpful if a users goal is process efficiency or improvement. This functionality can also be exploited by HATS applications deployed as portlets. Within the Portal environment there exists a framework (i.e. composite applications and cooperative portlets) for intra-application communication. This means modernized legacy applications are no longer isolated to their previous roles but can interface with other portlets and applications such as business process modeling, collaboration tools or even reporting software for tremendous flexibility and efficiency.

Complementing SOA Entry Points

The purpose of deciding upon a SOA entry points is to provide a focal area for clients when a SOA implementation begins which helps create a disciplined approach to reorganizing current infrastructure. WebSphere Portal is a principle focus of the people entry point because it focuses on making people more effective. WebSphere Portal provides extensive resources to end users in the form of centralized applications, process simplifications, and collaboration tools. Rational HATS fits squarely in the reuse entry point because its primary goal is to help clients continue to realize the benefits of existing assets in new ways. Portal and HATS together make a productive combination because companies rarely pick a single entry point to guide their implementation. These two

products complement one another to increase speed of deployment which allows end users to realize benefits quicker while having very few implementation dependencies.

Improved Efficiency

Obviously an end user would expect improved efficiency with applications like Portal and HATS but often they are not sure what that looks like. Below are a few areas where users can expect to be more productive when starting their SOA implementation with WebSphere Portal and Rational HATS.

Green Screen Enhancements

A primary effect of using HATS is improved efficiency through the addition of graphical user components that can be added to host screens. Traditional mainframe data entry pages required moving the cursor to the exact location where data belong and each field had to be populated with data of the appropriate type and length. Through HATS macros and global variables, drop downs, check boxes, radio buttons, calendar pop ups, and graphs can all be added to a mainframe application screen. Further, these components can be pre-populated based on previous input values and globally defined variables. These additions make completing legacy application screens exponentially faster through easier navigation and reducing input field value errors.

Aside from making a single mainframe application screen easier to navigate, HATS also allows developers to combine, skip, and split screens for a more logical and efficient flow. These screens can be customized based on user needs and preferences to include additional buttons, links, valid value lists, as well as modified graphics, colors, fonts, and layouts.

Aggregation

Using WebSphere Portal as the deployment platform for applications proves to be extremely useful in cases where end users have previously had to deal with many disparate applications from multiple locations. WebSphere Portal can act as an aggregator of data, applications, and people therefore improving the entire business landscape by bringing these resources into a single location that can be customized for a specific user's experience. This capability exists for HATS applications as well. Rather than HATS applications being deployed as stand alone Web pages they can be deployed and integrated with other applications through the WebSphere Portal framework. HATS is an ideal complement to WebSphere Portal not just because multiple pages can be deployed to a single portal but also because each HATS application can be customized to reduce the amount of information in each portlet thus maintaining a generally accepted portlet presentation as seen in figure 6 below.

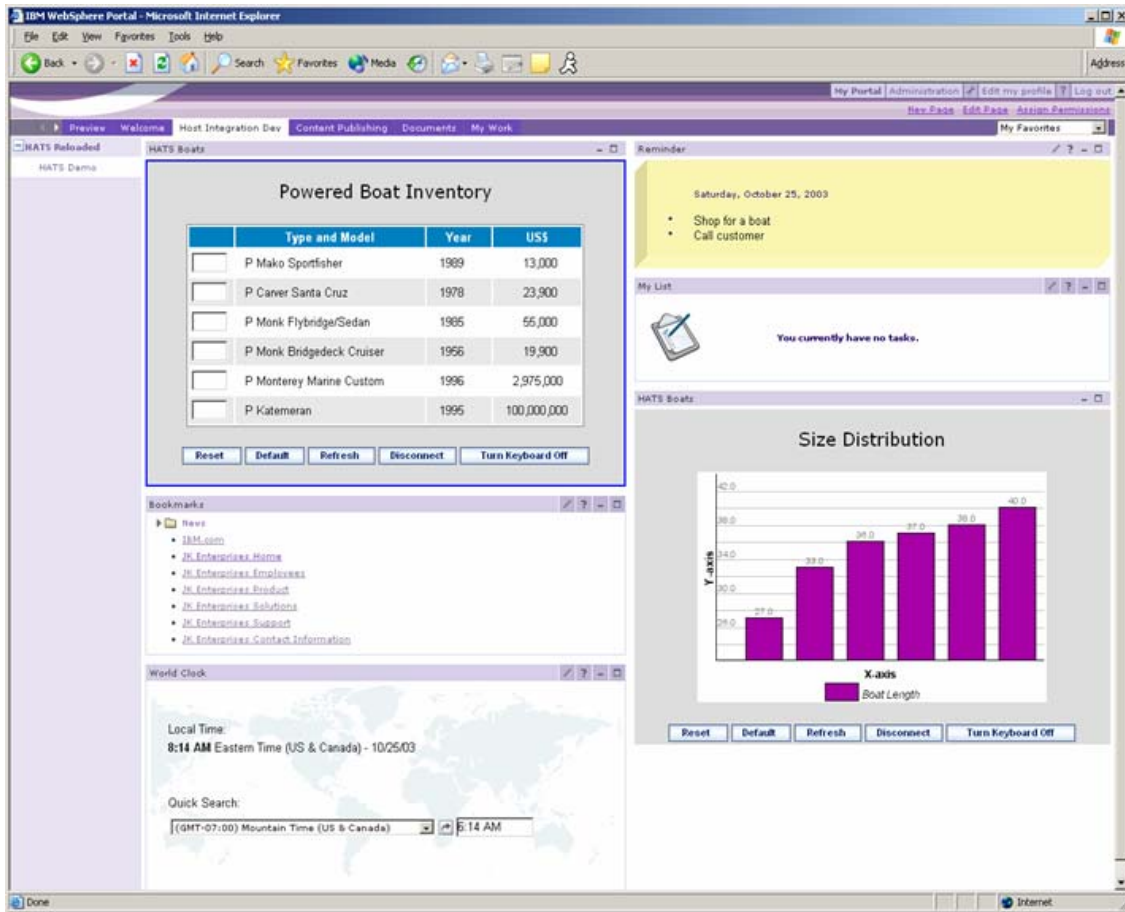


Figure 6

Workforce Training

An often overlooked and frequently underestimated cost associated with any enterprise application is user training. This is true when talking about mainframe applications because the younger workforce is less familiar with mainframe technology and navigation. Using Rational HATS and WebSphere Portal can help mitigate these financial burdens by enabling companies to present their mission critical mainframe applications in a way most employees are comfortable with and therefore reducing time spent on training.

Conclusion

The decision to adopt a SOA strategy to facilitate your business needs is one which should be closely followed by the establishment of long term goals and an SOA implementation strategy. Traditionally focusing on the People SOA entry point has been a productive first step in this process for several reasons. First, the people entry point addresses user level issues. Changes at this level have the most immediate impact for the business because people run business not processes or services. If you make people more productive, then the business's overall productivity increases. Secondly, a true SOA implementation will eventually extend to the user level. It is ideal that users will be able to exploit infrastructure improvement as soon as they are available; this is done by focusing initial efforts on people collaboration and productivity so that there is an interface that allows users to experience the full benefits of a SOA, on demand transparent service changes in response to business needs. WebSphere Portal meets this need.

An extension of WebSphere Portal's benefits is its unique integration support of Rational HATS applications. For many companies there is at least one component of their business that is reliant on extremely efficient mainframe applications that require tedious manual entry and a skill set that is waning in today's employees. Rational HATS takes the data stream from traditional 3270 and 5250 emulators that require these skills and dynamically converts that data stream to present the user with an interface that is both more efficient and easier to use.

By deploying Rational HATS applications to the WebSphere Portal framework, businesses which have adopted an SOA approach can, with little additional work, approach their SOA implementation from two concurrent entry points, People and Reuse. This SOA starting approach will allow business to experience increases productivity while continuing to derive value from existing assets and placing themselves in a position to exploit the future benefits a service-oriented architecture. No other software vendor can offer both a portal and application modernization solution with this level of integration.