Independent Assessment
by Alan Radding

IBM System z case study: Ball State University

Moving to Web 2.0 on the System z

PO. Box 590340, Newton, MA, 02459, 617-332-4369
www.independentassessment.com

Executive Summary

When word came down that the vice president of information technology was determined to make Ball State University (BSU), Muncie IN, into a leader in emerging media, the System z data center team knew just where to begin. They already had been working with SOA on the mainframe to solve a difficult address synchronization problem.

The mainframe looked to the data center team as an attractive platform for SOA and emerging media, such as social networking. The System z was there and running reliably. It readily supported Linux and Java, and it integrated with Windows. Little more needed to be purchased. And it was where the data and business logic already resided. The System z made perfect sense.

Having quickly mastered the basics of SOA on the System z by leveraging CICS, they turned their attention to what may possibly be the most popular application on campus, the student schedule. The goal was to deliver the student's schedule in real time, accessible over the Web using only a browser. And when Facebook released its public APIs, the System z team thought, why not?Ｗ ithin days, through the magic of reusable CICS services, they had a working proof of concept for BSU student schedules accessible via Facebook.

The System z team still has more to do and learn, particularly about governance and building a services repository. With the student schedule up and running on Facebook BSU is well on its way to becoming a leader in emerging media.

This case study shows how a forward thinking organization can leverage the power and flexibility of the System z to deliver cutting edge applications that provide immediate value.

Challenge—to Become a Leader in Emerging Media

When the Ball State University (BSU) data center team got the word that Philip Repp, Vice President of Information Technology, was determined to remake the 110-year-old, 20,000-student University into a leading force in emerging media they immediately thought System z. They already had begun developing services-based applications that fully leveraged the System z, CICS, and DB2. So, why not Web 2.0, social networking, and beyond?

Although the University had an extensive heterogeneous IT infrastructure, the System z was a natural to play the central role in meeting this latest challenge. The System z team had shown the way when it demonstrated the power of SOA in solving a tricky problem. Now, when the challenge was emerging media and Web 2.0, they already knew what their first target would be, possibly the most popular application in the cloud as far as BSU students are concerned—Facebook.

Led by Fred Nay, Director of University Computing Services; Brian Means, assistant Director of Computing Services; and Scott Linton, Senior Systems Programmer/Analyst , the team set out to enable students to connect with data residing on the System z through Facebook using only the student's browser. In doing so, they not only would be addressing the VP's challenge to turn BSU into a leader in emerging media, but they would be positioning the System z and BSU at the forefront of the most powerful trends sweeping the industry and the world at large—social media, cloud computing, SOA, Web 2.0, and mobile computing.. And they would do it by tapping the processing power, performance, reliability, and flexibility of the System z, which had been a proven workhorse at the University for decades.

Background—a Recognized Midwest Innovator

Located in Muncie, IN, a mid-sized Midwestern city one hour northeast of Indianapolis, BSU has been named by The Princeton Review as one of the best universities in the Midwest for four years running. It also ranked as the nation's top wireless campus in a 2005 survey conducted by Intel and published in *U.S. News & World Report*.

The residential campus is home to nearly 20,000 undergraduate and graduate students from across the country and abroad. It regularly wins awards for its achievements in green, conservation, environmental initiatives; for advances in gender equality; and for its Center for Information and Communications Sciences program

In terms of IT and systems infrastructure, BSU operates two System z9 BC machines running core administrative, student, and alumni applications. It recently upgraded to CICS 3.1, WebSphere Application Server 1.9, and DB2. It also added a zAAP to assist with Java workloads. It deployed an IFL and z/VM to run SUSE Linux on the System z [see box: BSU Systems and Software]

Like mainframes everywhere, the System z at BSU handles the school's core administration, student, staff, and alumni systems. A long-time mainframe shop, the System z was not acquired as an emerging media platform. However, the System z is capable of a wide range of tricks, especially in the hands of an innovative team. For example, BSU became an early adopter of SOA on the System z because services allowed the team to integrate systems more easily and effectively. The SOA skills it acquired along the way and the growing library of reusable services it is developing continues to pay off as the team forges connections to Facebook and other emerging media.

Welcome to Web 2.0

Web 2.0 lies at the heart of emerging media. The definition of Web 2.0 varies depending on the organization. A few things all agree on:

- Browser accessible
- Highly graphical
- Standards-based
- Able to handle any kind of applications running on almost any platform
- Focused on content in almost any form—data, documents, files, Wikis, blogs, spreadsheets, dynamic composite applications, mashups, live chat, instant messaging, and more.

For an organization like BSU that strives to be a leader in emerging media Web 2.0 is critical.

Web 2.0 makes extensive use of standards-based Web services. Web services produce location transparency; the end user doesn't know where the resource resides. The service might run on the System z or on a distributed system or multiple systems. Composite applications, created by calling multiple services, might be spread across different platforms, all of which are transparent to the user, who should never know nor care where the service resides as long as service expectations are met. The BSU student accessing her schedule through her Facebook account via a browser doesn't know that the application actually resides on the System z, is accessed through a CICS transaction, and the data resides in DB2.

The early adopters on Web 2.0 are the young people who have turned social networking websites like Facebook into a powerful global media phenomenon. They are the very students and faculty BSU hopes to attract and retain and who form the primary audience for Web 2.0 and emerging media that represent the University's future.

The ability to leverage its existing IT infrastructure, especially the System z, gives BSU a distinct advantage. Unlike organizations that must invest time and money in new systems to get to Web 2.0, the System z already is there and ready, little additional investment required. The data already sits on the System z and doesn't have to be migrated or touched. And the basic business logic and security is solidly in place. Little new has to be invented.

Overcoming Web.2.0 Challenges on the System z

This is not to say, however, that implementing Web 2.0/emerging media strategy on the System z is without challenges. In its pursuit of leadership in emerging media, the BSU System z team had to find ways to overcome various challenges, which, in the end, have proven to be quite manageable. The challenges include:

- Adopting services—Web 2.0 implies a services-based approach. For BSU this primarily means providing CICS data as a set of feeds compliant with Web services standards, mainly XML and HTTP. Tools like IBM's Host Access Transformation Services (HATS) can expedite the process of creating web services from existing mainframe code. BSU takes advantage of HATS for prototyping CICS applications as web services.
- Fast data access—data resides deep on the System z. Getting to it typically would entail creating and running stored procedures and spinning out the results to a distributed database server. At BSU, however, the DB2 database can handle XML natively. There is no need for an interim data server to act as a proxy.
- Accessing data via a GUI—this proved cumbersome in the past as mainframe shops resorted to screen scraping or Interactive System Productivity Facility (ISPF) to set up SQL queries in a screen panel or window. At BSU, the end users all have browsers, which provide the common GUI.

Other Web 2.0 challenges, such as mashups, can be handled through a portal like the WebSphere and various mashup tools.

Implementation

BSU, a long-time mainframe shop, began what turned into its pursuit of Web 2.0 and emerging media several years ago when it tried to solve what had become a vexing problem with student addresses. Many students, reflecting the changing nature of the modern family, had a multitude of addresses. Many had divorced parents, often there were step parents involved, and each entailed a different address.

The addresses were captured and stored in as many as 42 systems. It didn't take much before the data center team was scrambling to update and synchronize as many as a dozen different addresses associated with a given student in numerous systems.

The two big address databases resided on the mainframes, which ran the big alumni and student systems. The System z team decided to explore SOA as a mechanism to keep the addresses synchronized between systems.

The team looked at both Microsoft and IBM approaches to SOA, figuring they could do it either way at that point. Quickly, BSU decided IBM was two years functionally ahead of Microsoft as far as SOA and messaging buses were concerned, and it opted for the IBM approach.

BSU Systems and Software

System z9 BC
IBM DS8100 storage
IFL
zAAP
z/OS
z/VM
zLinux, SUSE Enterprise
Java
C$^{\#}$ .Net
HATS
CICS
DB2
DB2 Connect
WebSphere
WebSphere Enterprise Service Bus
Rational
Content Manager
WebSphere Repository & Registry

The SOA solution on the System z proved straightforward: BSU built a web service front end for users to enter a change of address. The change was passed in real time to CICS, which naturally communicates with web services. From there the web service would ripple the change through every system containing addresses for that user.

Some systems kept data in DB2, others in VSAM. It didn't matter to the CICS service. Recently, the team reported it had 99+ percent of the addresses correctly synched.

Other key components of the SOA initiative were:

- IBM Rational and WebSphere Enterprise Service Bus (now called WebSphere Process Server)
- zAAP processor to handle increased Java workloads
- HATS, which was used in conjunction with Rational to reach into the alumni system for addresses there

For the initial effort, which took several months, the team enlisted some IBM assistance. The alumni system, with its need for HATS, took a couple of weeks of additional work. Now the team reports it can knock out web services in a few hours. The initial effort engaged a team of about 20 mainframe and .Net developers.

Since the initial SOA success with the address system, BSU has been expanding and fine tuning its System z-based web services efforts. It discovered it could handle CICS in real time and boost performance by directing the web services to the COM area, which delivered sub-second response.

BSU also brought in zLinux (SUSE Enterprise), an IFL, DB2 Connect, and IBM Content Manager to expand its web services capabilities. For example, the team converted 3270 registration screens into web services. This enables continuing education and distant learning students to register for courses from anywhere, anytime via the Internet. They log in through their browser, select their courses, and know in less than three seconds if they got into the course of their choice.

At the time of this writing, BSU had developed and deployed about 20 reusable web services for the System z. Other BSU development groups on other platforms regularly contact the System z team about reusing those services in their applications.

Student Schedules and Facebook

One of the biggest and most used System z applications at BSU is the student class schedule. The students' lives revolve around their class schedules. When classes are moved, changed, cancelled, or rescheduled students want to know fast. As soon as the System z team had proven it could deliver browser access to mainframe data with the address application, it turned their attention to the student schedule application.

In the past BSU provided access to the schedule through a stored procedure that would deliver the data to a SQL Server database running on a distributed platform. This certainly worked to a limited extent but the results were a day old and performance was slow. Day old notice of a class cancellation was problematic at best. The team wanted to deliver real time results.

It turned to CICS to extract DB2 data in real time and generate it as a web service any student could access through a desktop browser while keeping in force all the expected security and access controls. The application proved to be an immediate hit. When the team discovered the use of COM for the transaction it could deliver the results fast, in 0.2 sec. The web service goes directly to CICS through the COM area.

The web service developers skipped Java and used C# to create Microsoft ASP .Net service that calls the CICS schedule service. C#, the developers felt in this case, was easier to code than Java and could deliver the same results.

With the advent of Facebook, however, simple browser access to the schedule wasn’t good enough. Students increasingly focus their lives around Facebook. The BSU team quickly realized that it needed to bring the System z schedule application to Facebook, which was used by an estimated 80% of the student body.

At about this time, Facebook started to publicly expose APIs. The BSU System z team spent about three days working with the APIs to come up with a proof of concept. It realized it could simply reuse its existing CICS schedule application almost as-is for Facebook. Facebook access to the schedule proved an immediate hit.

The Facebook success opened up the BSU team to an entirely new area of thinking about the System z. What else, the System z team asked, could they offer now that they are able to grab mainframe applications and data and make them available through Facebook.

Results

Success with SOA and Web 2.0 has empowered the System z team. “With web services, we feel we can integrate any data and systems,” declared Fred Nay, Director of University Computing Services. BSU’s growing skills with reusable Web services, CICS, DB2, WebSphere, and other SOA tools enables it to deliver new capabilities quickly and easily. Specifically, it has:

- Delivered virtually complete address synchronization for students and alumni systems
- Established the basis for a new master data model fashioned after the address synchronization effort
- Enabled students to incorporate the mainframe scheduling application into the very fabric of their lives through Facebook
- Established a growing library of reusable web services
- Facilitated collaboration between BSU’s mainframe and Windows platforms
- Expedited the solution of complex business issues through SOA

Most importantly, it has enabled BSU to effectively pursue leadership in emerging media by leveraging its investment in the System z

Future Directions

Certainly, BSU’s work on SOA and Web 2.0 is not finished. Already the team is looking ahead:

- The team has begun addressing governance issues, first within the IT organization and later to be extended to the business units. It aims to tie technology capabilities like web services to specific business needs.
- It has acquired WebSphere Service Registry and Repository and will use it to build an accessible library of reusable services.
- It is exploring different ways to expose DB2 data directly, such as through the use of DB2 Connect.

The school also is just starting to become involved in IBM’s System z Academic Initiative, which happens to have a large Facebook presence.

Lessons Learned

Looking back, the System z team was glad it started small with the address application. It was a complex issue but was tightly bounded. Until they had it right the effort could keep it within the System z group.

On the other hand, the team under-estimated the learning curve for WebSphere and Rational. It has succeeded with both but the learning curve involved should not be discounted.

The System z team also wished it had uncovered the use of COM with CICS sooner. The performance gains it enables make the use of web services even more attractive.

Independent Assessment analysis

Although it is a long-time mainframe shop BSU represents the new face of the System z. Its use of the System z for SOA and Web 2.0, and the ability to incorporate social networking websites points it straight into the future. If Philip Repp, BSU VP/IT, is determined to make the University a leader in emerging media, he has the horse to get there in the System z.

The System z can play in the cloud. It can handle mobile devices. It can out-virtualize any rival platform, and with Linux and OpenSolaris it can be open source too. And it does so with the reliability, scalability, manageability, and security that makes it the mainstay of enterprise computing. So Repp is on the right track; BSU indeed can ride the System z to leadership in emerging media, which is where it wants to go now. And when it gets there and sets its sights on the next goal, whatever that may be, don't be surprised that the System z will take it there as well.

What the BSU case shows is how versatile the System z has become. With assist processors and a rich software eco-system from IBM and its partners the BSU System z team pushed out the borders of mainframe data center computing: automatic address synchronization across 42 systems, online anywhere/anytime registration, real time schedule changes via Facebook. The only question left is the one the team itself asked when it nailed down the Facebook application: now that we have this capability what else do we want to do with it. As soon as the business side at BSU catches on to what's now possible, the team will find it is very busy.

About Independent Assessment

Independent Assessment (http://www.independentassessment.com) is the IT and business assessment, analysis, and writing service of Alan Radding, an independent business and IT analyst/writer for over 20 years. It provides independent ROI and TCO analysis, competitive assessment and positioning reports, case studies, and Web content.

Independent Assessment publishes *dancingdinosaur*, the independent blog covering the System z, **http://dancingdinosaur.wordpress.com**