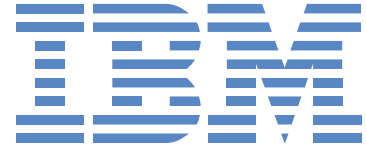


Customer Case Study: IBM Global Services

After working with many open source technologies for a period of time, writing lots of JavaScript code, and a development process which was taking much longer than they could afford, IBM Global Services (IGS) turned to Backbase to deliver their Rich Internet Application (RIA) on time. Backbase brought scalability, code maintenance, smart prototyping and best practices.



About IBM Global Services

IGS is the world's largest business and technology services provider. It is the fastest growing part of IBM, with over 190,000 professionals serving customers in more than 160 countries. IBM Global Services started in the spring of 1991, with the aim towards helping companies manage their IT operations and resources.

The Challenge

The "Fire Program Analysis" project from the National Interagency Fire Center (NIFC) involved working with several government agencies. It involved the integration of several scientific models, like fire behavior and containment and a lot of backend database integration as well. The application is used for strategic planning and budgeting for civilians, land management agencies and it deals with fire fighting preparedness, fuel treatments, and prevention activities. The models are designed to show the impact based on specific funding decisions. The impact would mean: number of fires contained, number of acres, etc.

The Rich Internet Application includes lots of users, with about 139 "fire planning units", distributed nationwide, all over the country. Users range in skill from "not really computer savvy," more operational people like firefighters, to pretty astute, tech-savvy users. So for the user interface in particular, IGS was looking for something that would give users a fairly intuitive view into their model input and the execution of the model and their results.

The Solution

"We planned on using a Rich Web application", explained Steve Carty, Infrastructure Architect. "For various reasons, small client footprint was a requirement. Being an interagency application you can't be assured that you'd even have administrative access to install an applet or even get permission required to run on a client system. Another reason we went with a Web application was because the teams themselves are distributed geographically, and even within a team there could be vast geographic areas that the different agencies collaborating have to work across."

“ With Backbase the amount of JavaScript coding is greatly reduced.”

Sudhir Rao
System Operation Architect

“In the initial phase, there was a form for every entity in the database – pretty much a 1 to 1 mapping – and the client wanted more interactive tables for using multiple records in a single application or single trip to the server. The client was looking for a richer client interface and Web 2.0 capability”.

IBM investigated a number of open source rich client JSF solutions. “JSF is appealing because it implemented most of our UI architecture on the client and the server making server integration easy for Java developers”, said Stuart Bedoll, IT Architect. “However, JSF did not address the requirement of evolving a working UI model with the customer. We needed a light-weight UI technology that could be easily consumed by the development team for subsequent integration.” JSF requires a server for rendering. IBM started using ExtJS and DWR to provide a mechanism that could support this process but this required lots of JavaScript and resulted in a heavier process overall. Stuart continues: “While we originally considered the JSF Edition, it was the Backbase Client Edition that enabled us to change the way we work with our customer”.

“We selected Backbase because of its technology and its support and development services”, said Sudhir Rao, System Operation Architect. “We had a fairly small team, for what we were trying to accomplish, with limited user interface skills, so the services offering was a big part of the decision. Backbase also had the documentation, an established and coherent framework using the XML and the XPath approach for the widgets and that is how we got to select Backbase.”

“ The reuse of prototype code for the final application saved us lots of time. Time saving could be as much as three to five days for a two-week iteration ”

Stuart Bedoll
IT Architect

The Result

“Our first step with using Backbase was to use it with local data sources and use it as a mock-up for the customer to review. It gave us an almost seamless migration path from this home-cooked JavaScript mock-up to a mock-up that we could hand over to the developers and be productive with, without having wasted the effort in developing the mock-up. So that was the initial step which I thought worked out really well.”

“We chose to use the client edition” explained Stuart. “The most appealing thing, and the main reason we were successful was because we could leverage the prototype. Our UI designer took care of the JavaScript, implementing basic functionality directly with the client because we could layout a page with reference to xml data and the customer could see pretty much what they were going to get – a true WYSIWYG experience. We separated our development in two steps: UI development and backend development which included integration.”

| Program Name | Description | Total Hours | Total Partner Cost | FTEs |
|---------------|--------------------------|-------------|--------------------|------|
| Lucent | Original Current Program | 0 | 0 | 0.00 |
| Copy-Current | Original Current Program | 0 | 0 | 0.00 |
| Need 4-2 Eopp | | 190 | 0 | 1.24 |
| Need 4-3 Test | | 100 | 0 | |

| UnitID | Role | Total Cost | FTEs |
|----------------|-------------|------------|------|
| AKMEA | partner | | |
| AKAKA | participant | | |
| AKSWD | partner | 1 | |
| AKGSD | participant | | |
| AKARD | partner | | |
| AKACD | partner | | |
| Totals | | 1 | 0.00 |
| Partner Totals | | 0 | 0.00 |

Time to market, as well as code integration and code maintainability were some of the other benefits. By separating the concerns associated with backend and front end

development, the development teams were able to avoid bottlenecks associated with one team being dependent on another and were able to work in parallel. This reduced the risk of delays and enabled innovation with the UI.

“Another selling point for Backbase was the availability of the consulting services”, added Steve. “With their help, we have delivered on time for our customer - which is saying something. In that respect, it is obvious that you can be very fast and productive when you master the technology”.

“I like the benefit of JavaScript integration”, said Stuart. “We wanted to reuse some of the source code already created with the open source Ajax libraries. We were able to reuse those, keep the same screens, and design the rest of the application to maintain the same look and feel - and the process was pretty seamless.”

“One big benefit of the Backbase framework is its architecture which facilitates code maintenance”, Said Sudhir. We started with lots of JavaScript code before using Backbase, but now the framework will make it easier to maintain the UI and will improve the process to modify and maintain the UI in the upcoming years. “Backbase’s strength lies in its XML support. With Backbase the amount of JavaScript coding is greatly reduced. When complexity dictates JavaScript, it can be abstracted and exposed via Backbase’s powerful XML templating framework which reduces the number of expert JavaScript developers one needs to build and maintain an RIA project.”

“Backbase technology allowed us to work directly with the customer on prototyping without server-side constraints”, Said Stuart. “It took a number of iterations to evolve our Client/Server design pattern – based on Backbase Client components, XML, and REST services. After that we were able to increase the speed of our UI development and integration, which resulted in shorter cycles and more agility. The reuse of prototype code for the final application saved us lots of time. Time saving could be as much as three to five days for a two-week iteration”. And 30 to 50% less time per iteration meant more iterations, more innovation, more involvement for the business and less risks of missed requirements or performance issues.

“What I like about the technology” said Sudhir, “is the ability to use the Client technology without having to build the backend. This allows us to have a more agile development process.”

“ **The Backbase client technology allows us to have a more agile development process** ”

Sudhir Rao,
System Operation Architect