CUSTOMER NEEDS AND STRATEGIES

OpenSpan's Approach to Mashing Up Applications Delivers Rapid Results

Maureen Fleming Jeff Silverstein

IDC OPINION

OpenSpan is a rapidly growing, privately held integration software company founded in 2004 that used its earlier expertise in customer contact centers to develop an unusual approach to developing composite applications and process automation. This approach is delivering impressive results. Highlights of our analysis are as follows:

- The OpenSpan Platform works by leveraging the communications between an application and the Windows operating system. Using the software, any object from any application can be exposed, normalized, accessed, and integrated with any other object. This allows developers to merge page flows from multiple applications into a single-purpose page flow for a group of users. In addition, the software allows developers to create a new user interface (UI) built on the underlying application objects.

- The more traditional and common way to integrate applications is to create a new user interface and then integrate the data from existing applications with the new application. OpenSpan fits into the emerging approach of creating a new application by mashing up components of other applications.

- OpenSpan's unique approach to integration makes it a vendor that should be considered for application integration and process automation projects that involve the need to improve productivity and quality for groups of users who work on repetitive, process-oriented tasks requiring the frequent use of multiple applications. Beyond contact centers, this product can be used to address a variety of areas, including sales and structured decision making, lending, underwriting, and claims evaluations.

Global Headquarters: 5 Speen Street Framingham, MA 01701 USA P.508.872.8200 F.508.935.4015 www.idc.com

Filing Information: November 2007, IDC #209547, Volume: 1

IN THIS STUDY

This document is part of a series of studies that examine the choices customers are making in integrating their applications. In this study, we discuss two customers' experiences with the OpenSpan Platform.

The goal of this study is to provide insight to help enterprises as they evaluate integration software. We wanted to learn more about why the OpenSpan Platform was selected for specific integration projects, the strengths and challenges of the platform, and the results customers are achieving once the platform is in production.

SITUATION OVERVIEW

The OpenSpan Platform is software that solves application integration and process automation problems using a far different approach than the myriad other solutions available. This platform integrates at the front end of an application rather than using the more common, data-centric back-end approach.

OpenSpan works by leveraging the communications between an application and the Windows operating system. Using the software, any object from any application can be exposed, normalized, accessed, and integrated with any other object. This allows developers to merge page flows from multiple applications into a single-purpose page flow for a group of users. The software also allows developers to leverage existing UIs or create new ones built on the underlying application objects.

The more traditional way to integrate applications is to create a new user interface and then integrate the data from existing applications into the new application. Instead, OpenSpan uses the emerging approach of creating a new application by mashing up the components of other applications.

During our interviews, we found that OpenSpan was selected for a variety of reasons and was put to use to support a variety of integration and process automation initiatives, including:

- Reducing call lengths and thus increasing call volume (Afni)
- Managing workflow by creating composite applications (Afni and Shepherd)
- Eliminating duplicate data-entry processes (Shepherd)
- Unifying internal systems without disrupting existing applications to work more efficiently with external systems (Shepherd)
- Integrating internal and external systems without modifying the external systems to improve workflow and call quality (Afni)

In customer interviews, we found common reasons behind the decision to purchase the OpenSpan Platform, including:

- Ease of deployment and testing
- Ease of training due to the similarity between OpenSpan's development environment and Microsoft's Visual Studio
- Lack of quick, nondisruptive integration alternatives
- Ease with which processes can be optimized

We interviewed OpenSpan customers to determine what triggered their search for a solution, the vendor selection process, how they identified their initial project, and what benefits they achieved by implementing the OpenSpan Platform.

OpenSpan Background

OpenSpan is a privately held integration software company founded in 2004. Based in Alpharetta, Georgia, the company had 50 employees as of October 2007 and grew more than 200% in 2006. Originally a professional services company focused on contact center integration solutions, OpenSpan productized its expertise and technology into the OpenSpan software platform, which was launched in 2005.

Contact center environments are good proving grounds for integration software because they offer a variety of parallel challenges: high volumes of incoming, simultaneous calls;, routing issues based on volumes that vary dramatically depending on time of day, day of week, or advertising campaigns; optimization of call results based on the proficiency of specific agents; the need for varying degrees of customization for each call; the need for multiple applications to service a call; and the high turnover of contact center agents.

OpenSpan's role in this environment is to help contact center agents respond to incoming calls more quickly, with better quality and less training. OpenSpan achieves this goal by simultaneously extracting information from multiple sources and applications and presenting it in a unified user interface. Thus, use of the OpenSpan Platform eliminates the training that would be required for agents to use the multiple source applications. This is a key consideration for managers in high-turnover contact centers.

OpenSpan also reduces the time the agents would have spent invoking, navigating, and extracting information from the underlying information sources and applications. These efficiencies can be quickly quantified and shown through the contact center's contact management system, allowing OpenSpan to document its value very quickly.

While its initial executive team was made up of call center experts, OpenSpan has broadened its mission and the expertise of its executives. When the company made the shift to become a product company, an experienced middleware entrepreneur, Francis Carden, came onboard as the new CEO.

In November 2006, OpenSpan raised $8 million from an investment group led by Sigma Partners and Matrix Partners.

In April 2007, four key executives came to OpenSpan from JBoss, the open source middleware company that was acquired by Red Hat in June 2006. OpenSpan is focused on selling its platform to companies in the telecom, financial services, and government markets and plans to begin targeting the healthcare industry shortly.

Key customers include Afni, Alltel Wireless, AON Integramark, Aspect Software, Atento Mexico, CSAA, First Citizens Bank, JCPenney, Regency Hospital Co., Shepherd Chartered Surveyors, and Westbrook Technologies.

OpenSpan's partners include Accenture, Aspect Software, Attachmate, HCTSi, Microsoft Customer Care Framework, Renewtek, Vertical Thought, and Westbrook Technologies.

Products

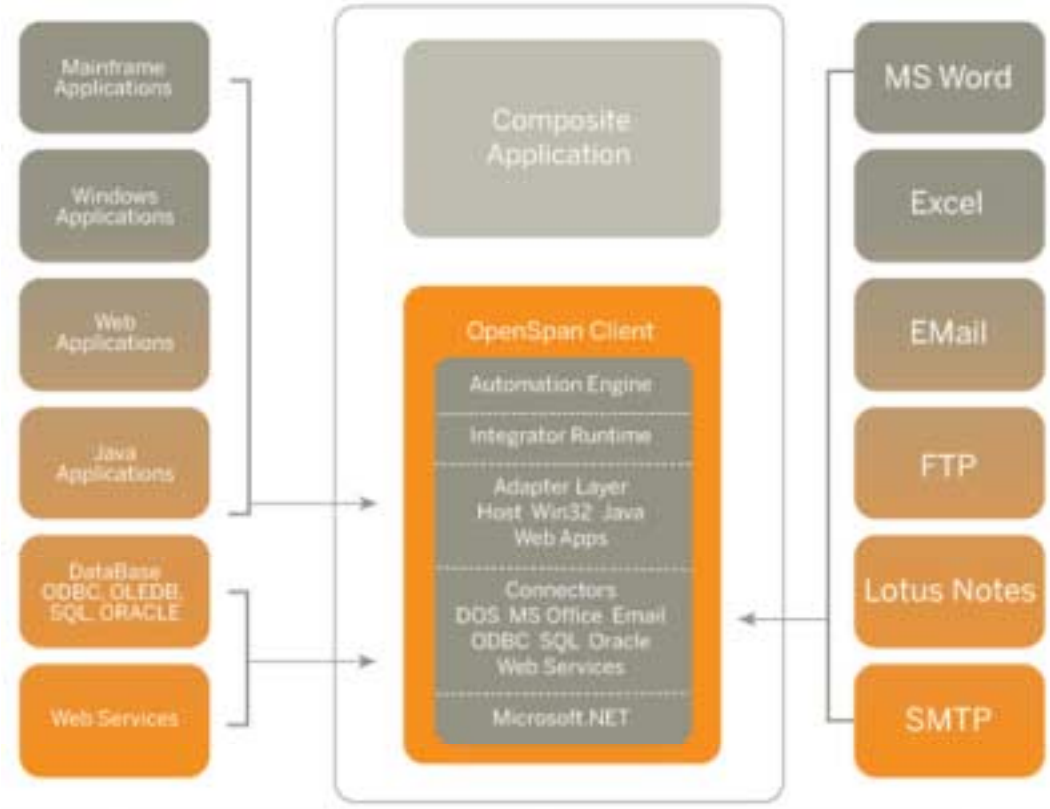
OpenSpan Platform 3.1, released in August 2007, leverages the communications between an application and the Windows operating system to integrate that application with others. Developers are able to do this integration without writing code or using application programming interfaces (APIs). With this software, any object from any application can be exposed, normalized, accessed, and integrated with any other object.

OpenSpan Platform 3.1 allows developers to merge page flows from multiple applications into a single-purpose page flow for a group of users. The software also allows developers to leverage existing UIs or create a new one built on the underlying application objects.

Going back to the contact center example, agents typically have to shift among several applications, accessing specific pieces of the individual applications to look up information and update records. With OpenSpan, the agents' workflow is created and then populated by the appropriate portions of the different applications. Figure 1 illustrates that OpenSpan is able to integrate thin client and desktop applications as well as Web services.

FIGURE 1

Diagram of OpenSpan Platform



Source: OpenSpan, 2007

There are three components to the OpenSpan Platform:

- **OpenSpan Studio.** Using OpenSpan Studio, a developer or business analyst can expose application objects, normalize them to a common object model, and assign them identifiers that allow them to be stored and reused as if they were packaged components. The construction of a composite application — in this case, a new application that either extends the functionality of an existing application or combines the functionality of multiple applications — Is built by dragging the normalized components that were exposed through the interrogation process to the appropriate target.

- **OpenSpan Integrator.** OpenSpan Integrator is the runtime environment for the OpenSpan platform. It is a thin desktop client that monitors and interacts with an application's internal code and data, including its user interface controls. OpenSpan Integrator's event-driven architecture allows developers to invoke code extensions to applications in runtime and orchestrate processes.

- **OpenSpan SOA Module.** This module interrogates and normalizes Web services in the OpenSpan Studio, allowing those services to be converted into reusable objects. Integrations and automations developed in OpenSpan Studio can also be integrated into a service oriented architecture (SOA) environment.

Afni

Afni, a privately held company based in Bloomington, Illinois, is a provider of customer interaction, call center, and credit and collections services for North American communications and insurance companies. The company has more than 4,600 employees.

We interviewed Afni IT Director Jeff Badger for this profile.

Call Center Outsourcer Saw Path to Improve Client Apps

As a call center outsourcing company with a client base that includes Fortune 100 companies, Afni is generally paid on a per-call basis. Thus, the company had a huge incentive to reduce the average time spent by its agents on each call.

But Afni had no control over the back-end systems of its clients, which Afni's agents had to work with as they serviced the clients' customers. The user interfaces and workflows of many of those external systems were complex and frequently involved duplication of data entry and clumsy navigation, wasting the time of Afni's agents and reducing the quality of the data input and the quality of the calls.

Application and data integration was a solution that couldn't be implemented in a traditional manner because Afni's contracts were too brief to allow for the development and implementation of major integration projects.

Along with an initiative to simplify the jobs of its call center agents, Afni wanted to streamline the processes of its clients and began searching for a noninvasive integration tool or platform.

OpenSpan Versus Jacada

Afni's IT environment was a mix of .NET and Java, along with WebSphere, with WebSphere being the dominant platform for internal applications.

The solution Afni was looking for, however, was an integration tool for external applications. It quickly winnowed its selection process down to a contest between OpenSpan and Jacada Ltd. (based in Atlanta, Georgia). In April 2005, Afni began a side-by-side proof of concept and had good experiences with both vendors. However, the Afni team did not like the way the Jacada components worked as a whole. Badger said that because Jacada's capabilities came through acquisitions, it was like having three separate tools that were strung together.

On the other hand, OpenSpan didn't require direct programming. It was a visual modeling environment usable by junior developers rather than requiring senior programmers. The Afni team believed it could complete its projects faster at a lower

cost with OpenSpan than with Jacada, which had a code-based development environment that would have required senior developers.

In addition, because OpenSpan was just getting off the ground, the Afni team felt it would have strong leverage on OpenSpan's technology development.

By October 2005, the purchase decision and licensing had been completed. Application development and deployment began the following month.

Rapid Training

Afni's development group consisted of about 10 people. The group included business analysts, who were focused on business requirements and process documentation, and developers, who were focused on code-based, programmed solutions. Although the business analysts don't write code, they were able to quickly learn OpenSpan's visual development environment. Two or three days of training was all they needed to begin.

Currently, Afni assigns a senior programmer to do training, introducing the system to new developers and giving the trainees a series of problems to do during a week.

Production Testing Experiment Yields Quick ROI Insight

Looking for insight into what changes to expect in call length and call quality, Afni did a six-week production prototype test involving a 15-agent control group that used client systems and 15 agents using OpenSpan integration and automation for the same client.

Initially, there wasn't a significant change because the savvy, high-performing agents, who had figured out shortcuts in the standard, inefficient environments, resisted the changes in the new OpenSpan environment. Some of the top performers actually saw their average call processing times increase. But Afni found that the lower performers made major gains in their call handling, with an average call-time reduction of 20 seconds on a five-minute call, a gain of nearly 7%.

While that gain may not seem dramatic, Badger said that in a volume-based environment, any amount of savings can return significant income. A small percentage gain can translate into one more call each day for each agent, which translates into a software payback time period of 12 months.

Armed with the results from the prototype, the team was fully assembled in November 2005, and application development for the first formal application began. The project involved interaction with Afni's client's proprietary billing application. In March 2006, this application was deployed to 800 desktops. OpenSpan's footprint on each desktop was approximately 2MB of RAM and 50MB of hard disk space, and each agent machine typically had 256–512MB of RAM.

For its next project, involving a major telecommunications company, Afni created a graphical Agent Control Bar that consolidated multiple pages of underlying CRM information that was determined to be the most likely to be used during a call.

Qualitative Gains in Addition to 12-Month Return on Investment

Badger said the cost of OpenSpan, including training and developer time, was recovered in the first year of deployment without requiring additional hardware or upgrades for Afni's agents. For Afni's first deployment, the average call time improved 43% for new agents, dropping from 700 seconds to 400 seconds. A subsequent engagement for a telephone company yielded a 60% improvement in call time, from 300 seconds to 120 seconds.

But the value of OpenSpan isn't just in the ROI. OpenSpan also gives Afni the ability to do things it couldn't do in the past, including controlling and modifying client applications as well as automating processes in client applications. OpenSpan also allows Afni to deal with a wider range of client application environments, especially those dominated by Windows and Web-based applications, using a single tool.

Badger said that OpenSpan is very different now than it was a few years ago, and Afni's usage of OpenSpan has grown with the new versions. Initially, developers had to do additional coding to address specific problems and applications, such as a Windows application that had an internal browser. For the future, Badger would like to see a debugger that would make it easier for developers to discover and solve problems (promised in the 3.2 version slated for release in November 2007), but he sees OpenSpan as a sound platform for the future. No tool can handle all situations, and Badger noted, "We worked with them, they worked with us." Most of the specific problems were solved within a few days.

The similarity of OpenSpan's development environment to Microsoft's Visual Studio was also an important advantage because Afni's development team was so familiar with Visual Studio.

Afni has now completed seven or eight projects using OpenSpan and has several additional projects lined up that will keep the development team busy for the next 6–8 months.

Shepherd Chartered Surveyors

Shepherd Chartered Surveyors, a member of United Kingdom–based Shepherd Direct, has 270 employees. The 17-year-old company does property valuations and home inspections in the United Kingdom for banks, lending institutions, and consumers.

We interviewed Shepherd's Group IT Director Martyn Wells for this profile.

Microsoft Shop Integrating Legacy and Web Applications

Before the introduction of OpenSpan, Shepherd's contact center agents worked with a thoroughly Microsoft infrastructure, using SQL Server and other Microsoft solutions, as well as three third-party Web applications, XIT2, VALNET, and QUEST, to process surveyor reports and transmit them to the back-office systems of the lenders who used them. Each system had an independent workflow management system associated with it.

The agents had to enter survey and valuation information into reports and distribute them across a variety of internal IT systems. They did this by manually rekeying data into each internal application and then replicating it onto the systems associated with the individual lenders. The process was labor intensive, time consuming, and error prone.

Shepherd wanted to streamline the processes and become more competitive by cutting the survey delivery turnaround times. The key to this strategy was to unify internal systems. The company did this through a unifying, proprietary front end called SIMSY, which is a case management tool with database, routing, and workflow. OpenSpan's role would be to link SIMSY to the back-end applications by automatically taking information from SIMSY and populating the appropriate case data into XIT2, VALNET, and QUEST, which are the systems that communicate with most of Shepherd's lender customers.

Internal Development Project Might Have Cost $320,000

The IT staff estimated that developing an internal solution to the problem would have taken anywhere from 140 to 400 development days, with a cost estimate of up to $320,000. In addition, the project would have required a substantial lead time, while draining the resources of Shepherd's small IT staff.

OpenSpan benefited from the arrival at Shepherd in April 2007 of a new CEO who came from a major bank that used OpenSpan to process credit card payments.

Represented by Attachmate in the United Kingdom, OpenSpan was asked to do a comprehensive, noncompetitive proof of concept (POC) involving data retrieval from two browser-based applications and one client-server application.

With the business side and IT working together, Shepherd developed the specifications for the POC in four days. On OpenSpan's side, it took three days to get the POC up and running. The specs were tested for two weeks and then deployed to 12 users who tested the prototype for two weeks.

The company completed the POC in May 2007 and signed a per-desktop licensing agreement in mid-June. By the end of June, Shepherd had extended the POC and deployed the first formal solution using OpenSpan, which had a footprint of 26MB of storage and 27MB of RAM on each computer.

There was no initial training for OpenSpan. Since the initial project, four members of Shepherd's IT team, two senior developers and two junior help desk people, have been trained. Wells said OpenSpan is easy enough to use for a trained business analyst to do the development work.

Results: Year One Cost Already Amortized; 20% ROI Projected

ROI analysis is important to Shepherd. The company needed a 12% ROI for the OpenSpan deployment to be cost effective. That analysis involved the cost of people, processes, and systems. Wells said that, within the first few months, the year one cost had been amortized, and Shepherd was projecting a 20% ROI within 6–9 months of the deployment.

Aside from the ROI, the deployment resulted in improved data quality for the lenders, fewer support calls, faster report processing, and reduced staffing. Wells said that some personnel were redeployed to other parts of the business.

Extension of Deployment Waiting for IT Revamp

While there is a backlog of business requests that will take several months to complete, a broader extension of OpenSpan is waiting for a review of Shepherd's data structures and its data mart.

Shepherd eventually wants to use OpenSpan to bridge other non-API applications to its SIMSY case management system.

Like Afni, Shepherd wants to see the addition of a debugging environment similar to that of Visual Studio (OpenSpan expects to include a debugger in the 3.2 version slated for release in November 2007).

FUTURE OUTLOOK

OpenSpan's unique approach to process automation makes it a vendor that should be considered for application integration and process automation projects that involve the need to improve productivity and quality for groups of users who work on repetitive, process-oriented tasks requiring the frequent use of multiple applications. Beyond contact centers, this product can be used to address a variety of areas, including sales and structured decision making, lending, underwriting, and claims evaluations.

ESSENTIAL GUIDANCE

OpenSpan's customers offered the following advice to other enterprises facing similar decisions:

- **Make sure success criteria are established.** Afni didn't just look at ROI as its sole success metric. The company also looked at key qualitative factors in addition to ROI. For instance, OpenSpan enabled it to modify client applications and processes that had previously been inaccessible. The opening up of the client IT environment paved the way for process efficiencies and for the leveraging of those efficiencies into other client engagements.

- **Meet with the business side, brainstorm the requirements, and then test the solution.** At Afni, the company enlisted a group of its call center agents to do live testing, which proved to be a valuable source of feedback.

- **When faced with an integration problem, look for solutions that promote reuse.** OpenSpan's ability to expose application functions and turn them into reusable components helped Afni to boost its value to its customers. It allowed testing without a major investment and gave Afni a prototyping system. Now Afni's customers are looking at what Afni is doing with its applications and processes for best practices and possible adoption.

- **Develop a POC and live prototype.** Shepherd did some baseline financial metrics and developed a POC that was a partial solution to some key integration challenges. Rather than using a POC as a competitive analysis tool to help determine the solution with the best fit, Shepherd used the POC to verify OpenSpan's capabilities and begin live testing with a limited set of users for two weeks. That testing gave Shepherd strong indicators about the ultimate benefits the company could expect from a wider deployment of OpenSpan.

  Afni also did a prototype and live test to give it the confidence needed to move forward into a larger project. The prototypes benefited both organizations because they highlighted potential problems, helped in knowledge acquisition, improved planning, and helped manage expectations.

- **Take advantage of training opportunities.** Shepherd was impressed with both the speed and level of training provided by OpenSpan. Wells urged prospective customers to take advantage of the quick-start training — two days of formal training plus two additional days of more informal work and additional onsite consulting.

LEARN MORE

Related Research

- *Worldwide Business Process Management Suite 2007–2011 Forecast and 2006 Vendor Shares* (IDC #207954, August 2007)

- *People, Process, Information: Are Organizations Focused on Their True Needs?* (IDC #205429, February 2007)

Published Under Services: Business Process Integration and Deployment Software with Vertical Views; Business Process Management Software