

# BWH News Letter

Volume 6 Number 1

## The SaaS, “Software as a Service”, Model

We have heard how “Software as a Service” (SaaS) is becoming the new model for delivering IT services, but it has not been clear (at least it hasn’t been to me) what this change means to the Retailer CIO. As I walked the fringes of the NRF Big Show this January, I talked to a lot of the small vendors in their 10x10 foot booths who are developing applications under the SaaS model. I use SaaS to mean the technology model where a complete business function can be delivered over the Internet or from independent applications on the Retailer’s platform. All the retailer has to do is provide the transaction data necessary to drive the service and be ready capture its results.

Many CIO’s are battling tightening budgets against fixed maintenance and licensing fees for applications they bought over previous years. Their business organization has become “hooked” on some of these applications from both the business and technology perspectives. This creates a difficult situation for the CIO who must continue providing service while reducing costs despite the fact that much of the IT budget is comprised of fixed licensing fees paid to outside vendors. The only budget flexibility is with the internal staff which has probably already been reduced. SaaS promises the retailer CIO a way out of this box.

The best analogy I can come up with for the impact of SaaS on the computer software industry is the effect of “garage bands” on the music industry. The classic model for the music industry has been the heavily promoted musician marketed through the support of a record label. Labels used a large advertising budget, product tie-ins, and viral merchandising to promote the musician. Some might say this model went too far as promotion began to outweigh talent and groups who lip synced their concerts like Milli Vanilli became popular. Now, garage bands have proven there is plenty of music talent out there and the spread of audio production software, MP3 players, and the Internet have made it possible for anyone to distribute their own recordings. The classic music distribution model is being challenged.

The classic music model parallels the established software companies such as Oracle, Microsoft, and IBM. Much of the expense associated with the classic software model has been marketing. The typical software company spends as much as 80 percent of its revenue on marketing. Personal business contacts and trade shows are used to find new customers and retain current customers. The software vendor must continuously monitor customer accounts to make sure their needs are met and that customer plans to use different software are avoided.

SaaS has the potential to replace the classic software distribution model with the software equivalent of the garage band model. Colleges and High Schools have been turning out computer savvy students for decades. Internet based hosting services now provide redundant hardware and software platforms on a subscription basis. These are often located in “undisclosed secure locations” that provide 24/7 availability from almost anywhere in the country. This means the production and distribution costs for software have declined immensely and a few talented individuals with in-depth knowledge of a business process can develop software to provide unique services. Now, specialists with detailed experience in a business area can offer software services that meet the specific requirements of a particular department. The challenge for the Retailer CIO becomes avoiding the Milli Vanilli’s and finding the service providers with superior offerings.

## A Common Business Model

The key to making the SaaS model work is having a clear understanding of the role of each service. Just as you manage the business processes, the SaaS model requires clear responsibilities for each service in order to avoid functional overlaps and minimize conflicts. Retail industry groups need to take the lead in defining the service requirements for their members. By using standard industry models for fashion, general merchandise or supermarket retail, the various service providers can develop their service to meet specific needs.

# BWH News Letter

Volume 6 Number 1

Just as the retailer may use an outsourcing service to care for the restrooms, they can use a software service to perform labor scheduling. To do this, the retailer needs to find SaaS vendors who have built their service with the same business model in mind or who are willing to adapt their service to the retailer's model. For instance, one of the functions necessary for warehouse purchase order processing is to assign a receiving door for the delivery. Should the receiving door be assigned by the purchasing system when it issues the PO, by the transportation management system when the trucker is assigned the pickup, or by the warehouse management system when the trucker confirms they are enroute? There are valid reasons for doing it at each of these steps and a retailer has to decide how they want to approach it. The best answer may be for the retailer to manage the receiving door schedule and accept inquiries and reservation requests from all the various external service providers.

Data ownership is a key point when considering SaaS. Just as the retailer expects the restroom service to provide their own mops, buckets and cleaning supplies, software service providers will own certain pieces of data within their service. Other pieces of data are like the toilets and sinks will be owned by the retailer. This means that things like item attributes or personnel skill levels might be passed to the service provider from a central database maintained by the retailer. This "persistent data" will be shared across various service providers so that all providers have one common definition for the various data. The service provider will return transaction data to update the various databases maintained by the retailer.

## **More Than Your Really Want to Know**

So when you get down to the really technical level, what does all this mean?

"The SaaS model means that retailers must have a Logical Object Class model which is supported by an Enterprise Service Bus that is capable of sharing transaction and object data with both internal and external services."

Huh?

This says that everyone must agree on what objects comprise a particular type of retail business, what their attributes are, and what events or procedures can cause attribute changes (a change of state). Services that manipulate these objects can be executed by either internal or external service providers and can be developed by the retailer or third party developers. The ESB serves as the "transportation hub" providing the interfaces, archives, and translation services necessary for transactions to flow between services. The ESB should be owned by the retailer, either at their own data center or through an outside "service hub".

Some of the services may be event driven and initiated according to scheduled times or as the result of a chain of events. Replenishment activities may be initiated based on time of day or triggered by an "empty shelf" event. RFID and expansion of in-store sensing devices that detect temperature changes or movement of merchandise will produce a whole new set of event driven services as "process driven" activity is enhanced with "event driven" activity.

So the job of the retail CIO is to put in place the ESB to support their industry object model. Fashion, general merchandise, and supermarket retailers will likely have different combinations of logical models. If nothing else, the emphasis will be different so the extent of object features required between industries and retailers can choose services that meet their requirements. To get there, the retail CIO must first choose an ESB solution, map existing services to the industry model they have chosen, and then put in place the ESB. Once the ESB is in place, they can finally go shopping for the garage band that is playing their tune.

The technical platform for SaaS is built around XML. With the ESB in the middle, providing translation and integration services, XML data should be easily convertible to meet individual service provider requirements. Software adapters that connect various platforms will manage the individual protocols that connect internal and external services.