Pipeline
Knowledge Is Power

www.pipelinepub.com Volume 4, Issue 10

Data-Driven Rapid Product Assembly - the Key to Bringing New Products to Market Quickly and Cost Effectively

by David Jacobs and Joe Frost

In the increasingly competitive environment communication service providers (CSPs) live in today, the ability to reliably meet customers' needs, drive loyalty and derive as much revenue as quickly as possible is paramount as traditional revenues disappear.

Internet-based and -derived services present a considerable opportunity for CSPs that can successfully harness them; conversely they represent a considerable threat to those that cannot. This means being able to provide customers with compelling, personalized and timely services which extend well beyond the boundaries of traditional communications services. Twenty-first century communications providers will compete on the breadth of services they supply, even within a market niche, as well as on the eye-catching innovation behind these services.

In today's globalized, internet-driven market, with so many competitive services on offer, CSPs have neither the internal resources nor the brand reach to develop an entire "long tail" of services by themselves. As a result, they will increasingly need to feed their product pipelines with services sourced from third parties and they will need to be able to "mash up" these services with their own internal services into new, more attractive and differentiated products.

Communications service providers are fast becoming intermediaries in a digital value chain in which their traditional voice and data services are bundled into more sophisticated, value-added products almost as a given. While traditional communications services are contributing a shrinking proportion of a network operator's revenue, the fact that the world can't function without them puts the communications service provider in a powerful position when it comes to leveraging its customer relationships to sell a wider range of products, providing that the operator has the ability to bring new products quickly and cost-effectively to market.

Rapid product assembly is therefore a critical capability for network operators playing this intermediary role.

Rapid Product Assembly Defined

Rapid product assembly is the ability to construct new products, or to tailor products to a customer's specific requirements, from a well-defined set of components, guided by rules that govern which components can work together. The rapid product assembly process is agnostic as to the source of the components it works with, whether these have been developed in-house or externally by third parties, as well as to how these products are eventually used. Speed of development comes from the fact that products can be assembled from multiple, reusable components, limiting the amount of new component creation needed to bring a new product to market. The reassembly of components in new ways supports innovation and the CSP's ability to refresh product offerings on a regular basis.

New components can easily be defined to the rapid product assembly process and the rules base updated, continually expanding the feature sets available to product developers for the creation of new, or the enhancement of existing, products. Companies can therefore maintain a "long tail" of products in their portfolio, since products can quickly be customized for many different markets. Rapid product assembly supports two important corporate goals: faster time-to-market and increased revenue generation from a broad product portfolio.

The end goal of many CSPs is to put product creation into the hands of end-users, with the customers themselves specifying the components they would like assembled, or, to use a term from the Web 2.0 movement, "mashed up" into their own, personalized products. Such user empowerment supports the further corporate goal of improving the customer experience, increasing customer "stickiness", or loyalty.

The concept of rapid product assembly is an important contributor to business agility and the business case has been proven over the past decade in different industry sectors including defense, automotive and electronics/IT. Companies such as Dell Computers have demonstrated the power of a rapid product assembly capability in the physical world, while the concept is being applied in the virtual world by Internet companies, including Amazon and Google. Rapid product assembly is now under evaluation by leading communications services providers.

Implications of Rapid Product Assembly for Operational Systems

Rapid product assembly requires CSPs be able to model internal and external services as reusable components that can be assembled together into products. CSPs are identifying four types of product components that they would like to include in rapid product assembly:

- Network services, both legacy and next generation services. The latter are likely to be decoupled from the network in an application layer. The former will be tightly bound to the network resources they run on. In the case of next generation network services, service logic and supporting network elements can be modeled as separate reusable components.
- IT applications, which are increasingly SOA/Web services based. Applications written as Web services are inherently easy to model as components.
- External content from professional third party sources, such as ringtones,

video streams.
- User-generated, "Web 2.0" content.

The discovery and specification of which components should be assembled into new products is only one aspect of a rapid product assembly process. While operators may be able to create new services quickly in this way, unless they can also deliver such composite products, on demand, to customers, the concept will fail in its objectives of accelerating time to revenue and developing the customer relationship. Preparing the operational systems, and especially the fulfillment system, to support rapid product assembly is as important as re-engineering the operator's service creation process.

When a customer orders a product, possibly personalizing all its constituent parts (product components) through a web portal, every component in the product will need to be provisioned. Compared to provisioning a single communications service, such as voice or a broadband connection, the provisioning activity facing operators in the past, fulfilling a next generation assembled product is a considerably more challenging process. The fulfillment system will need an overview of all the components that make up the product and the ability to decompose the product into its constituent components, coordinating the provisioning of each individual component in order to fulfill the product as a whole. This includes being able to validate what the customer has requested is possible in their environment, understanding and operating on combinational factors such as managing conflict across shared resources, and being able to recognize and reuse orphaned resources, for example.

Although the ideal is for rapid product assembly to support the discovery and reuse of any type of component across the four categories identified above and potentially others in the future, CSPs implementing this concept are likely to want to do so a step at a time, starting with their network services. CSPs network services today are typically fragmented, running on different platforms and often in different networks, making it difficult to maximize their value by reusing them in multiple product contexts.

As a first task, CSPs need to identify, model, and expose their network services as components available for rapid assembly and this means making information (metadata) about the services available through a product catalogue. They will then need to revisit their traditional approach to fulfillment to ensure that they can deliver each service when it is specified as part of an assembled product.

Can Existing Fulfillment Systems Support Rapid Product Assembly Today?

The simple answer is that traditional fulfillment systems are typically attached to a particular service. When CSPs launch a new service, they often introduce new BSS/ OSS implementations to accompany it, this being the fastest way of bringing the service to market. CSP's argue, rightly, that there are a number of benefits to maintaining this "silo" approach: they can get a service up and running quickly, demonstrating its potential, and initial overall capital investment is low. If a CSP wants to offer a limited portfolio of services to a small number of customers, this approach remains viable. However, if it wants to scale the service and to be able to

combine it with others in new product bundles that deliver convergence, not only is this approach inflexible, typically resulting in slow customer adoption for example, but the total cost of ownership associated with it will soar. CSPs will potentially have to then try to integrate multiple fulfillment systems to support product bundles, and if they do this in a process-driven way, they will be narrowing their options as to which network services can be assembled together into products.

Even when CSP's have rationalized their fulfillment systems, selecting a single system to support multiple products, the product is usually defined to the fulfillment process as a monolithic entity and tightly coupled with that process. The fact that the product exists inside a discrete process and has no external, data-driven description means that it cannot be reused as a component in a larger, value-added product. Not only is the product unable to participate in a rapid assembly process, changing it is expensive because this means also changing the process.

Data-Driven Fulfillment: the Key to Rapid Product Assembly for CSPs

If CSPs wish to broaden their service portfolios and reap the benefits of rapid product assembly, they must embrace a data-driven approach to product creation and fulfillment. Rapid product assembly needs to be supported by a single, joined-up fulfillment process that can handle any product assembled from multiple different service components. This means that the process has to exist independently from any specific component (service), and that it understands components as pieces of data it can manipulate in different ways, according to well-defined rules.

Data-driven fulfillment does not mean that CSPs need to have a single fulfillment system capable of provisioning every possible network service and operators do not have to rip and replace existing fulfillment systems to gain the benefits of rapid product assembly. When CSPs put in place a central technical product catalogue and common fulfillment process, these can use standard middleware or proprietary connectors to talk to individual fulfillment systems responsible for provisioning and activating specific services.

Data-driven fulfillment is initially slower and more expensive to implement than the traditional silo process-driven approach to fulfillment, since CSPs will need to build a product model, populate a technical catalogue, define component assembly rules, and put in place standards-based integration mechanisms to federate multiple sources of service component data and underlying provisioning systems. The overriding benefit, however, is that CSPs will become more flexible and agile in their ability to bring new products to market, which they can do more cost-effectively and quickly.

Reducing the cost associated with launching new services is a major consideration as CSPs move into a next generation service world. Here they will generate revenue from a "long tail" of inexpensive, niche services rather than from a few, costly, generic services: inexpensive services do not justify the development of their own OSS/BSS infrastructure. Data-driven fulfillment, with its ability to support the rapid addition of new components to the technical product catalogue, and which does not require any changes to the common fulfillment process as a result, is far more cost-effective in the longer-term than traditional fulfillment.

If you have news you'd like to share with Pipeline, contact us at editor@pipelinepub.com.