

Skyway Builder from Skyway Software



Reaching for the sky

Services Oriented Architectures (SOA) rarely start from scratch. In most enterprises, they are built in gradual steps as part of an overall migration and architectural strategy. Along the way, existing legacy systems must be enhanced to support the required interface technologies, service gaps have to be filled by building new systems or acquiring and leveraging external services through third-party providers, and all elements in the architecture must be orchestrated and combined to form the final catalog of services. One solution that helps address these challenges is Skyway Builder from Skyway Software.

Skyway Software takes a three-layer approach to SOAs: Produce, Manage, Extend. The Skyway Builder platform is the Produce Layer and provides the capabilities required to build, integrate, provision, govern, and deploy composite applications in an enterprise.



WRITTEN BY

BRIAN BARBASH

folder offers access to Skyway Builder's configuration and administration components. While the paradigm takes a while to get used to, it's effective in developing the specific kinds of applications Skyway Builder creates.

Applications created with Skyway Builder consist of the following objects:

- Structures and Collections – These objects make up an application's object model

- Web Processes – Objects that can be used to build a Web-based user interface for composite applications and/or WSRP-enabled portlets
- Processes – Defines services that manage control flow and business logic
- Web Services and Java Services – Touch points to external systems using Java and Web Services technologies

For purposes of this review, I will use these components to create a simple stock portfolio manager. To simulate a service-oriented architecture, the portfolio manager will serve



FIGURE 1 | Skyway Builder IDE

mainly as a data store and rely on a set of external Web Services for data validation, pricing, and other information.

Modeling Data

Central to any system or application is the data model. Skyway's Structures and Col-

Working with Builder

The first thing developers will notice about Skyway Builder is the user interface. Traditionally, development methods require some form of file manager or explorer type of layout. Typically, a hierarchy of projects and objects is located in a tree-view from which components on which to work can be selected and opened. Skyway Builder uses a desktop paradigm (Figure 1). In this virtual desktop, there are two system objects, Control Panel and System, while each individual project is represented as a folder. The Control Panel offers access to basic system operations including Debug, Deploy, Import, and Export. The System



Company Info

Skyway Software
208 S. Hoover Blvd
Suite 100
Tampa, FL 33609
Phone: 813-288-9355
Fax: 813-288-8266
www.skywaysoftware.com

lections components build an object model. Structures are analogous to a class or JavaBean, while Collections are the containers in which multiple instances of a Structure reside at runtime. Skyway Builder allows structures to be created using a wizard, or to be imported from an external Web Service, Java Service, XML Schema, or data source.

In this example, data objects are required for all Web Services to execute. Skyway Builder can import a WSDL file from a remote URL or local filesystem. Three services are used in the portfolio manager: a locally hosted service for storing stock and transactional data, and two external services for getting the latest price and market data.

Integrating with Services

As mentioned above, Skyway Builder provides a way to import service definitions from WSDL files and Java classes. To execute methods from one or more Java classes in a JAR file, all dependent JARs must be in both the Skyway Builder client classpath and the Skyway Server runtime classpath. For our purposes, only Web Services will be executed.

The standard Skyway Builder Process object provides the entry point to all Web Service invocations and the mechanism for controlling application flow. Figure 2 shows the Process object designed to handle calls to local Web Services from within the Portfolio Manager application.

A decision tree exists at the entry point of the process that determines which Web Service/Operation combination to execute based on an action variable. Since the Web Services executed in this process are document-style interfaces from the same provider, the serviceInput variable is passed to each and the results collected in the serviceOutput variable.

Skyway Builder's Process objects contain a wide array of additional functionality to manage control flow, data manipulation, integration, and error handling so developers can create very complex logic flows. And any Process in Skyway Builder can easily be exposed as a Web Service simply by setting a flag in the Process settings. This flexibility lets Processes be shared across an organization.

User Interfaces

The user experience of composite applica-

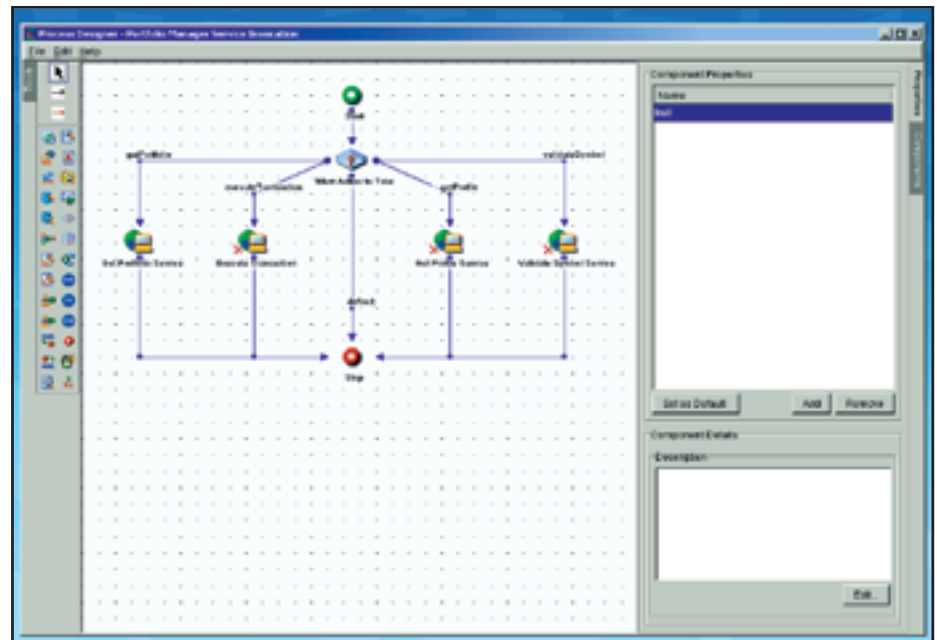


FIGURE 2 | Process Designer

tions in an SOA should be no different than their “stove-piped” counterparts. However, the back-end design and architecture of these interfaces can be. Skyway Builder brings to the UI a componentized approach well-suited to SOAs called Web Processes. A Web Process consists of one or more Web Components and possibly a collection of Events. Individual Web Components define a series of fine-grained steps. Examples include collecting and laying out HTML elements, modifying session and local variables, and executing and gathering data from other processes.

Web Processes, like their Process counterparts, are based on a flow and created in a similar designer. In the portfolio manager, an example Web Process for entering a transaction executes the following steps:

1. Accept input parameters
2. Validate stock symbol externally
3. Check latest price and calculate total cost
4. Add the transaction to the Portfolio Manager
5. Return confirmation

With the steps defined, the developer can focus on creating the HTML to display the UI itself. Skyway Builder provides an internal HTML authoring tool that developers use to access the structures, collections, and variables, as well as invoke the actions defined in a Web Process.

Summary

Skyway Builder from Skyway Software provides a solid collection of tools for working in a Services Oriented Architecture. Its focus is on the composition of higher-level services from existing fine-grained services. While the GUI paradigm isn't traditional, the transition to the virtual desktop concept is smooth and it proves to be effective in this context. Overall, Skyway Builder is a solid foundation for building services-oriented architectures.

Skyway Builder is a platform now rather than just a tool... and now actually builds, integrates, provisions, governs, and deploys... and in addition to web processes - web based user interface for composite applications and/or WSRP-enabled portlets.

Its focus is on Service Based Components of any granularity, and is built from the ground up for this purpose; not just their creation but also the enterprise provisioning of resources and governance of those resources. From very small services to full applications.®

About the Author

Brian R. Barbash is the product review editor for *Web Services Journal*. He is a senior consultant and technical architect for the Envision Consulting Group, a management consulting company focusing on contracting, pricing, and account management in the pharmaceutical industry.

■■■ bbarbash@sys-con.com