zapthink zapnote

ZAPTHINK ZAPNOTE™

INTELLIUN
MODEL-DRIVEN DEVELOPMENT FOR
SERVICE-ORIENTED BUSINESS APPLICATIONS

Analyst: Jason Bloomberg

Abstract

Taking a model-driven approach to implementing and representing Services is a core Service-Oriented Architecture (SOA) best practice. The central technical challenge of SOA is building and maintaining the business Service abstraction, and maintaining that abstraction relies upon creating and maintaining the Service model. Ideally, organizations should be able to use the Service model to create working implementations of the underlying Services, as well as for representing the composition of those Services into Service-Oriented Business Applications.

Intelliun's model-driven approach to Service development supports this vision for the Service model. Intelliun helps to transition firms that have well-developed, but entrenched experience in traditional software development to a Service-oriented approach that can help companies achieve the agility benefits they so desire. Intelliun's VE Software Platform's model-driven approach follows an important SOA best practice.

The Model-Driven Approach to SOA

Organizations of all sorts are touting the benefits of Service-Oriented Architecture (SOA). Yet, achieving these benefits has proven to be a complex task for many IT organizations. Part of the challenge lies in the fact that SOA represents more of a change in the way things are done rather than a new technology or product that organizations can simply install, configure, and deploy. But a greater part of the challenge is that SOA requires changes to the very way that the business manages IT as an asset rather than a cost center, requiring changes not only to the method of IT development and delivery, but also IT management, governance, and value creation.

One of the methods that cautious companies that desire the benefits of SOA are adopting is to incrementally separate the various concerns of IT and business. Developers have gotten used to separating the concerns of data and application logic, and increasingly presentation and application logic. Many people are currently grappling with separating the concept of business process from the underlying IT systems that implement aspects of those processes. But even more fundamental is the idea of abstracting the requirements of the business that continue to change from the underlying IT systems and implementations.

Mastering ongoing change by abstracting the business requirements on the one hand and IT capabilities on the other is the first concrete step on the path to SOA. This abstraction requires a representation that allows the business users to communicate their desires for the IT capabilities as well as a means for IT implementation teams to map their capabilities and describe how they are composed to meet the current needs of the business. This representation often comes in the form of a model.

Increasingly, Model-Driven Development (MDD), and as a larger concern, Model-Driven Architecture are important improvements upon today's tightly-coupled, brittle, and inflexible waterfall-style software development lifecycle that is poorly suited to meeting the continually changing needs of the business. While ZapThink often talks about the model as a runtime artifact that can serve as a direct representation of the system and business as it currently operates, companies can take the first step to this vision by leveraging the power of a model during their design-time activities. These models serve two key roles: to shorten the development time, thus increasing the number of iterations in a development cycle, as well as making more precise and simplifying the mapping between business requirements and IT implementation.

Intelliun's Model-Driven Approach

Intelliun is one of the first vendors to champion MDD. MDD is an approach to application creation that uses visual models to introduce substantial efficiencies and rigor to the software development process. When using the Intelliun approach, application developers use visual models to reduce the time needed to create applications, thus reducing costs and improving software quality.

.

Intelliun's MDD Platform, *The Virtual Enterprise* (VE), is essentially a software development and deployment platform that facilitates end-to-end application development using visual models. VE provides visual modeling tools for capturing business logic in the Unified Modeling Language (UML), which is a widely adopted modeling language for producing a variety of different types of applications, including Service-Oriented Business Applications (SOBAs), which are composite applications built with Services in a SOA environment.

VE supports three main activities that form the core of the software development lifecycle (SDLC): analysis, construction, and deployment. The analysis activity concentrates on identifying, scoping, and validating the business requirements. The construction phase concentrates on taking the analysis artifacts and turning them into a concrete and executable computer implementation. Finally, deployment focuses on making the implementation operational and ready for its intended use.

The construction phase in particular concentrates on taking the analysis artifacts and turning them into a concrete and executable implementation. It's possible to divide the construction activity into sub-activities, which can vary in formality depending on the adopted SDLC process and the overall size and complexity of the developed software. These activities are:

> ➢ Developing an abstract design to satisfy the functional requirements independently of the architecture

> ➢ Developing an architecture that satisfies the nonfunctional requirements

> ➢ Developing a detailed design by mapping the abstract design into the target technology artifacts in accordance with the architecture

> ➢ Coding the detailed design in the target programming language.

VE narrows the construction activity to developing the abstract design, focusing primarily on the application logic, thus relieving the development team from the remaining tedious and error-prone activities. Consequently, VE reduces the number of required skill sets and the overall team size. Furthermore, when implementing SOBAs in the context of SOA, it is essential for the abstract design to represent the business Services that make up the SOBAs in the form of the Service model. In other words, the Service model becomes the central abstraction for SOA, in that it separates the creation of SOBAs above the business Service abstraction from the underlying implementation details for coding the Service implementations in software.

Because of its model-driven approach, VE is particularly well-suited for SOA-based application development. VE narrows the construction activity to the creation of the abstract design, focusing primarily on the application logic, and relieving the development team from focusing on implementation details. Consequently, VE reduces the number of required skill sets, and the overall team size.

VE's use of UML, furthermore, provides a way to organize functional requirements as use cases, and represent the external entities (typically users) as actors. Actors can trigger the execution of use cases, or respond to use case requests. Consequently, in the abstract design activity, use cases can identify the collective state and behavior of the system, along with the rules and constraints that apply to the system.

VE takes an object-oriented approach for capturing and representing the abstract design of a system. However, VE captures behavior as processes and operations, while representing rules and constraints as formulas. VE uses UML for representing the abstract design artifacts, namely, the class diagram and activity diagram for objects and processes/operations respectively, as shown in the figure below.

Intelliun VE Designer Object Model
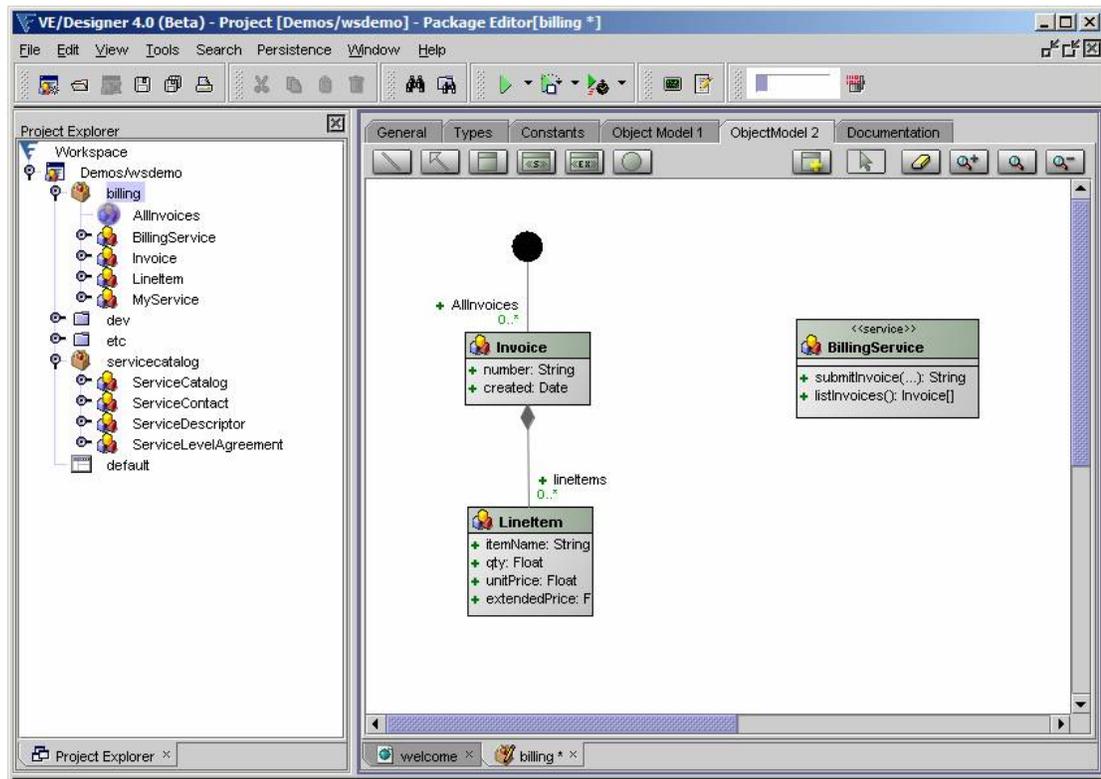
Source: Intelliun

By creating models in VE, then, application creators actually build a fully functional module including a ready to use Web browser interface, Web Service interface, and even a personal digital assistant (PDA) browser interface, if necessary. VE also creates a portal to capture the behavior of the application, and to maintain additional information for each access device. For example, a portal might have look-and-feel information for a Web browser, while it maintains a WSDL for a Web Service client.

In addition, by clicking on the database deploy button in the VE Designer and answering a few questions regarding the target database, VE automatically generates the appropriate database schema, object-relational mapping, and all of the SQL statements necessary to carry all of the functionality that the model specifies. Application creators can apply the same approach to support the data tier underlying any SOBA.

Furthermore, by separating all of the interface-related logic from the application logic and abstracting out all of the architecture and technology-related constructs, VE creates an environment for expressing an executable form of application logic via objects, processes, and rules, as shown in the figure below.

Intelliun VE Designer Process Diagram
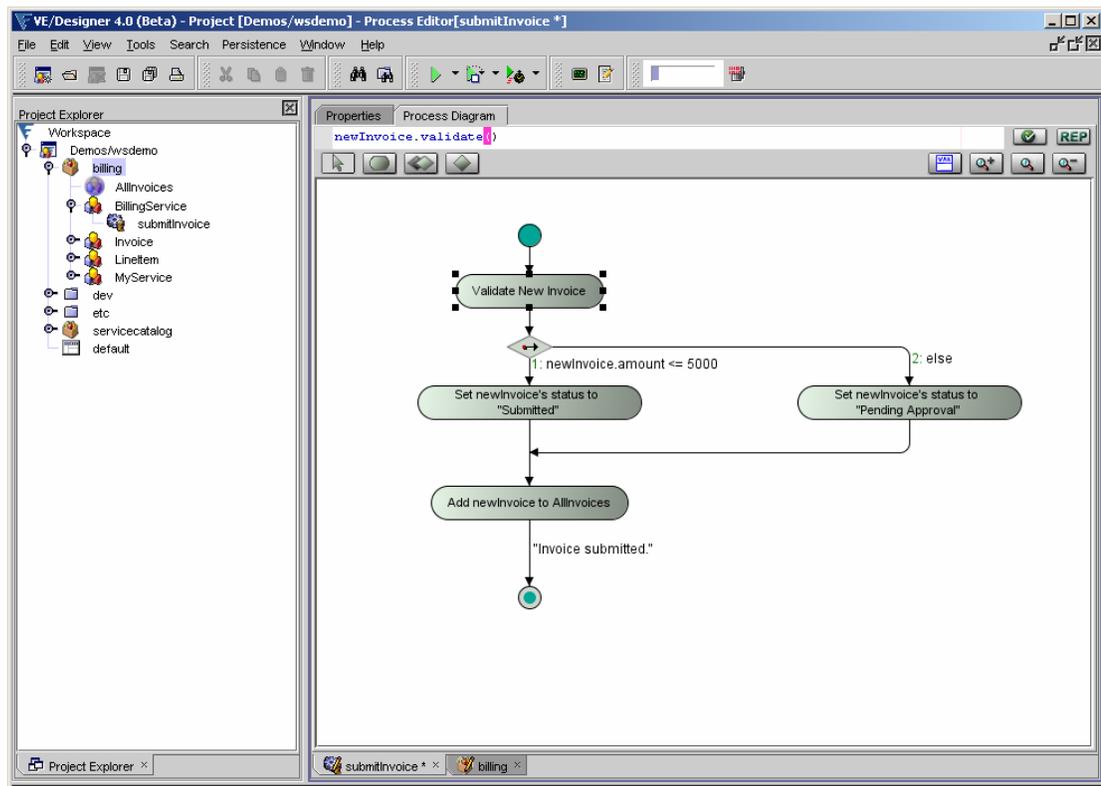
Source: Intelliun

VE also automatically generates the appropriate interface logic, and provides the application creator with WYSIWYG personalization tools. As a result, VE eliminates the need to know anything about HTML, JavaScript, SQL, Java, or the other various technologies involved in building n-tier applications, and also enables application creators to build Web Services without any knowledge of their underlying implementation.

Intelliun Virtual Enterprise

Overview:

The Virtual Enterprise (VE) is a Model-Driven Development platform that facilitates end-to-end application development using only visual models. Its support of Web Services and the creation of the Service abstraction make it a useful tool for building Service-Oriented Business Applications.

The Virtual Enterprise Features:

VE/Designer

- ➢ Develop Web applications using UML models
- ➢ Instant execution of UML models and validation of application logic
- ➢ Dynamic generation of the web interface and Web Services
- ➢ WYSIWYG Web personalization
- ➢ Dynamic generation of object- relational database mapping

> ➢ Formula auto-completion

> ➢ Support for embedding Java code and JAR files

> ➢ Support for embedding hand-coded SQL statements and stored procedures

> ➢ Built on top of Netbeans 6.1 IDE

> ➢ Model-Driven Web Services

> ➢ Code Generator for .NET (VE 6.0 Professional Edition only)

> ➢ Unit and remote testing framework

> ➢ Provides full localization capabilities

> ➢ PDF Document Generation

> ➢ Integrated XML API with convenient creation and query capabilities

> ➢ JMX Integration

VE/Server

> ➢ Runs on any J2EE Web and/or application server

> ➢ Runs on a any Java supported platform including Unix, Linux, Windows and AS/400

> ➢ Supports a range of relational databases including MS-SQL, Oracle, DB2/UDB, MySQL, Pervasive, and Sybase

> ➢ Supports SOAP and WSDL in both client and server scenarios

> ➢ Supports REST in both client and server scenarios

> ➢ Supports JMS for messaging and events

> ➢ Auto-generation of WSDL

Value Proposition:

> ➢ Simplifies development by reducing the number of required skills in the underlying technologies, specifications and standards

> ➢ Protects intellectual property investments from evolving technologies

> ➢ Reduces the development time, cost, and effort of Service-Oriented Business Applications and Web Services

> ➢ Increases application agility to better align with continually changing business needs

> ➢ Reduces QA time while improving application quality.

The ZapThink Take

ZapThink frequently writes that the greatest technical challenge of SOA is creating and maintaining the business Service abstraction. This abstraction maintains an agile representation of IT capabilities for the business, while hiding the underlying implementation

details. Furthermore, it is essential that this abstraction be maintained over time, as requirements continue to evolve.

An essential tool in the architect's toolbelt for maintaining this abstraction is the Service model, which represents both Services in production as well as requirements for new or changed Services. Ideally, the SOA team can use this model to actively implement the underlying Services. Without this MDD capability, it becomes increasingly difficult for the model to serve its purpose, and thus for the business Service abstraction to stay in place.

Therefore, Intelliun's MDD approach is well-suited for both representing the Service model as well as creating the underlying implementations of the Services that make up the model. Organizations who are struggling with creating Service implementations in the context of an abstracted Service model, especially if they have strong UML skills, should consider Intelliun for their modeling and application creation environment.

Profile: Intelliun	April 2008
Funding:	
Self-funded	
President and CEO:	
Iyad Jabri	
Products:	
The Virtual Enterprise	
US Address:	
5700 Granite Pkwy.	
Suite 200	
Plano, TX 75024	
URL:	
http://www.intelliun.com	
Phone:	
(469) 364-7300	
Contact:	
sales@intelliun.com	

Related Research

- ➢ *IDV Solutions* ZapNote (ZTZN-1224)

- ➢ *JustSystems* ZapNote (ZTZN-1222)

- ➢ *Skyway Software* ZapNote (ZTZN-1218)

- ➢ *Corizon* ZapNote (ZTZN-1216)

- ➢ *Nexaweb* ZapNote (ZTZN-1207)

- ➢ *JackBe* ZapNote (ZTZN-1204)

- ➢ *TenFold* ZapNote (ZTZN-1197)

About ZapThink, LLC

ZapThink is an Enterprise Architecture (EA) strategy advisory firm. As a recognized authority and master of Service-Oriented Architecture (SOA) and EA, ZapThink provides its audience of IT practitioners, consultants, and technology vendors with practical advice, guidance, education, and mentorship solutions that assist companies in leveraging SOA to meet their business needs and presenting viable SOA solutions to the market. We provide this audience a clear roadmap for standards-based, loosely coupled distributed computing – a vision of IT meeting the needs of the agile business.

ZapThink provides IT practitioners strategic insight and practical guidance for addressing critical agility and change management issues leveraging the latest EA and SOA best practices. ZapThink helps these customers put EA and SOA into practice in a rational, well-paced, and best practices-driven manner and helps to validate or recover architecture initiatives that may be heading down an unknown or incorrect path. ZapThink assists with solution vendor, technology, and consultant selection based on in-depth, objective evaluation of the capabilities, strengths, and applicability of the solutions to meet customer needs as they relate to EA initiatives and as they map against emerging best practices. ZapThink enhances its customer's skills by providing education, credentialing, and training to EAs to develop their skills as architects.

ZapThink helps to augment consulting firms' EA offerings and intellectual property by providing guidance on emerging best practices and access to information that supports those practices. ZapThink provides frameworks for product-based consulting based on ZapThink insight and research, such as SOA Implementation Roadmap guidance, Governance Framework development, and SOA Assessments, and provides a means to endorse and validate consulting firm offerings. ZapThink also accelerates consulting firms' efforts to attract, retain, and enhance the skills of EA and SOA talent by providing education and skills development

For solutions vendors, ZapThink provides retained advisory for guidance on product strategy, as well as marketing, visibility, and third-party endorsement benefits through its marketing activities, lead generation activities, and subscription services. ZapThink enables vendors to leverage ZapThink knowledge to transform their offerings in a cost-effective manner.

ZapThink's Managing Partners are widely regarded as the "go to advisors" and leading experts on SOA, EA, and Enterprise Web 2.0 by vendors, end-users, and the press. Respected for their candid, insightful opinions, they are in great demand as speakers, and have presented at conferences and industry events around the world. They are among the most quoted experts in the IT industry.

ZapThink was founded in October 2000 and is headquartered in Baltimore, Maryland. Its customers include Global 1000 firms and government organizations, as well as many emerging businesses. Its Managing Partners have worked at such firms as IDC, Saga Software, Mercator Software, marchFIRST, and ChannelWave, and have sat on the working group committees for standards bodies such as RosettaNet, UDDI, and ebXML.

Call, email, or visit the ZapThink Web site to learn more about how ZapThink can help you to better understand how SOA will impact your business or organization.

ZAPTHINK CONTACT:
ZapThink, LLC
108 Woodlawn Road
Baltimore, MD 21210
Phone: +1 (781) 207 0203
Fax: +1 (815) 301 3171
 info@zapthink.com
www.zapthink.com