

IBM Research Report

Solution Templates Tool for Enterprise Business Applications Integration

**Shiwa S. Fu, Jeaha Yang, Jim Laredo, Ying Huang, Henry Chang,
Santhosh Kumaran, Jen-Yao Chung**

IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Yury Kosov
IBM Software Group
Burlingame, CA



Research Division

Almaden - Austin - Beijing - Cambridge - Haifa - India - T. J. Watson - Tokyo - Zurich

Solution Templates Tool for Enterprise Business Applications Integration

Shiwa S. Fu, Jeaha Yang, Jim Laredo, Ying Huang,
Henry Chang, Santhosh Kumaran, and Jen-Yao Chung
IBM T. J. Watson Research Center, Yorktown Heights, NY, U.S.A.
Email: {shiwa, jeaha, laredoj, Ying_Huang, hychang, sbk, jychung}@us.ibm.com

Yury Kosov
IBM Software Group, Burlingame, CA, U.S.A.
Email: kosov@us.ibm.com

Abstract

Today's enterprises must meet the rapidly changing market demands while collaborating with much broader range of trading partners to maximize supply chain efficiency and improve services to customers and their profit margin. To accomplish these business goals, their business application systems, such as enterprise ERP, must integrate and adapt rapidly to meet these business demands and collaborate with other business system without major reimplementation effort. Our proposed approach addresses these problems by using model based approach of creating reusable solution template that consists of off-the-shelf artifacts to be quickly assembled as a solution.

In this paper, we introduce a Solution Template Tool that simplifies the life cycle for creating an integration solution, through the flexible design and customization of solution templates, and an interactive environment driven by wizards. By providing levels of abstraction, the proposed tool allows users to compose templates of platform independent model without worrying about implementation details. The tool transforms the composed templates into platform specific IT execution model and a deployable solution, thus easing the task of solution integration lifecycle. A solution template of UCCnet illustrates our study.

Index Terms: Solution Template, UCCnet, RFID, Platform Independent Model, Platform Specific Model.

1. Introduction

Today's enterprises must meet the rapidly changing market demands while collaborating with much broader range of trading partners to maximize supply chain efficiency, improving service to customers and their profit margin. To accomplish these business goals,

enterprise business application systems, such as ERP, must integrate together and adapt very rapidly to meet these business demands and collaborate with other business system without major reimplementation effort. To seamlessly integrate these disparate systems there is a need for an intermediate or middleware layer. Enterprise Application Integration (EAI) [1] emerges perfectly to fill this niche of market. Though EAI systems generally provide many packaged off-the-shelf components that can shorten the middleware development cycle and ease subsequent maintenance efforts, it still requires expensive and time consuming IT programming effort to develop and maintain EAI solution.

The Model-Driven Architecture has been advocated by the Object Management Group (OMG) [2]. The objective of the Model-Driven Architecture is to move the focus from programming to solution modeling. Our proposed tool employs model driven approach toward EAI solution at a higher level of abstraction of solution composition closer to the problem domain. The tool exhibits the following features: (1) levels of abstraction, (2) separation of concerns, and (3) reusable assets. By providing the levels of human-friendly abstraction of composition and development wizard, the users can focus on developing higher-level Platform Independent Model (PIM) while the tool transforms the PIM to Platform Specific model (PSM) that can be used to generate code. These abstractions also facilitates the participation of users with different skills at each stage, creating a separation of concerns that ties each skill with a different set of activities that are clearly defined and isolated from each other. To facilitate reusability we propose the use of Solution Templates [3]. The Solution Template, stored as an asset, is designed with customizable points to increase its flexibility by rapidly adapting to changes, and acting as the unifying artifact

throughout the life cycle of the solution to accumulate the useful knowledge that is gathered from development phase through the deployment phase. The tool is adapted as a WBI SE Tech Preview, and is available from IBM Developer Works website [4].

This paper is organized as follows. Section 2 describes the features of Solution Template Tool including different roles in solution composition. To facilitate the application business integration scenario, we choose to use the UCCnet as an example of solution composition in Section 3. The composition of Solution Template and the transformation of PIM into WBI SE runtime environment are introduced in Section 4. Section 5 presents the model transformation functions. Finally, the paper is summarized in Section 6.

2. Solution Template Tool

The Solution Template Tool is developed as a Eclipse-based plug-in with IBM Websphere Business Integration Server Express (WBI SE) [5]. It consists of two major components, PIM Composer and Reusable Asset Repository, as shown in Figure 1. The PIM Composer helps users to model and customize the Solution Template, navigate through WBI SE tools, and realize the solution template in WBI SE runtime. The Reusable Asset Repository stores solution templates and artifacts as reusable assets to ease future solution composition.

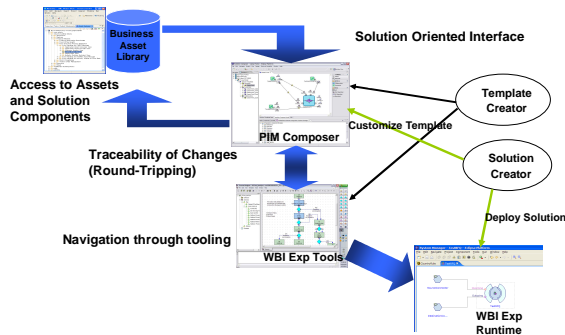


Figure 1 Proposed Solution Template Tool.

The tool maintains traceability between the solution template level artifacts and those platform dependent components in the runtime. For example, any property or interfaces change in the WBI SE components will trigger the corresponding artifacts in the Solution Template model to propagate the changes. Each artifact in the Solution Template will launch the associated WBI native editor to modify the

corresponding component which will then update the WBI runtime when realized.

Several users may participate in the creation of an integration solution at different stages, and each user's responsibility requires a different set of skill and has a specific mission in the process. There are three roles available in the tool: Template Creator, Solution Creator, and Component Builder. The Template Creator is usually a Subject Matter Expert with enough technical skill to abstract a given solution, and creates a corresponding template in a way that it can be used multiple times. The Template Creator also determines variability points and expresses them into this template. The Solution Creator makes use of templates created by the Template Creator. The Solution Creator is able to identify the right template candidate and may be able to perform the configuration tasks. He may be able to do some additional composition to complete the solution. The Component Builder is the more IT skilled user of the three, understands the underlying technology and is able to create new reusable components based on requirements. He will be able to create flexible components with its customizations points exposed as Points of Variability to be described in Section 4.1.

The tool provides interactive wizard (PIM Composer) for Template Creator to create new template, or locate and import the reusable template from the asset library. The business description and requirements that are available are gathered and matched against the Solution Template information to find closest Solution Template to be used. Importing the reusable template will be simpler and thus reduce the development time than designing from scratch. Also the wizard will accelerate learning curve by identify dependencies and guiding the Template Creator to locate them. The tool follow the top down design approach where the Template Creator can create high level design with the artifacts and then each artifact can be further refined.

3. Case Study

The Solution Template Tool is developed with WBI SE as the initial target platform to demonstrate solution lifecycle and reuse capabilities via integration with the asset library which contains the reusable Solution Templates and artifacts. The demonstration of prototype uses an example where SMB business supplier to publish their product description and order information in the UCCNet catalog as shown in Figure 2. UCCnet [6, 7] is a subsidiary of Uniform Code Council (UCC) that uses industry standards to

synchronize the item information between trading partners. It is a third-party external exchange that provides product registry services to enable synchronization of item and location information to reduce mis-shipment and return, and shorten setup time for new products. It targets high-volume, low margin, inefficient industries such as retail, grocery. While the EPC Network [8] offering dynamic product information specific to an individual item, such as expiration dates and shipping details held on the Radio Frequency Identification (RFID) [9] tag, the UCCnet contains static attributes common to all products.

The solution (UCCnet_ItemSync) integrates the supplier's backend system (SAP) with the UCCNet hub (AS2 Server) for registration and validation of data. The product order information (via Item Data) from the supplier will trigger the registration (new item publication request and registry catalogue item registration) of the product information in the UCCNet catalogue. The rest of the paper will reference this example when we describing features and functionality.

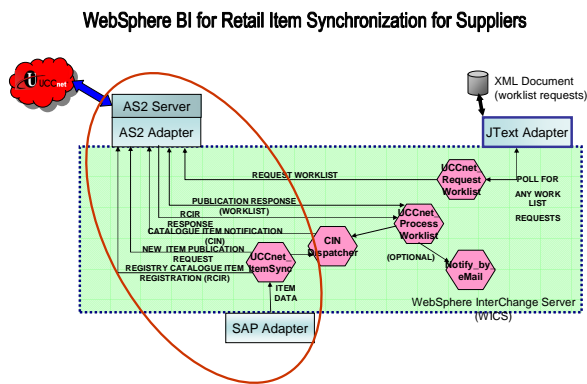


Figure 2 UCCnet Item Synchronization.

In our example, two Connectors (for SAP and AS2 server) and one Collaboration (business process to publish product ordering information) are the Solution Artifacts used in the template. Solution Artifact exposes a web-service like interface. For the Collaboration, a port in the Solution Templates Tool corresponds to the port in its corresponding WBI SE collaboration and an operation corresponds to the supported verb by that port. For a Connector, a port in the Solution Templates Tool corresponds to a supported message formats and an operation corresponds to the supported verb by that message formats. The Template Creator can simply wire them together according to the business logic. To ensure message formats from both artifacts are interchangeable, WBI map editor will be launched to

mediate the conflict between two message formats during wiring.

4. Solution Composition and Transformation

Solution Template is a reusable asset at a higher level of abstraction that can be transformed into a solution (or application) in service oriented architecture environment. We used it for creating e-business integration solutions to reduce the complexity and the cost of creating the integration solution by providing more repeatable, cumulative and transferable knowledge obtained during the life cycle of the integration solution.

4.1. Solution Composition

Solution can be composed by capturing the reusable components (other templates or Solution Artifacts) and define the relationships amongst its components according to the business requirements. Solution Artifacts can be a service container usually representing process flows (Collaborations), adapters (Connectors), screen views, and other elements that can be reused [3]. These components can be created new or imported along with its definition, and its platform requirements and performance characteristics. Components are then wired according to their business logic. Once created, it becomes a reusable business services (templates) that publish or operate on business data. As shown in Figure 3, the UCCnet item synchronization artifacts illustrated in Figure 2 is composed as a Solution Template using the PIM Composer of the proposed Solution Template Tool.

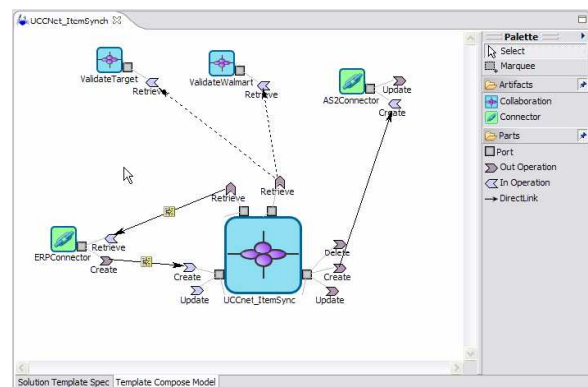


Figure 3 Template Composition View.

To address the customization aspect, the notion of points of variability is introduced. A given artifact may offer customization points or points of variability that identify options for a given artifact. As an example

shown in Figure 4, the Process flow of UCCnet_ItemSync defined in a collaboration artifact could have two points of variability: “New Item Publication Request” and “Registry Catalogue Item Registration”. Similarly, in a connector artifact, inbound/outbound message formats, business rules, functional options, properties, etc could be defined. This tool detects if a variability point is available and presents the Solution Creator with options to choose amongst them, further more would a mismatch occur, this tool offers options to resolve the incompatibility. When appropriate, an interaction is initiated by the tool to move ahead the integration process and minimize any second-guessing by the user.

Variability in a Process Flow Model
(Using Process Flow Artifact as an Example)

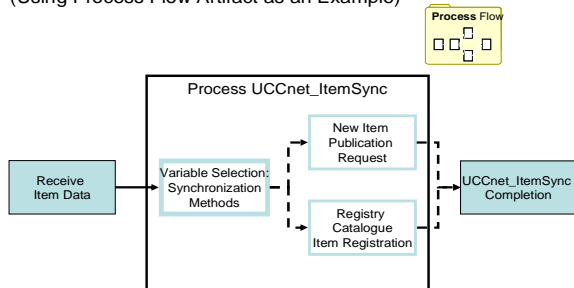


Figure 4 Variability in a Process Flow Model.

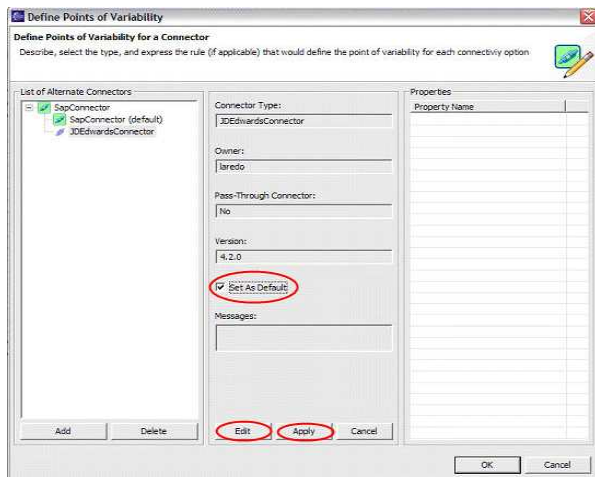


Figure 5 Points of Variability for Connector.

The Template Creator can define the point of variability to be consumed by the Solution Creator. For example, the Template Creator can add the flexibility to this template for other suppliers who want to publish their products into the UCCNet catalogue. Connector supports point of variability by providing

choices of the connectors under the same component, a generic connector container. So more connector implementations can be added for different suppliers under a generic connector container (e.g., ERP connector) as shown in Figure 5, and allow more options (SAP or JDEdwards) later when configuring the template. Only one connector can be selected to be active and the active one is the one considered when doing the composition with other components.

Collaborations offer point of variability for their properties and interfaces. The tool supports three types of conditions to govern the point of variability: mandatory, optional and conditional as illustrated in Figure 6. Some interfaces of the collaboration are mandatory and needs to be presented when the template is configured. Some interfaces are optional and it can be enabled or disabled at configuration time at the discretion of the Solution Creator. Some interfaces will be automatically enabled if the rule associated with the point of variability is met; otherwise, it will be disabled.

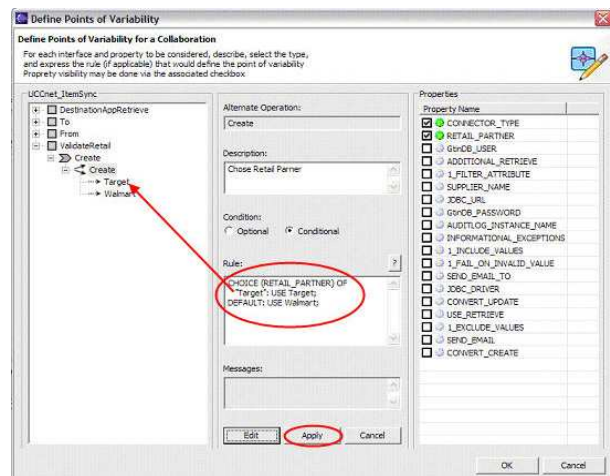


Figure 6 Points of Variability for Collaboration.

In our example, we can define another point of variability for the interfaces of an artifact. The suppliers can have many different business partners to sell their products, and the supplier’s message formats should be compatible with the business partner’s. Therefore the Template Creator can add new collaborations to validate the supplier’s message formats against business partner’s message formats. If the Template Creator can add validation logic for Wal-Mart and Target stores, the Solution Creator can select appropriate business partners by choosing the corresponding interfaces of the business partner. For example, the value of an artifact property such as RETAIL_PARTNER can be used to enable the

collaboration’s operation to Wal-Mart or Target store as an operation of choice as shown in Figure 6.

The rule can also constrain the value of the property and the default value can be set to a literal or a reference to another property. In most cases rules can be used to define the point of variability. A very simple rule set has been defined to create this notion. Associated with the artifact or template properties, these rules can be used to perform configuration to determine the behavior of the solution. Artifact properties, treated as string values, can be used in the rule expression where two types of rules are supported. First one is the Boolean expressions where the operators are $<$, $>$, $!=$, $==$, $||$, $\&\&$, $(,)$, and the second one is the choice selection driven by the value of a property. For example, if $RETAIL_PARTNER == Target$ then the Retrieve operations from “ValidateTarget” collaboration is used; otherwise, the Retrieve operation from “Wal-Mart” collaboration.

4.2. Solution Transformation

Solutions are composed in a platform independent fashion regardless of their ultimate execution environment and become reusable templates to create other solutions. Solutions templates, once created, can be published in the form of a *jar* file to the file system for later reuse, sharing, or upload to asset library.

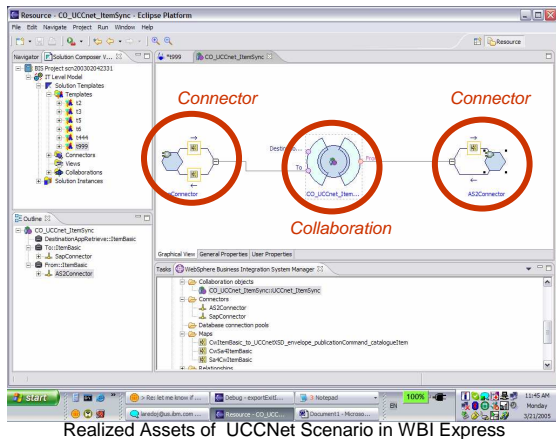


Figure 7 WBI SE Runtime View.

Once the template is configured per the user requirements for certain platform, it becomes a Solution Instance as illustrated in Figure 7 where one Collaboration and two Connectors are instantiated. The Solution Creator needs to register and generate all the necessary elements into a specific platform to ensure that the solution can be executed on that platform. Once the physical topology is determined,

the template can be augmented to provide a deployment layout that matches the solution requirements. This will further enrich the solution with component specific information to fine tune each component in terms of configuration settings and performance parameters. As the template is enriched during deployment, the added information can be provided to the deployed solution management phase and allow the management tool to provide a complete view of the solution. Solution upgrades can reuse this information, expediting the process to deploy the new version. Furthermore, collected information can be applied (or reused) to the different configuration with similar requirements. Template becomes the unifying artifact throughout the life cycle of the integration solution.

In our example, we used WBI SE as our platform and all artifacts (Collaborations and Connectors) and their related assets (Business Objects and Maps) will be registered under a new or existing Integration Component Library (ICL) in WBI SE. The links will determine the Collaboration Objects to be created and necessary compilations will be performed. At this point the ICL is ready to be configured into a User Project and ready for the deployment to test the integration solution.

5. Models Transformation Functions

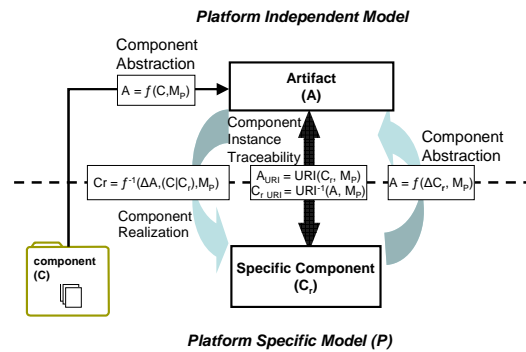


Figure 8 Model Transformation Functions

Our tool seamlessly integrates the Platform Specific Model (PSM) and Platform Independent Model (PIM) workspaces in the same environment, Eclipse-based platform. It provides an integrated working environment for solution composition, solution artifact realization, and run time collaboration execution. The transformation functions between PIM and PSM, shown in Figure 8, consists of three major parts: Component Abstraction, Component Realization, and Component Instance Traceability as described in the following three subsections.

5.1. Component Abstraction

The objective of the Model-Driven approach is to move the focus from programming to solution modeling. A PIM artifact is a high level abstraction of the corresponding components in PSM. To abstract the PSM components into PIM artifact, only the required elements in PSM components are extracted. A transformation function, f , converts PSM component, C , into the corresponding PIM artifact, A , is expressed as $A = f(C, M_P)$, where M_P is the mapping table representing for a particular PSM P . The transformation is based upon the mapping table which defines the relationships between PSM components and PIM artifacts. Ideally, same transformation function can be applied to different PSM with the change of the mapping table.

In a runtime environment as shown in Figure 8, the transformation function is expressed as $A = f(\Delta Cr, M_P)$, where only the changed elements (Δ) of the runtime PSM component (Cr) are converted. Our tool keeps track of instances of components in a runtime environment. This is made possible by the Component Instance Traceability function as described in the next subsection.

5.2. Component Instance Traceability

The Component Instance Traceability function utilizes URI to dynamically trace the component instances in both PIM and PSM. Given a particular PSM component (Cr) and the corresponding mapping table (M_P) of a particular PSM platform, the transformation function (f_{URI}) generates and traces the corresponding URI value (A_{URI}) of the artifact in the PIM space, i.e., $A_{URI} = f_{URI}(Cr, M_P)$.

Reversely, given a PIM artifact (A) and the mapping table (M_P) of a particular PSM platform, the reverse transformation function (f_{URI}^{-1}) generates and traces the corresponding URI value (Cr_{URI}) of a component in that particular PSM space, i.e., $Cr_{URI} = f_{URI}^{-1}(A, M_P)$.

5.3. Component Realization

The Component Realization transformation is expressed as $C_r = f^{-1}(\Delta A, (C|C_r), M_P)$. The transformation function (f^{-1}), a reverse of component abstraction function, dynamically transforms changed elements (Δ) of the PIM artifact into the corresponding components of a particular PSM. Similar to Component Abstraction function, it requires a mapping table (M_P). The different part is that it selects an existing PSM

component if such one exists in the runtime PSM; otherwise, a PSM component template is selected.

6. Conclusions

In this paper, we present a Solution Template Tool for enterprise applications integration to simplify the life cycle of an integration solution, through flexible design and customization of Solution Templates and an interactive environment driven by Wizards. The tool provides levels of abstraction, separation of concerns, and reusable assets to ease the task of solution composition. The WBI Express SE is chosen as our platform specific platform with the model transformation functions introduced. The tool is adapted as a WBI SE Tech Preview, and is available from IBM Developer Works website [6].

We also describe the implementation of UCCnet as a Solution Template, how it is composed as a platform independent model, how the points of variability are defined, and its transformation to the platform specific model for instantiation and deployment in WBI SE.

References

- [1] J. Lee, K. Siau, and S. Hong, "Enterprise Integration with ERP and EAI," *Communication of the ACM*, pp. 54-60, February 2003.
- [2] Joaquin Miller and Jishnu Mukerji, "MDA Guide Version 1.0.1," OMG/2003-06-01. <http://www.omg.org/docs/omg/03-06-01.pdf>
- [3] Ying Huang, Kumar Bhaskaran, and Santhosh Kumar, "Platform-Independent Model Templates for Business Process Integration and Management Solutions," *2003 IEEE International Conference on Information Reuse and Integration (IRI)*, pp. 617-622, Oct. 2003.
- [4] <http://www.ibm.com/developerworks/websphere/downloads/simpleWBISE.html>
- [5] Bill Moore, Knut Inge Buset, Dennis Higgins, Wagner Palmieri, Mads Pedersen, Joerg Wende, Alison Wong, "WebSphere Business Integration Server Express The Express Route to Business Integration," <http://www.redbooks.ibm.com/abstracts/SG246353.html?Open>.
- [6] UCCnet. <http://www.uccnet.org>.
- [7] Lee Gavin, Geert Van de Putte, Suresh Addala, Rajaraman Hariharan, Daniel Max Kestin, Pushkar Suri, Kiran Tatineni, "Implementing WebSphere Business Integration Express for Item Synchronization," <http://www.redbooks.ibm.com/abstracts/SG246083.html?Open>.
- [8] EPCglobal Inc. <http://www.epcglobalinc.org>.
- [9] "A Guide to Understanding RFID," *RFID Journal*, <http://www.rfidjournal.com/article/gettingstarted/>.