# Research on Connection Modeling Based on EAI PLATFORM

Ping Xu
Jianghan University
School of Mathematics & Computer Science
Wuhan, China
xp7812@126.com

Jun Tao
Jianghan University
School of Mathematics & Computer Science
Wuhan, China

***Abstract*-The trend towards heterogeneous environments comes with an increase importance of Enterprise Application Integration (EAI). An integration platform consists of a set of inbound adapters, a core message broker, and a set of outbound adapters. We develop the model-driven EAI Platform Connection approach. A key component of any application integration software is the ability to provide connectivity between various sources of information in a business enterprise. In this paper, we give an overview on the connectivity problem in an integrated application environment. Based on this problem, we give out the EAI Platform Connectivity Solution. Finally, the approach is deal with the model-driven generation of dynamic adapters for integration platform.**

***Keywords -EAI, Connectivity problem, Connection model process, Individual flows***

## I. INTRODUCTION

The trend towards heterogeneous environments comes with an increase in importance of Enterprise Application Integration (EAI). Such an integration platform consists of a set of inbound adapters, a core message broker, and a set of outbound adapters. The large number of supported external system types result in the need for data independence and, simultaneously, for efficient integration task processing. These requirements—but particularly the first one—typically result in very generic inbound and outbound adapter architectures. Therefore, the architecture of these adapters is quite monolithic, resulting in low functional flexibility of the soft-ware components. This means that for each external system type, a single adapter is required, although specific functional modules could be reused.

To tackle this problem, we develop the model-driven EAI Platform Connection approach. A key component of any application integration software is the ability to provide connectivity between various sources of information in a business enterprise. A successfully integrated system must have a dependable, yet flexible, way to connect applications. In addition, a connection must be able to use transactions to ensure the integrity of its data and must be able to move that data to and from a wide variety of sources. EAI Platform uses connection models to provide a solution to this connectivity problem.

A connection model is a set of data flows that define how information is transported from a data source (such as a file or a database) to a data target (such as a data channel). The paper is organized as follows: Section II gives an overview on the connectivity problem in an integrated application environment. Based on this problem, Section III gives out the EAI Platform Connectivity Solution. Then, Section IV includes the core Connection Modeling approach description, starting with an overview of the complete process, followed by explanations of the individual steps as well as examples and the resulting benefits of this approach. Furthermore, we highlight open problems as well as research challenges. Finally conclusion is given in Section V.

## II. THE CONNECTIVITY PROBLEM

Connecting divergent applications seems like a relatively straightforward request. Supposed, for example, your company has a database used by the Human Resource department to keep track of all company employees. Now the accounting department has a different database that it's used to manage the company payroll. This database is part of the application software used by the accounting department and has a different data structure from the database used by the Human Resource department.

Now let's suppose that all salary information for the company is stored in the Human Resources database because all salary increases are approved by the Director of Human Resources. At some point, however, information about a salary increase must be communicated to the Accounting department so that the employee's paycheck can reflect the raise. To solve this problem, you need to create a connection between the Human Resources database and the Accounting database so that data can be moved from one database to another. To create a connection, you must know details about your data at its source and about the data target. For example, before you can connect the two database applications in our example, you need information about the schema of the table in the Human Resources database that contains the information you want to transport. You need similar information about the table in the Accounting database that is going to receive the data.

If you follow the two-phase commit transaction model, you cannot use the simple connection from database to database shown in the previous illustration. Instead, you must have two transactions for each event—something you cannot do with a single connection. The EAI Platform solution to this problem is to use EAI Platform connection models as the middle step in this connection.

Published by Atlantis Press, Paris, France.

## III. THE EAI PLATFORM CONNECTIVITY SOLUTION

### A. *Connection Model Processing Structure*

An EAI Platform connection model provides the interim step required to transport data within the boundaries of a transaction. Data events are pulled from a data source into EAI Platform by way of a connection model. EAI Platform processes and transports the data in the first connection model. It then uses a separate connection model to push data out to the data target.

EAI Platform is responsible for managing all activities by the connection model. This means that, along with the connection model management services, connection models can use some of the other EAI Platform features, including: Integration with the publish/subscribe elements of the EAI Platform environment; Access to transformation services including visual and code-based transformations; Object naming and access control; Testing, troubleshooting, debugging, and monitoring capabilities; Secure, persistent, and recoverable storage for application data objects in the EAI Platform Repository. In reality, connections can be quite complex and can do much more than transport data. For example, you can include multiple transformers to manipulate data inside the connection model, or you can output data from the connection model to more than one target. Connection Model Process structure is described in figure 1.

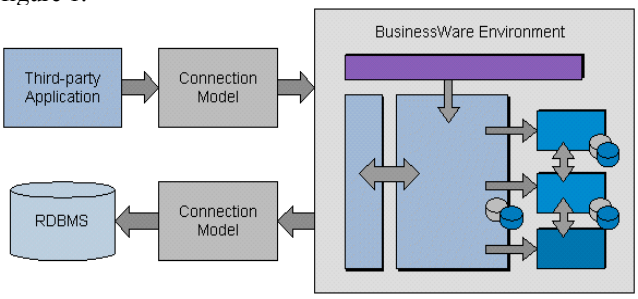


Figure 1. Connection Model Processing Structure

### B. *Inside a Connection Model*

Connection models are made up of data flows that determine what happens to data as it moves from one application to another through EAI Platform. Each flow has one or more specific tasks to accomplish. The task could be as simple as moving data events from an outside source to another flow, or could be as complicated as transforming data into a new format based on the occurrence of other events.

In its simplest form, a connection model does the following things: It can talk to the outside source of your data so it can receive data. It can push data out to a target destination. If an outside source sends a request and waits synchronously for a reply, the connection model can return a reply to the source.

A source connector flow pulls data from an outside source and changes the data to events that can be transported through the connection model. A transformer flow processes data and performs data transformation or translation according to user-defined specifications. A target connector flow pushes the transformed data out of the connection model to a specified target—in the previous figure, the target is a channel. We can use different kinds of input and output connectors, including connectors that take data to and from an EAI Platform channel. Once data are in EAI Platform, you can use another connection model to send it back out through a different channel to a different target such as a file, a third-party application, or another database.

### C. *Connecting to EAI Platform Applications*

We can create connection models to provide connections between two outside applications, but we also use a connection model to send data to EAI Platform for being processed or analyzed. All data that comes into EAI Platform to be used in business process models or real-time analyzers does so through channels. To use EAI Platform effectively within your organization, you need to distinguish between data transformation and a business process. Both are essentially processes where a series of actions are required to produce a result. However, there are important differences. Create connection models with Connection Modeler when you need to do any of the following: Migrate data, Connect legacy systems, Transform data, Route data. Process models let you model business processes. A business process involves all of the following: Validation of a set of events through time that are related to some process, for example, ordering a widget, Judgment of the same set of events through time. The need persists state information and history of the process. Data transformation and simple data migration do not involve any of the above conditions.

## IV. CONNECTION MODELING APPROACH

### A. *Design the connection model*

Ideally, a single connection model should have all of its flows moving in the same direction. That is, a connection model should contain flows going into EAI Platform from a data source, or coming from EAI Platform to a data target, but not both. This also makes the connection model easier to test and easier to troubleshoot if you have problems. When you need to describe flows going in both directions, you should create connection models for the flows in each direction.

Architecturally, there is no limit to the number of flows in a connection model. However, computing resource limits such as virtual memory size do place limits on the number of flows in a connection model. In practice, most connection models contain 10 or fewer flows. The figure 2 shows a connection model that takes different events from the same data source, does different transformations to the different events, and then sends that transformed data out to a variety of data targets.

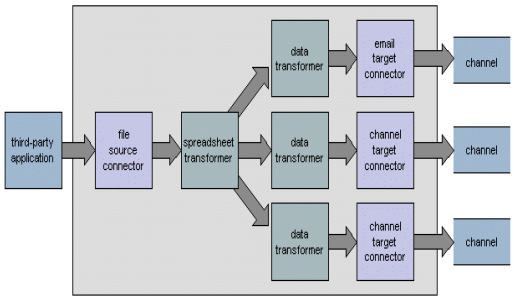


Figure 2. Single-Source Multi-Channel Connection Model

In a real modeling environment, we might require multiple connection models to achieve scalability or to implement separate projects and procedures that use the shared resource. The shared resource can be a data source from which the connection models send events to various EAI Platform channels. A shared resource can also be a target. In this case, multiple connection models send events from various EAI Platform channels to the target resource.

### *B. Build a connection model*

EAI Platform uses connection models to determine how data is to be transported between business applications in an integrated system. Connection models are made up of flows that define what data is received by the connection model, how the data is pushed out to targets, and what happens to the data along the way.

Building a connection model is the process of designating which flows we want to use, wiring those flows together in the order we want them to occur, and then configuring the flows to specify what information is to go through the flows and how the data is processed. Create the connection model item in the navigation pane using a pre-defined template or create our own custom design. Add, remove, or rename flows, and adjust the wiring in the model as needed. When we create a custom connection model you use the Flow Palette to select the flows you want in your connection model. Each icon in the palette represents a flow we can use in a connection model. The Flow Palette shows the flows currently available to use in your connection model, organized according to their primary function. The Flow Palette is described in figure 3.

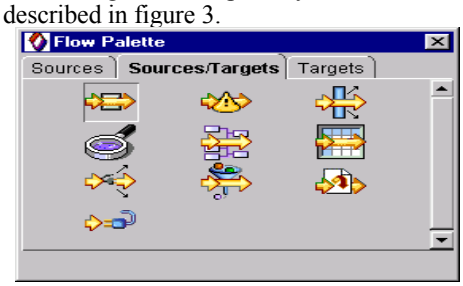


Figure 3. Flow Palette

### *C. Configure connection model and configure each of the individual flows*

For each flow in our connection model, we specify what information, in what format, is coming in or going out of the flow. We also specify what transformations (if any) happen inside the flow. Statistics dialog lets you send statistics to other EAI Platform components. In this dialog, you can turn on and off the ability to send statistics about how many events have been processed by an individual flow. The Statistics dialog is shown as the figure 4.

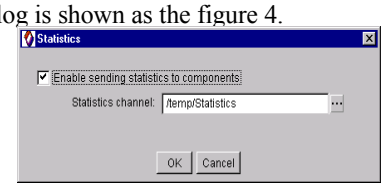


Figure 4. Statistics dialog

Each flow has its own unique set of properties to set. Some flows (such as the File Source connector) have many properties; while other flows (such as the Simple Data Transformer) have only one property, but require other information such as specifying which transformer methods the flow will use.

For some of the flows in our connection model, we must specify which event interfaces the flow should process. EAI Platform uses interfaces to define which information from our data source we want to pull in to our connection model. Many flows are designed to handle only one kind of event interface, so it is not necessary for you to specify an interface. If a flow does not need you to specify an interface, then the Add and Remove buttons on the Input and Output tabs at the bottom of the workspace are dimmed to show that the interface information is read-only. The configuration of the flows describes in figure 5.

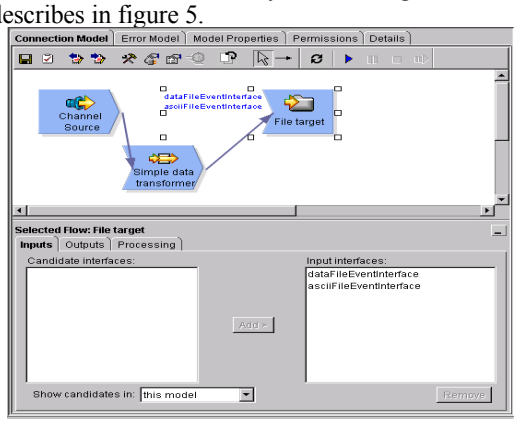


Figure 5. Specifying Event Interfaces for a Flow

## V. CONCLUSION AND FUTURE WORK

The concept of Enterprise Application Integration (EAI) is widely used for integrating heterogeneous applications and systems via message-based communication. Typically, EAI servers provide a huge set of inbound and outbound adapters used for interacting with the external systems and for converting proprietary message formats.

In this paper, we give the detailed connectivity problem characterization, followed by a discussion of the EAI Platform connectivity solution, and we introduce our model-driven connection approach. Finally, the approach is deal with the model-driven generation of dynamic adapters for integration platform.

REFERENCES

[1] Hohpe, G., Woolf, B.: Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions. Addison-Wesley, 2004
[2] Kleppe, A., Warmer, J., Bast, W.: MDA Explained. The Model Driven Architecture: Practice and Promise. Addison-Wesley, 2003
[3] Amelunxen, C., Konigs, A., Rotschke, T., Schurr, A.: Moon: A standard-compliant meta modeling framework with graph transformations. In Rensink, A., Warmer, J., eds.: Model Driven Architecture-Foundations and Applications, 2006
[4] Konigs, A.: Model transformation with triple graph grammars. In: MODELS, 2005
[5] Dorda, C., Heinkel, U., Mitschang, B.: Improving application integration with model-driven engineering. In: ICITM, 2007
[6] Van den Heuvel, W.J., Weigand, H., Hiel, M.: Congurable adapters: the substrate of self-adaptive webservices. In: ICEC.,2007, 127-134