TURNING INFORMATION INTO ACTIONS

From Data to Business Processes through Web Services

Youcef Baghdadi
Department of Computer Science, Sultan Qaboos University, Po Box 36 Al Khod 123, Muscat, Oman

Keywords: Business Model - IS - Factual Dependency - Web Services - Business Process - B2B Integration.

Abstract: Sharing Web services across the enterprise and to support B2B integration becomes more intensive and critical for businesses. This paper proposes a process to generate Web services from the attributes describing the business objects and the coordination artefacts as described in the highest abstraction level of a business model i.e. the universe of discourse where the elements are unique. The process is based on a new concept we introduce and call factual dependency. Factual dependency is a mechanism used to aggregate attributes that are concerned by the same DB CRUD operations with respect to the time and the space. Factual dependencies are then validated with regard to the possible business events to keep only the relevant ones. Each distinct and valid factual dependency is specified in terms of input/output parameters to generate a lowest level of granularity Web services. These Web services are then registered to be discovered and (re)used at request by business processes in their reengineering or composition.

1 INTRODUCTION

The information system (IS) is a representation of the business. This representation is used to control the business. It mainly consists of data, applications and middleware.

Nowadays, there exit various models to construct data, and tools to build and integrate applications. However, we are still facing issues such as: (1) What is a business process or exactly what level of granularity we consider when we define a business process? (2) Which business processes are implemented by the applications? (3) Why do we need to integrate applications and which types of middleware are used? (4) How do we proceed to integrate? (5) What are the business objects represented by data? (6) How business objects are related to business processes and vice-versa?

These issues are resulting from a misunderstanding of the model (if any) used to abstract the business, namely the abstraction levels and the relationships/interfaces between them (Bubenko, 1994). Firstly higher and lower abstract levels are described with different languages. Secondly, the lower levels are more complex than the higher ones as they contain more description details. For instance, a customer may be a physical person or an organization at the higher abstraction level (universe of discourse), and various heterogeneous representations in a lower level (IS) such as different DB tables, flat files, or XML.

Accordingly, we propose a process to derive business processes by composition of Web services from only the knowledge a business has on its perpetual elements at the highest level. These elements are: (i) the business objects such as products, parts, accounts, or partners/suppliers; and (ii) the states of the coordination artefacts used by the business processes. Indeed, these elements, with simple semantics, are sound. That is, the knowledge in terms of attributes we have on these elements are sufficient to determine the business events they undergo or trigger. Each business event involves many operations. The chain of operations forms a business process. We are interested only on database CRUD operations to allow a lowest level of granularity we can master without any ambiguity. Each distinct and specified operation in terms of input/output parameters generates a lowest level of granularity Web services. The Web services are registered to be further discovered and (re)used at request by business processes. Accordingly, we can dynamically compose business processes at request by using the registered Web services.

This process is based on the concept of factual dependency we define as: ‘an attribute Y is factually dependent on an attribute X if they are concerned by the same CRUD operation’. A factual dependency

Baghdadi Y. (2004).
TURNING INFORMATION INTO ACTIONS - From Data to Business Processes through Web Services.
In *Proceedings of the Sixth International Conference on Enterprise Information Systems*, pages 570-574
DOI: 10.5220/0002631005700574

between attributes, describing the elements of the universe of discourse, allows us an aggregation of the attributes concerned by the same business event.

2 BUSINESS MODEL

A business is an open system that seeks some goals or responds to events. It consists of: (1) Business Events, Input, and Output, (2) Production System, (3) Logistic System, (4) Partners, (5) Business Management/Control System, and (6) IS. These components may be classified as components of the universe of discourse, components of the enterprise IS or components of the business management system.

Business events, input/output, production system, logistic system and partners are parts of the universe of discourse. The universe of discourse contains four types of elements: (1) Perpetual tangible as well as intangible elements. We call them business objects (BO). (2) The business events (BE) which are the elements characterized by the space, the time and the effect they have on the business objects or the coordination artefacts. (3) The business processes (BP) that transform business event/input into output are the decompositions of the value chain. (4) The Coordination artefacts (CA) used to coordinate and interface the BP.

BO, BE, BP and CA are differently represented in the enterprise IS.

The enterprise IS is a technology-based representation of the elements of the universe of discourse. This representation consists of data, applications and middleware. Hence, it should contain only one representation of each element as defined in the universe of discourse. However, the actual enterprise IS contains different representations of the same element. This is due to our different intuitions and perceptions of the reality, different languages we use, and the variety of technologies. For instance, a customer may be perceived as an account or a partner. A track of this perception is kept in different DB tables, files or XML. Similarly, a BP is differently implemented (e.g. application, components, objects or manual). This representation breaking requires integration for multiple reasons namely: (i) the need of reconstructing the entire representation and (ii) some BP involve more than one partner.

It is clear that the elements of the universe of discourses are unique. That is, we have only the original there is no clone. If we damage or lose an element, we cannot reconstitute it. Whereas, the elements of the enterprise IS have various representations. That is, we may have various technology-based copies of the same element. If we damage or lose the copy, we can reconstitute it.

Integrating elements in the enterprise IS is more complex; and mostly influenced by IT rather than business perspectives. This is especially more evident in the case of e-business, where new BP are innovated, reengineered, or completely built from scratch to respond to specific BE.

We propose a method to dynamically compose or reengineer BP by using the registered . It is based on the concept of factual dependency, which we detail in the next two sections.

3 FACTUAL DEPENEDENCY

Our approach to turn data into BP through Web services is based on the concept of factual dependency between attributes of the BO or CA as they are described in the universe of discourse.

3.1 Definition of Factual Dependency

A factual dependency is a dynamic-oriented constraint between two attributes X and Y describing the same element or distinct elements of the universe of discourse. The constraint stipulates that two values *x* of X and *y* of Y are inserted/update/deleted/retrieved when a BE occurs.

An attribute **Y** is factually dependent on an attribute **X** if the attributes **X** and **Y** are concerned with the same CRUD operation. A factual dependency is denoted X ➔ Y.

The concept of factually dependency allows an aggregation of the attributes describing the elements of the universe of discourse with respect to the BE they undergo. That is, the attributes having the values inserted, deleted, updated or retrieved by the same BE are grouped together.

In a nutshell, each combination of attributes may lead to an operation, which is particularly true for retrieve operation where each possible project operation (in the sense of the relational algebra). Therefore, criteria to select relevant combinations are required. These are the relevant BE.

3.2 Rules for Factual Dependencies

The concept of factual dependency is different from the well-known concept of functional dependency used in the relational schema design (Codd, 1990). Therefore, the inference rules of the functional dependency are not applicable for the factual dependency, except the reflexive rule. However, factual dependencies respect certain rules, which are:

- Rule 1: Reflexivity
 X ➔ X
- Rule 2: Non-Augmentation
 X ➔ Y does not automatically imply XZ ➔ YZ
- Rule 3: Non-Transitivity
 X ➔ Y and Y ➔ Z does not imply X ➔ Z
- Rule 4: Commutative
 X ➔ Y then Y ➔ X.

If the attributes X and Y are concerned by a CRUD operation o1 then Y and X are concerned by the same operation o1.

- Rule 5: Business events generation
 X ➔ Y may lead to more than one BE.

Assume that the BO customer is described by the attributes: name, address, balance, and mode of payment. The distinct aggregations are as shown in Table 1.

FD1: {Name, Address, Balance, Mode} is used for new customer.

FD2: {Name, Balance} is used in two kinds of operations: update and retrieve. For the update operations, it is used when the BE ‘customer order’ occurs. It is also used when the BE ‘customer pays’. For the retrieve operation, it is used to inquiry the balance in order to trigger a BE. The rule 5 that stipulates that a factual dependency may generate more than one BE is applied.

FD3: {Name, Address} is used to update the customer address when the BE ‘customer changes address’ occurs.

FD4: {Name, Mode} is used to update the mode of payment in different BE occurrence.

That is, we may have the same aggregation concerned by many CRUD operations. Each operation correspond to a distinct BE.

Theoretically, each combination of attributes is a factual dependency that leads to an operation. However, not all the factual dependencies are relevant. We will consider only those corresponding to the actual pertinent BE. A CRUD operation is not applied if there are no BE occurring in the universe of discourse. We update (create, modify or delete) the states of the elements when BE occur. Similarly, we attempt to retrieve information about BO or CA in order to trigger BE or to assist us in performing operations. However, there is a myriad of BE that occur. We consider only those BE which occurrence has an effect on the BO or CA. Hence, the relevant BE are deductible from the factual dependencies.

BO and CA are defined by attributes and BE they undergo or trigger in a similar way than the objects of the object-oriented paradigm are described by attributes and operations. To validate the set of generated factual dependencies that will lead to Web services, we confront each factual dependency to an actual BE.

3.3 Web Services Generation

A Web service provides a standard way for any user, through an application, to access BO and CA. Indeed, once we keep track of a representation of BO and CA in a legacy DB, a flat file or XML DB, the Web services can access it via a simple CRUD operation (e.g. stored procedure in the case of the relational database). A Web services in turn is accessed by any application (e.g. Java application running on a Web server) that presents information to end-user.

The specification of the Web services can be generated from a set of valid factual dependencies. Indeed, rule 5 stipulates that each factual dependency may generate 1:N Web services (Fig1). To keep a lowest level of granularity, each factual dependency leads to 1:N operations depending on the number of BE related to this factual dependency. Each operation will require input/output parameters.

4 TURNING DATA INTO BP

The process of turning data into BP consists of:

Part 1: Turning data into Web ser*vices*

This part consists of the following steps:

Step 1: Selection of BO and CA

Step 2: Description of BO and CA

This consists of defining, in terms of attributes, what knowledge we need to use about BO and CA.

Step 3: Exhaustive list of factual dependencies related to the set of attributes describing BO and CA:

(i) Generate the factual dependencies by combining the attributes.

(ii) Deduce the relevant factual dependencies by confronting them to the business events as they may occur in the universe of discourse (table 2).

(iii) Specify each factual dependency as a CRUD operation with a focus on the input/output parameters.

(iv) Implement, in the IS, the operations related. This implementation may be a program, a stored procedure, a component, an object, Java Bean, etc.

Step 4: Generation of Web services corresponding to these operations.

This step is easy since the operations are specified in term of input/output parameters. We can use a CASE tool or any other tool at this stage to automatically generate the Web services (e.g. www7b.boulder.ibm.com/dmdd/library/tutorials/0308freeze/0308freeze-2-1.html).

Step 5: Registration of the generated Web services.
The resulting Web services are registered in a registry and discovery artefact so that it can be easily discovered.

Part 2: Composition of the Web services onto BP

This part consists of the following steps:
Step 1: Identification of the BE
From the universe of discourse, identify the relevant BE such as ‘customer order’.
Step 2: Description of the flow of the actions corresponding to the BE (e.g. business rules).
Each BE triggers a set of actions with a particular flow. These actions begin with the capture of the event to the production of an output. We can use tools to model the flow (e.g. BPEL4WS).
Step 3: Identification of the actions involving CRUD operations.
The automated actions correspond generally to a set of CRUD operations.
Step 4: Matchmaking between the operations and the registered Web services.
Step 5: Replace in the flow (BPEL4WS) the CRUD operations by their Web services.
Part 2 of the process is triggered by any new BE, a B2B perspective, or when re-engineering the existing BP.

5 RELATED WORK

The problem of integration is addressed in a number of different ways through schema integration of heterogeneous databases (Batini et al. 1986), interoperability (e.g., COBRA, DCOM, RMI, JDBC), e-Services (L. Wong, 2001), to the Web services (Box, 2003; Chen et al., 2003; Kreger, 2001) and their composition (Jablonski, 2003 ; Leyman, 2002) in order to integrate the applications involved in the business processes. Each of them is concerned with a specific aspect of the problem.

In the last three years, the object-oriented paradigm has been extended with the introduction of Service-Oriented-Architecture (SOA) model, which helps to separate business intent from IT implementation. This allows a sharing of business services across the enterprise and to support B2B initiatives. The e-services model is composed of a set of business services, a set of business components, a set of IT elements, and a set of business rules.

The approach described in this paper is in touch with the e-services model, however, it is mainly based on the factual dependency at the highest abstraction of a business to determine Web services that enter in the composition of any business process. The Web services definition and specification are not intuitive to the analyst for less ambiguity.

6 CONCLUSION

This work considers that traditional approaches for integration are more IT-oriented, that is they are proposed from an IT perspective not from a business perspective. They focus on the complex elements of the IS, which makes the integration task harder. Our approach allows generation of de facto standards Web services that facilitate the integration, from the highest abstraction level (universe of discourse) where the elements namely the business objects and the coordination artefacts are easy to capture with less analyst intuition. The Web services are generated from valid factual dependencies regardless of the business processes which will use or reuse them in their composition.

This is a significant issue nowadays where organizations are looking to sharing Web services across the enterprise, to support B2B integration, and to reengineer or compose business processes, which is more and more intensive and critical for a business survival.

We will develop after a global architecture and a supporting tool that allows organization to really turn information into action.

REFERENCES

Batini C., Lenzirini M. and. Navathe S.B. (1986). A Comprehensive Analysis of Methodologies for Database Schemas Integration. *ACM Computing Surveys*, 18(4), 322-364.

Box,B., The Web Services Revolution, http://www.eds.com/thought/thought_leadership_web_revolution.pdf, 2003

Bubenko J., Enterprise Modeling, Ingenierie des Systems d’Information, 2(6) 1994

Chen M., Chen A. K. N and Shao B. B. M, Implications and Impacts of Web services to Electronic Commerce Research and Practices, Journal of e-Commerce, Vol 4. No 4. (2003), pp 128-139

Codd E. , Relational Model for Data Management Version 2. Addison~Wesly, 1990

http://www7b.boulder.ibm.com/dmdd/library/tutorials/0308freeze/0308freeze-2-1.html

Jablonski S. and Petrov I., Web Services, Workflow and Metadata Management as the Means in the Eletrocnic Collaboration Era, ICEIS 2003, Angers/France.

Kreger H., Web services: Conceptual Architecture (WSCA 1.0), IBM Software Group, May 2001

Leyman F., and Roller D., A Quick Overview of BPEL4WS, IBM Developer Work, August 2002

Wong L., e-Services: A Key Component for Success, eAI Journal, March 2001, pp 18-25

Table 1: Factual Dependencies as Aggregation of Customer Attributes

Factual Dependency	***Attribute*** ***CRUD Operation***	Name	Address	Balance	Mode	***Comment***
FD 1	Create	X	X	X	X	Their values are inserted together
FD 2	Update Inquiry	X		X		Balance is updated in distinct occasion and also queried in distinct occasions
FD 3	Update	X	X			Changing the address
FD 4	Update	X			X	Changing the mode of payment

Table 2: Factual Dependencies and Corresponding Business Events

Factual Dependency	***Attribute*** ***CRUD Operation***	Name	Address	Balance	Mode	***Business Event***
FD 1	Create	X	X	X	X	New Customer
FD 2	Update Inquiry	X		X		Order/Payment Balance Inquiry
FD 3	Update	X	X			New address
FD 4	Update	X			X	Change Mode

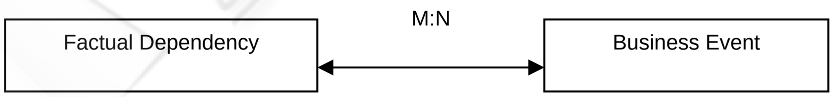

Figure 1: Relationship between Factual Dependencies and Business Events