Using CCTS Modeler Warp 10 to Customize Business Information Interfaces

Applies to:
SAP NetWeaver

Summary
The SDN Article "How to Solve the Business Standards Dilemma - The Context Driven Business Exchange" describes a relative new approach, the "Context Driver Principle" and the SDN article. This mechanism enables semantically unambiguous precision of business information interfaces. Furthermore, the SDN article "Accelerate your Business Data Modeling and Integration Issues by CCTS Modeler Warp 10" mentions that Warp 10 is using the context driver principle for getting customized business information interfaces that most closely fits your unique requirements. This article explains how you can use this feature of Warp 10 for especially shaping a business data interface such as a purchase order or invoice to your business needs in a matter of seconds rather than weeks.

Author: Gunther Stuhec
Company: SAP AG
Created on: 22. November 2007

Author Bio
Gunther Stuhec

Since his master's degree (MSC, 1993) Gunther Stuhec has worked with communications and EDI technologies. As a consultant in a software house for middleware and EDI systems he developed strategic concepts for customers and was responsible for various EDI projects. He joined SAP SI as a consultant in 1999, where he was responsible for implementing XML/EDI projects in conjunction with SAP systems. Since 2001 Mr. Stuhec works for SAP AG as a Standards Architect and has been involved in standardizing business standards on both semantic and syntax levels. Gunther is the chair of the UN/CEFACT Techniques and Methodologies Group (TMG) that is responsible for the development and maintenance of overall methodologies for the development of collaborative business processes and business data on semantic oriented but technical syntax neutral level. He is also the chair of the UN/CEFACT project team that develops the CCTS standard. Furthermore, he is a member of various international and national standardization bodies, such as UN/CEFACT, ISO TC 154, and DIN. He is actively involved in developing standards and serves as an interface between these bodies and SAP, introducing SAP's requirements into their work and incorporating their latest findings into SAP's development activities.
Table of Contents
Overview ....................................................................................................................................................... 3
Current Efforts ............................................................................................................................................... 3
Generating a Schema Using Warp 10 .......................................................................................................... 4
Summary ....................................................................................................................................................... 7
Related Content ............................................................................................................................................. 8
Copyright ....................................................................................................................................................... 9
Overview

In our previous blogs and articles we have promised that CCTS, and more recently our CCTS Modeler Warp 10 tool, will significantly help you to improve the integration of business data interfaces. We have said that Warp 10 will enable a smooth integration, extension and consolidation of SAP GDT based data interfaces with your existing data integration investments, and will result in a radical reduction of your integration efforts going forward. We have tantalized you with an overview of the tool, but not given you any specifics that support our promises. Tantalizing time is over, and we are starting a series of articles and blogs that describe the various aspects of the tool in more detail.

If you an expert in business information exchange mapping, you know that most of the publicly available business data interfaces and B2B message interfaces are typical kitchen-sink like models that usually consider every eventuality for a broad spectrum of users. You typically require several weeks to capture the necessary business intelligence to first understand the interface components and then to strip it down to something that is usable for your specific business needs. In this article, we show you how you can shape a business data interface such as a purchase order to your business needs in a matter of seconds rather than weeks.

Current Efforts

What do you usually do, if you want to process business data by a service or even exchange business data between applications and business applications? First of all you will conceptually define which of the business data should be processed and exchanged in your given context. In other words, you will conceptually describe, how a business data interface should look like for your specific needs. You'll also make sure that the available structures of your system, your business partner or even the other business application will be considered. But the difficulties are starting here.

Let's assume, you are an integration expert and you have to implement a purchase order request interface for a US American car manufacturer and need also the purchase order request for a Canadian seller of consumer products. Naturally, these interfaces are partially similar but will also have differences according to the specific requirements of the industries, roles and countries. But anyway, first of all you will check what you have already in your system and how an external business data interface looks like. Surprisingly, you will get interfaces similar to those shown in Figure 1. Of course, these are overly lengthy, essentially unreadable structures since they reflect the complete (expanded) representation of all possibilities of data to be used. But believe it or not, these are the typical interfaces of a business to business (B2B) purchase order request. The purchase order on the left represents a B2B interface as defined and provided by a standards development organization (SDO) in the high tech industry, and the on the right is a message interface of a typical business application for ordering.

As you can see, a detailed analysis of the information is tedious, time consuming, and exhausting. These interfaces are typically extremely complex, have way too much unnecessary data that no one ever uses, and are typically constructed with a complex hierarchical structure that makes their understanding and use difficult at best. Unfortunately, since you must support multiple users of the source scheme, you must support every possible permutation or combination of the defined data elements. In essence, you must always ask the following questions:

- Which of the hundreds of elements in the source schema actually fit to the needs of the US American car manufacturer or the Canadian seller of consumer products?
- Are the elements really appropriate to the needs?
- Which of the elements should I delete now?
• What happens, if I just delete or add elements?
• How and where, can I delete and add elements?

Due to the complexity of the interfaces, the fundamental differences in the structures of the two representations, and the detailed questions that require in depth analysis, the typical analysis effort will take weeks of dedicated engagement on the part of both data modelers and business experts.

Generating a Schema Using Warp 10

But wait a minute – we said that Warp 10 can significantly reduce the level of effort necessary to generate a schema for your unique business requirements from weeks to minutes. So let’s show you how it is done. First of all, you have to connect to Warp 10 via your preferred web browser (yes – Warp 10 is browser agnostic!). Once connected, the user interface shown in figure 2 appears. In this user interface, you can select the specific message interface you are interested in, such as:

• The "Message Type". In this case purchase order;
• the business process that defines the specific intention of the sender (why he is sending a message to the recipient) such as – propose, accept, reject;
• the action that defines an instruction to the recipient of a message (how the recipient is to process the message) such as – create, change, delete; and
• the activity of the transaction that defines an atomic unit of work in the collaboration scenario – such as request, response, confirm,…

Additionally, you will define the context that is to be used to tailor the purchase order to your specific requirements and uses. The context is defined by the context categories and its context values. The Warp10 prototype currently supports five context categories – country (geopolitical), industry, business process, business process role, and business system (system capabilities). Additional context categories are possible and expected as users become more familiar with Warp 10 and begin to contribute content to the repository. Each context category provides a fixed list of context values, in which you can select zero, one or more values.

• Zero values means that no specific restriction in the context category is required. For example the selected message interface should consider the aspects of all business processes and business systems. This is the equivalent of an “in all contexts” value.
• One value means that only these elements and aggregations should be selected that are required in this single defined context.

• A list of values means that the specific entities could be required in the defined values, respectively.

For example, let's select the values Country = "UNITED STATES", Industry = "Automotive", and Business Role = "Manufacturer of goods". Using these context values Warp 10 will filter and assemble the context specific message interface as shown in Figure 3.

![Figure 3 – Representation of Complete Data Model of Interface](image)

The diagram of message interface representation in Figure 3 follows the ISO 15000-5 Core Components Technical Specification conventions. The diagram will be described in greater detail in one of the next articles. Of particular interest in Figure 3 is the complete representation of the message interface in context we specified. You can drill down into this context by simply clicking on a magnifier icon to obtain the more detailed view of Figure 4.

![Figure 4 – Representation of complete Interface for US Automotive Industry Manufacturing Unit](image)

As you can see, the structure in Figure 4 is significantly smaller than the examples that were shown in Figure 1. Why? Warp 10 has filtered all entities from the “kitchen sink” to only those that are required for
the defined context. Let’s rerun Warp 10 for the same purchase order request, but this time for a Canadian seller of consumer products. Warp 10 will give us the results in Figure 5.

Figure 5 – Representation of Interface in the Context of a Canadian Reseller

The structure in Figure 5 is different than that of Figure 4 to reflect the different needs of the Canadian Reseller for his specific context. For example, the industry “consumer products” will have different properties for describing their line items than the US car manufacturer. From a country perspective, Canada has specific requirements for information due to its different laws. These are not necessarily considered in United States – or vice versa.
Summary

Manually finding the correct use from the myriad possibilities, combinations and contextualization inherent in today’s message structures is extremely difficult. Warp 10 can dramatically reduce the effort required by essentially performing this function for you automatically using its revolutionary context methodology to assemble message interfaces on demand for your specific business purpose.

The context driver principle is a combination of set logic, predicate logic and graph theory. This unique combination enables you to discover the correct structure through the re-use of existing elements available in the common repository. The context driver principle enables the clear categorization of how these entities are really used within given contexts. In fact, it is not just a reuse; it is a classified reuse that precisely defines what is reused and how it is reused across the myriad context possibilities. For example it monitors, tracks, and categorizes through its wiki concept which of the elements of an address are really necessary in the context “United States”, “Germany” or even “China”. Warp 10 makes this information visible to the data modeler and integrator with minimal effort on their part. This increased visibility of reuse of common building blocks in specific context offers significant improvement over current approaches and will contribute to achieving true semantic interoperability within and across organizational boundaries.

Moving forward, it is possible that some of the provided elements from Warp 10 are unfamiliar to you in a given context, or that some elements are missing for your required context. You wonder if CCTS Modeler Warp 10 can help you to effectively add and delete these elements by the support of the context, through the consideration of the existing elements in the repository and without losing the interoperability. The good news is that the answer to this question is an unequivocal yes. Our next article will detail for you exactly how you use Warp 10 to accomplish the tailoring of its initial results.
Related Content

Please read also the following articles and blogs:

- [SAP Network Blog: CCTS Modeler Warp 10 - The Speed of Data Integration and B2B](#)
- [How to Solve the Business Standards Dilemma - The Context Driven Business Exchange](#)
- [How to Solve the Business Standards Dilemma – The CCTS Standards Stack](#)
- [How to Solve the Business Standards Dilemma - CCTS Key Model Concepts](#)
- [How to Solve the Business Standards Dilemma - The CCTS based Core Data Types](#)
- [Getting Started with UN CEFACT XML NDR for CCTS](#)