

DocFlow project's example

The DocFlow project. Current contributors are Aymeric Vincent, Albert Benveniste, Loic H  lou  t, Eric Fabre, Blaise Genest, and Serge Abiteboul

February 25, 2008

Contents

1	Aim of this document	2
2	Verbatim quotes from [1]	3
3	Informal description of the application, from our perspective	5
3.1	Overall architecture	5
3.2	Documents	7
3.2.1	Order form	8
3.2.2	Plant Document	8
3.2.3	Supplier Document (to be done)	8
3.2.4	Dell supervisor Document (to be done)	9

1 Aim of this document

This document introduces the example that we intend to use throughout the DocFlow ANR project. Due to the versatility of the AXML framework, many implementation choices are possible, and this document should serve as a repository for the evaluation and discussion of these choices.

The example we have selected is the Dell supply-chain. It has been considered as a relevant and challenging example by many teams in diverse research areas, ranging from Web-scale data management in computer science to inventory management in business science.

With this application we would like to cover the following aspects of a business process case study:

- Gathering and structuring information through the Web. This is the part of this application that was studied, in the area of data management, in particular by Victor Vianu, where model checking techniques were applied to ensure sanity of the Web interface of this application.
- Moving information through a given set of peers through the Web, and, at the same time, transforming this information. In this case, an order form is stepwise transformed into a computer for delivery. Such a piece of information is referred to as a *business artifact* by Nigam and Caswell at IBM [2] and is central in their view of business management systems.
- A set of peers concurs at the system functioning by processing and moving around business artifacts. One key role of these peers is to *desynchronize* the different flows of business artifacts, therefore allowing for their “speculative” processing. For example, as lengthily detailed below, some peers have the duty of anticipating (or forecasting) queries generated by possible future artifacts. By doing so, supply chains can avoid suffering from accumulated latencies.

Our aim with this application is to assess the following regarding AXML:

- Most models used in the business process area are quite rich by distinguishing several notions (the workflow, business artifacts, etc). While these notions are relevant from the application viewpoint, we hope that they can be conveniently expressed on top of the more basic and uniform AXML model. This will allow to keep the richness of the model from an application viewpoint, while reducing the complexity of the actual execution architecture.
- Does AXML offer the needed means to express this application? What do we need to push outside AXML, e.g., into Java, thus losing analysis capability?
- Which aspects of the application can we keep under the scope of *analysable AXML* as recently identified by Blaise Genest and Anca Muscholl?

This report is organized as follows.

The case study is detailed in Section 2 by quoting verbatim an entire section of [1], a report on a study performed under contract for Dell. This is a study focusing on the inventory optimisation aspects of the problem. To achieve this, the authors must first perform a detailed, system level, analysis of the application. We benefit directly from this. While oriented toward quantitative inventory management, this study reveals that optimised inventory calls for sophisticated distributed algorithms managing complex information. This motivates us to demonstrate that AXML can serve as a beneficial infrastructure for deploying such processing.

2 Verbatim quotes from [1]

Dell's supply chain works as follows. After a customer places an order, either by phone or through the Internet on www.dell.com, Dell processes the order through financial evaluation (credit checking) and configuration evaluations (checking the feasibility of a specific technical configuration), which takes two to three days, after which it sends the order to one of its manufacturing plants in Austin, Texas. These plants can build, test, and package the product in about eight hours. The general rule for production is first in, first out, and Dell typically plans to ship all orders no later than five days after receipt. There are, however, some exceptions. For example, Dell may manipulate the schedule when there is a need to replace defective units or when facing large customers with specific service-level agreements (who have nonstandard quoted manufacturing lead times) for their orders.

In most cases, Dell has significantly less time to respond to customers than it takes to transport components from its suppliers to its assembly plants. Many of the suppliers are located in Southeast Asia and their transportation times to Austin range from seven days for air shipments to upwards of 30 days by water and ground. To compensate for long lead times and buffer against demand variability, Dell requires its suppliers to keep inventory on hand in the Austin revolvers (for "revolving" inventory). Revolvers or supplier logistics centers (SLCs) are small warehouses located within a few miles of Dell's assembly plants. Each of the revolvers is shared by several suppliers who pay rents for using their revolver.

Dell does not own the inventory in its revolvers; this inventory is owned by suppliers and charged to Dell indirectly through component pricing. The cost of maintaining inventory in the supply chain is, however, eventually included in the final prices of the computers. Therefore, any reduction in inventory benefits Dell's customers directly by reducing product prices. Low inventories also lead to higher product quality, because Dell detects any quality problems more quickly than it would with high inventories.

Dell wishes to stay ahead of competitors who adopt a direct-sales approach, and it must be able to reduce supplier inventory to gain significant leverage. Although arguably supply-chain costs include all costs incurred from raw parts

to final assembly, Dell concentrates on Dell-specific inventory (that is, parts designed to Dell's specifications or stored in Dellspecific locations, such as its revolvers and assembly plants). Because assembly plants hold inventories for only a few hours, Dell's primary target, and ours in this project, was the inventory in revolvers.

Dell has a special vendor-managed-inventory (VMI) arrangement with its suppliers: suppliers decide how much inventory to order and when to order while Dell sets target inventory levels and records suppliers' deviations from the targets. Dell heuristically chose an inventory target of 10 days supply, and it uses a quarterly supplier scorecard to evaluate how well each supplier does in maintaining this target inventory in the revolver. Dell withdraws inventory from the revolvers as needed, on average every two hours. If the commodity is multisourced (that is, parts from different suppliers are completely interchangeable), Dell can withdraw (pull) those components from any subset of the suppliers. Dell often withdraws components from one supplier for a few days before switching to another. Suppliers decide when to send their goods to their revolvers. In practice, most suppliers deliver to their revolvers on average three times a week.

To help suppliers make good ordering decisions, Dell shares its forecasts with them once per month. These forecasts are generated by Dell's line of business (LOB) marketing department. In addition to product-specific trends, they obviously reflect the seasonality in sales. For home systems, Christmas is the top time of the year. Other high-demand periods include the back-to-school season, the end of the year when the government makes big purchases, and country-specific high seasons for foreign purchases (foreign language keyboards are especially influenced). Dell sales also increase at the ends of quarters (referred to as the hockey stick). After the center of competence (COC) checks a forecast for predicted availability of components, the forecast goes to Dell's commodity teams and becomes the basis for a six-month rolling forecast that they update weekly. The commodity teams make generic forecasts for systems and components and break those forecasts down to a level of the specific parts that need to be ordered. If the forecast is not feasible, the LOB marketing department revises it, although such revisions are very rare. The buyer-planner for each commodity receives an updated rolling forecast weekly; suppliers receive forecasts monthly.

The objectives of our project were to recommend target inventories for the revolvers to minimize inventory-related costs subject to a service-level constraint and to develop a process and tools for identifying and updating target levels for inventories of the items in the revolvers. (Suppliers who make the replenishment decisions attempt to follow Dell's targets and guidelines.) Dell had been setting inventory targets based on empirical data and judgment with no clear reference to any desired service levels. Dell hypothesized that it could reduce revolver inventory markedly by using a more rigorous approach and gaining better visibility of the inventory throughout the supply chain. Once it determined an optimized inventory level, Dell could collaborate with its suppliers to eliminate excess inventory.

Dell emphasized that it wanted to sustain any changes over the long term, which would require integrating them into its informational infrastructure. ValueChain is a program intended to extend Dell's successful direct-sales approach back into the supply chain with the goal of increasing the speed and quality of the information flow between Dell and its supply base. The corresponding Web site `valuechain.dell.com` is an extranet for sharing such information as points of contact, inventory in the supply chain, supply and demand data, component quality metrics, and new part transitions. Dell envisions using this site to exchange with suppliers current data, forecasted data, new product ideas, and other dynamic information that might help it to optimize the flow of information and materials in the supply chain.

By integrating the process and associated tools that we developed with `valuechain.dell.com`, we want to make the tools part of Dell's and its suppliers' procurement-business processes. Dell and its suppliers through ValueChain can share such information as target inventory levels to support collaboration on future improvements.

3 Informal description of the application, from our perspective

3.1 Overall architecture

The overall architecture is shown on Figure 1.

- `order` is the order sent by a customer. It must contain the name and address of the customer, payment data, and a technical description of the ordered machine (CPU, keyboard, display, software, hard disk, memory, etc.). It will be represented by an AXML document.
- `Dell Webstore` denotes the Web site. It is composed of 1/ a web interface, that collects customers orders, ensures with the bank that payment data are correct, and 2/ a command dispatcher that dispatches the order to the plant closest to the customer's home. The dispatching policy is set up by the system supervisor. The web interface is made of a finite number of `html` pages. A next version of this report will include such a description, since the Webstore is the part of the application where states are most important.
- `bank` is the trustworthy third party checking validity of credit card data. We only represent it in its generic form—there are in fact many instances of it.
- `plant` is a plant. It is an *active peer*. Each plant processes upward an order, by disassembling it and forwarding the relevant subdocuments to the relevant revolvers. It processes downward the returning “materialized” order—which consists of a (software or hardware) part of the computer.

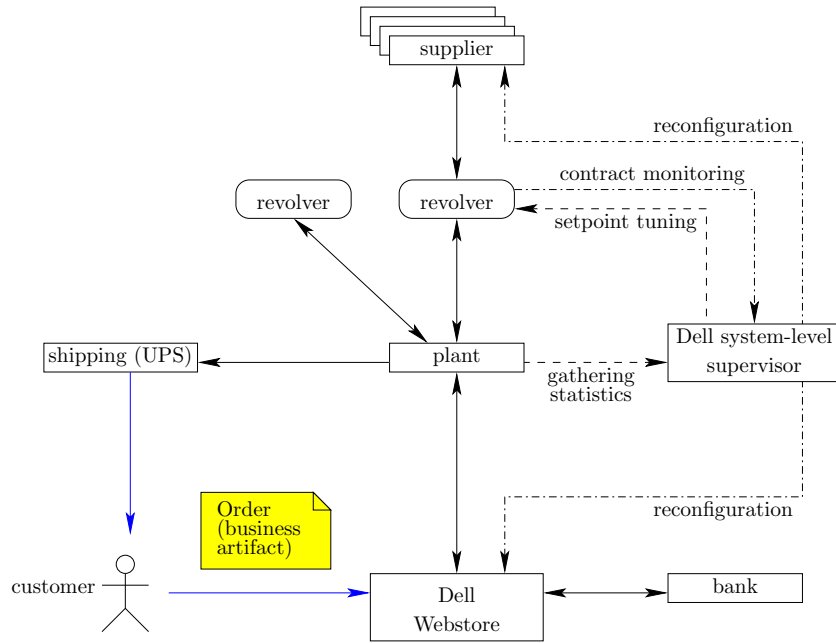


Figure 1: Architecture of the example. Multiple boxes indicate that there exist several instances of the considered peer. Each rectangle symbolizes an AXML peer. A box with rounded corners is a passive AXML peer. The solid arrows indicate the move of business artifacts. Dashed arrows indicate the interactions related to setpoint tuning, by the Dell supervisor. The dash-dotted arrows indicate interactions related to reconfigurations, by the Dell supervisor. Finally, blue arrows and the customer are outside the scope of our modelling work.

We model this materialized document as the original one, with proper tag added to it. Business artifacts handled by the plant are the orders and associated subdocuments, in raw or materialized form.

- revolver is a platform consisting of a set of inventory buffers, one for each part type. The revolver performs *desynchronization* of the flow. It is a *passive peer* that only responds to queries from the plant and from the suppliers it is linked to. Reponse to a query from the plant consists in “delivering” one item of the inventory buffers for the requested part type, and thus decrementing the corresponding buffer level by 1. Response to a query from a supplier consists in incrementing the buffer level of the shipped part type by a certain amount. Each inventory buffer has a level associated with it that we call *setpoint*. A contract between Dell and its suppliers specifies how much the actual buffer level can deviate from its setpoint.
- Shipping is a shipping company. This company receives the assembled machines and send them to the customer that has passed the order. The shipping part is outside the system that is described here.
- supplier is a supplier. It is an *active peer*. Suppliers observe the buffer level for each type of part they supply. They apply whatever policy they want in order to respect their contract with Dell supervisor. We will consider two cases regarding the pool of suppliers interacting with a given revolver:
 - (a) Dell supervisor assigns a setpoint for each buffer for each individual supplier.
 - (b) Dell supervisor assigns a setpoint for each buffer for the entire pool of suppliers, considered as a single contractor; the suppliers team at meeting this global contract using whatever strategy they want.
- Dell supervisor is the Dell supervisor of the entire system. It is an *active peer*. The supervisor is responsible for the following:
 - Gathering sales statistics for proper tuning of the setpoints at each buffer.
 - Monitoring contract violation with suppliers, for possible reconfiguration of the suppliers’ pool.

3.2 Documents

AXML documents are attached to the different peers. In addition, each order is regarded as a *business artifact*, also represented by a document.

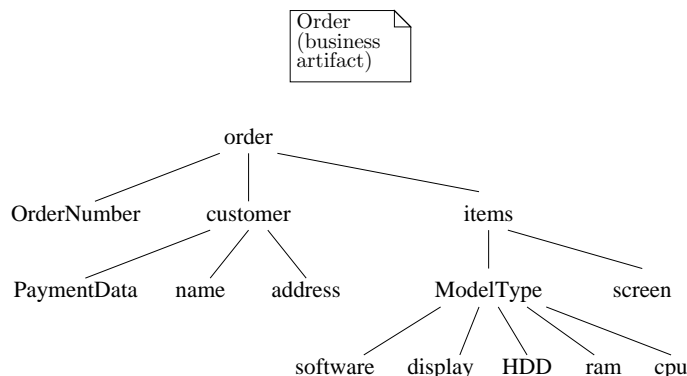


Figure 2: The order form as an XML document

3.2.1 Order form

Figure 2 shows the structure of an order document. As indicated before, this document contains information about the buyer, his name and address, and information on the hardware and software bought. Note that this XML document is not active.

3.2.2 Plant Document

Figure 3 describes the AXML document associated to each plant. This document is a collection of active order forms (that is order documents that have been augmented with service calls, shown in blue).

Let us consider an example of service that occurs in the document of Figure 3, namely *!display@revolver*. *!display@revolver* is a service located on peer *revolver*, with parameter “15”. Its role is to decrement the inventory level of type *display* “15” in *revolver*, and to return the needed part. All nodes in blue represent calls to services located either on the plant or on other peers. All nodes in red are roots of subtrees expanded, from this root, by applying the same procedure as for node “display”.

When an order is passed by a customer, a passive order document is output by the Webstore and sent to the appropriate plant. This passive document is enhanced with service calls (compare Figure 3 with Figure 2) and shipping information (using information on shipping companies and revolvers related to a given plant), and appended to the plant’s tree. The machine is considered as assembled when service *assemble@plant* has tag “Done”. Note that the enhancement process turns the passive order documents into active ones.

3.2.3 Supplier Document (to be done)

The supplier task is to maintain the buffer level so that the contract is met. We may consider simple strategies or more sophisticated team strategies from

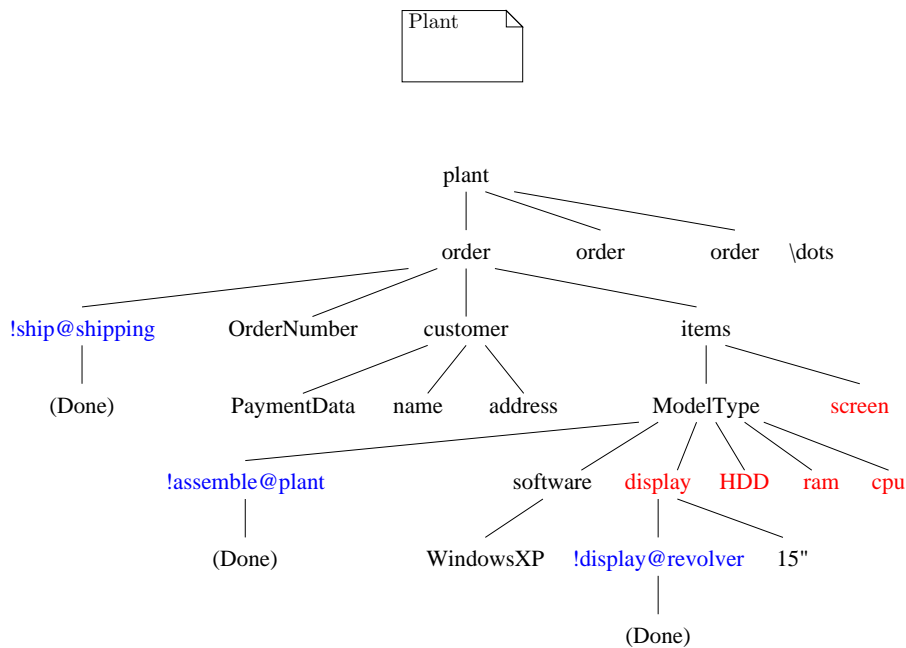


Figure 3: The Plant's document

games. This is left for future refinements.

3.2.4 Dell supervisor Document (to be done)

The Dell supervisor performs the following types of tasks:

- Setting the contracts with the suppliers. These contracts specify how the actual inventory levels can deviate from the given setpoints. This is an off-line activity.
- Gathering statistics regarding sales and tuning the setpoints accordingly. This involves statistical one-month ahead predictions (or so) and is precisely the subject of study [1]. For our purpose, we only need to capture the shape or structure of this tuning policy.
- Monitoring contracts for possible violation and then performing supplier reconfiguration accordingly, or applying penalties.

References

- [1] Roman Kapuscinski, Rachel Q. Zhang, Paul Carbonneau, Robert Moore, and Bill Reeves. Inventory decisions in Dell's supply chain. *Interfaces*,

34(3):191–205, 2004.

- [2] Anil Nigam and Nathan S. Caswell. Business artifacts: An approach to operational specification. *IBM Systems Journal*, 42(3):428–445, 2003.