

Dell Supply Chain application in Active XML 2.1.4

Pierre Bourhis



INRIA Saclay - Ile-de-France,
Université de Paris Sud 11,
ENS Cachan

DocFlow Meeting

Overview

1 Application's Overview

2 Demonstration

3 Implementation

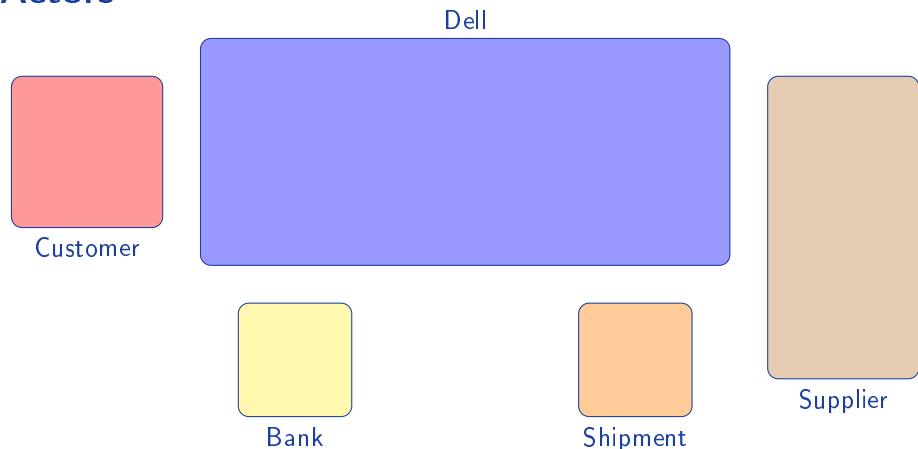
Plan

1 Application's Overview

2 Demonstration

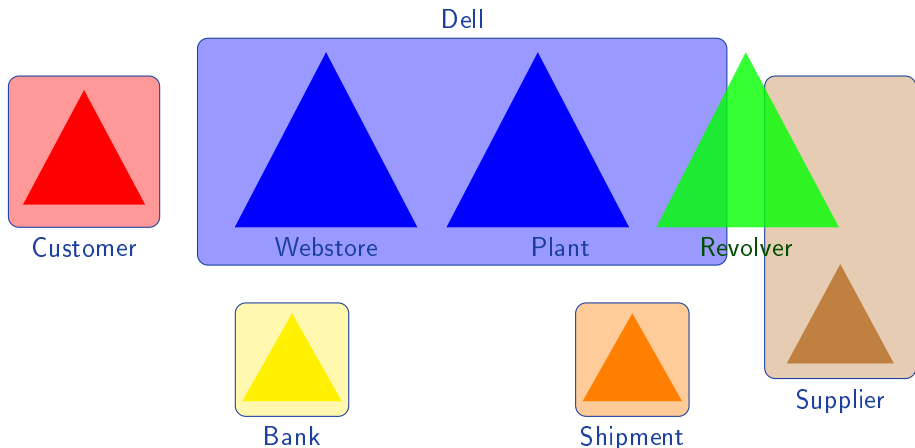
3 Implementation

Actors



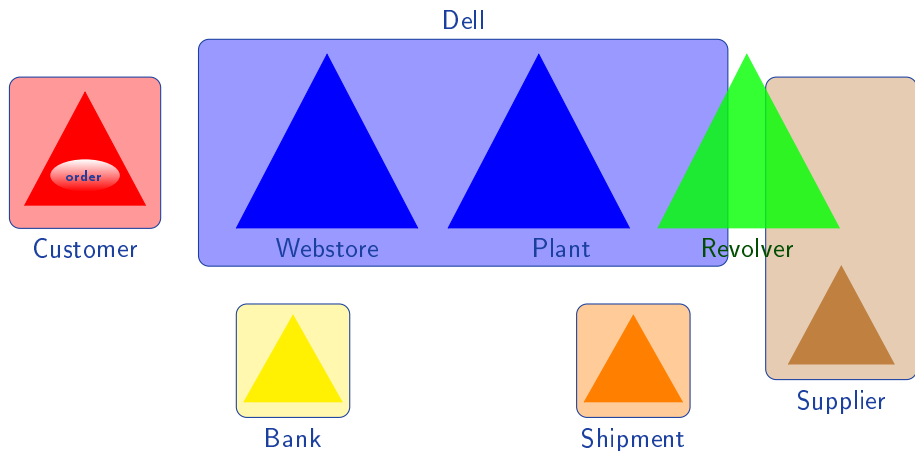
A **Customer** orders a computer to **Dell**. Dell checks from the **Bank** if the Customer is credit-worthy from the **Bank**. The computer will be built from items sent by the **supplier**. After the assemblage, the computer will be sent to the Customer by the **shipper**.

Peers



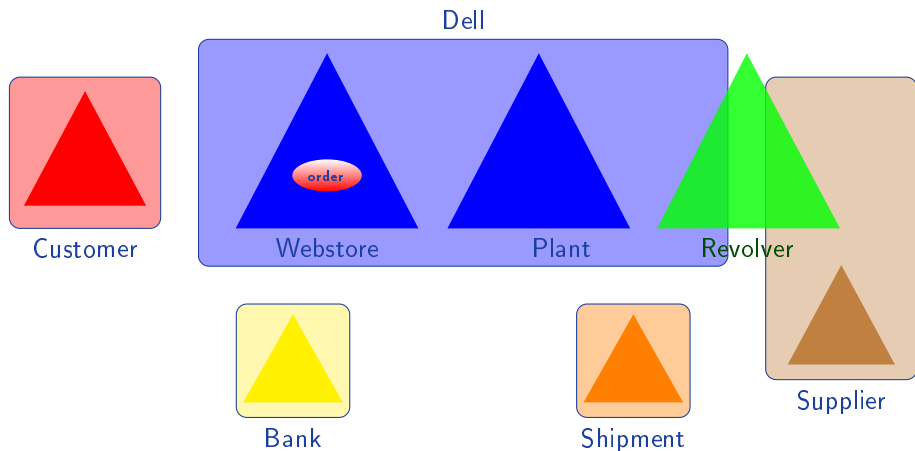
The **revolver** is a buffer between Dell and Supplier. The items used for assemblage computers are stored in the revolver.

Scenario(1/6)



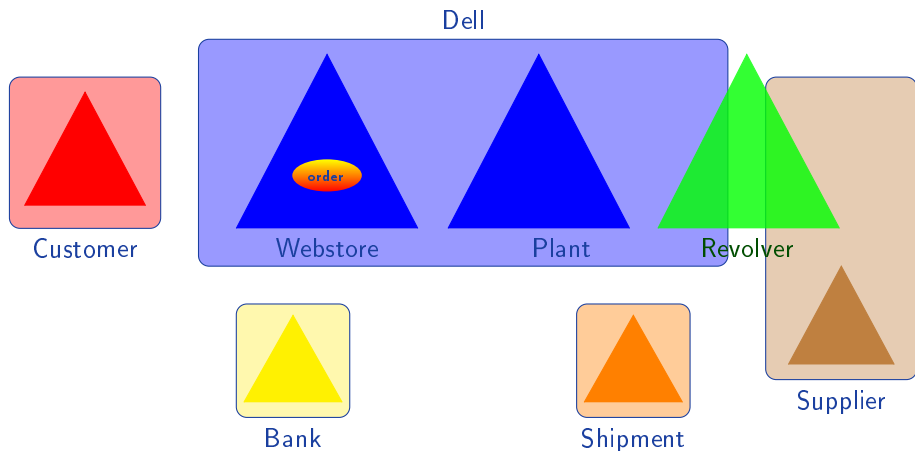
An **order** is built by the Customer.

Scenario(2/6)



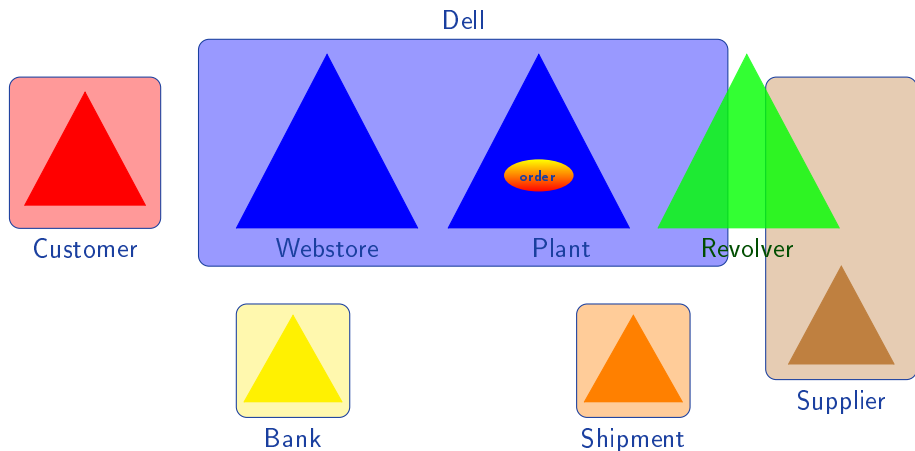
The order is sent to Dell

Scenario(3/6)



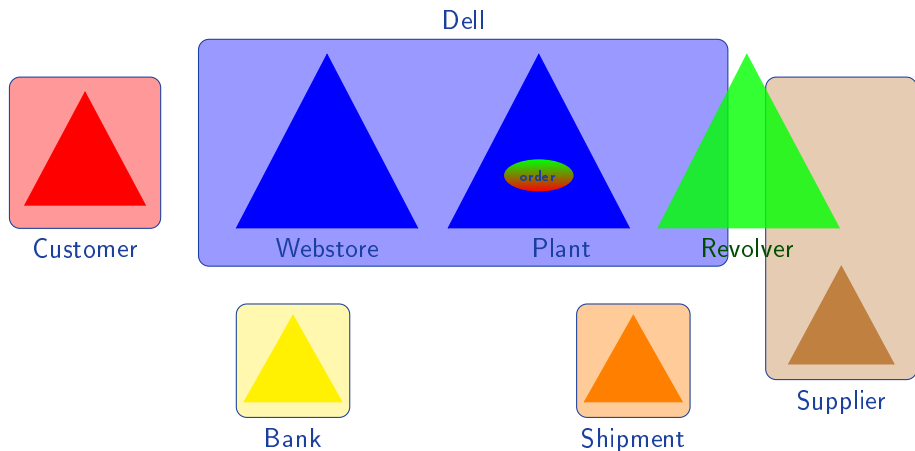
The order can be enriched by others peers: Bank

Scenario(4/6)



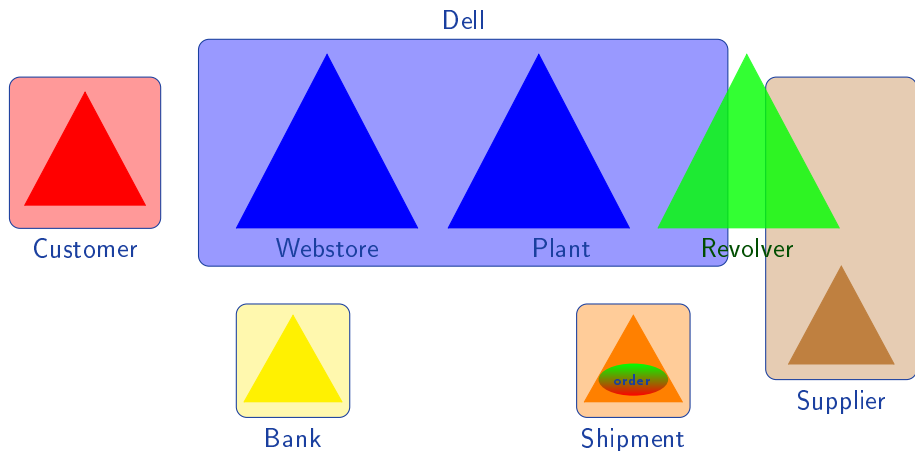
The order is propagated in the application

Scenario(5/6)



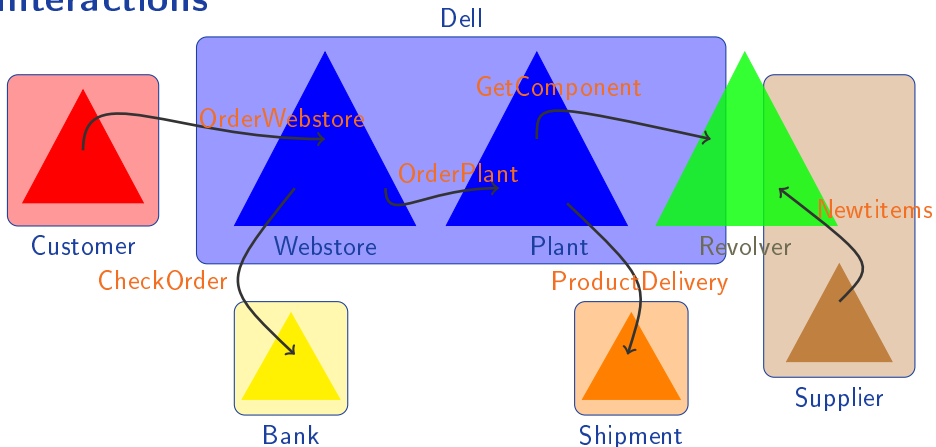
The computer of the order is built from items received from the revolver

Scenario(6/6)



The order is ready to be sent to the Customer

Interactions

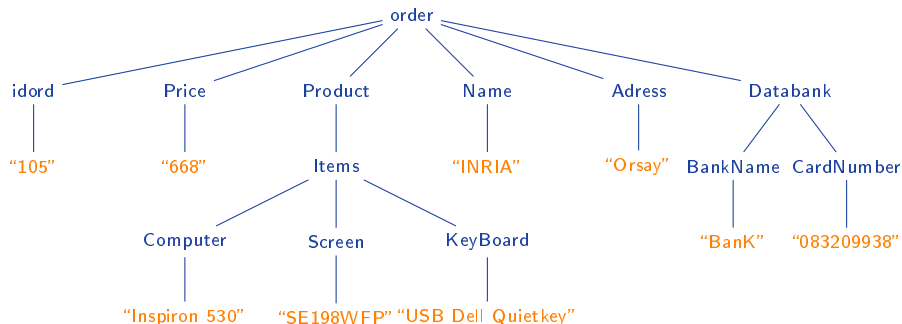


The interactions between peers are done by web service calls:

- To propagate the orders: **OrdersWebstore**, **OrdersPlant**, **ProductDelivery**
- To enrich the orders:
 - ▶ By checking if the Customer is credit-worthy: **CheckOrder**
 - ▶ By adding nested items: **GetComponent**
- To manage the items: **GetComponent**, **Newitems**

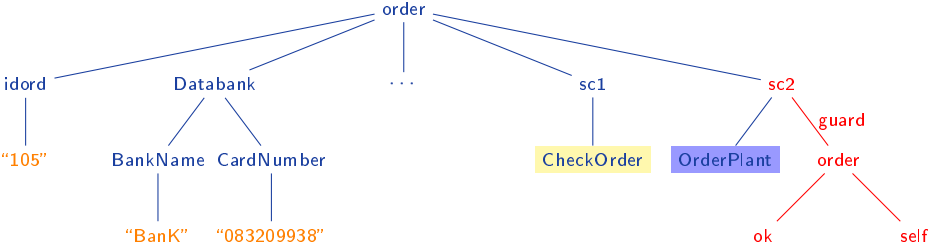
The order

At the beginning:



The order:

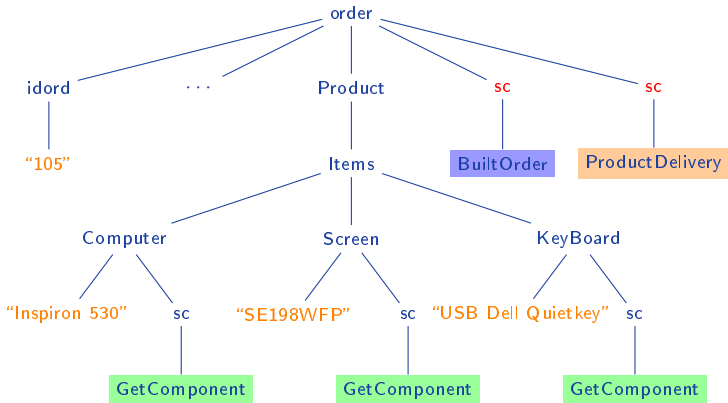
At Webstore:



CheckOrder can send messages `ok` or `nok` following if the Customer is credit-worthy or not.

The order:

At Plant:



GetComponent sends a message **processed**.

BuiltOrder sends a message **processed**.

Plan

1 Application's Overview

2 Demonstration

3 Implementation

Plan

1 Application's Overview

2 Demonstration

3 Implementation

Services of peer

The services of peers are public queries (Xquery + Xupdate) described in the documents. Their invocations are made by calling the service `Applifun(docid,name of query, params)`.

An adaption for `GetComponent`:

Definition

`GetComponent` gets the asked item. But, if this item is not in the revolver, the revolver waits to answer until he finds one.

- 1 The items asked by calls of `GetComponent` are added to asked items.
- 2 A periodic query finds the matches between asked items and items which are in the revolver.
- 3 When an asked item and item in the revolver are matched together, a message is sent to Plant by invoking a specific service of Plant, `Additem`.

Guarded Calls

Remark

Each call is guarded at least by the document where the order is. It can be called only in one document.

Transformation :

Guarded calls for each order → Periodic queries defined in each document

Each time, a query finds a result for a specific order, a service is called for this order and the result is added to this order using another query.

The method of Axml services used for implementing guarded calls are *continousCall* and *continousQuery*.

Conclusion

- An implementation of Dell Supply chain in Axml 2.1.4 .
- The implementation uses more global periodic queries than services.
- A first language for describing external services based on Xquery+Xupdate.

Merci

Des questions ?