1 CoCoME - The Common Component Modeling Example

Sebastian Herold[1], Holger Klus[1], Yannick Welsch[1], Andreas Rausch[1], Ralf Reussner[2], Klaus Krogmann[2], Heiko Koziolek[2], Raffaela Mirandola[3], Benjamin Hummel[4], Michael Meisinger[4], Christian Pfaller[4],

[1] TU Clausthal, Germany
[2] Universitt Karlsruhe, Germany
[3] Politecnico di Milano, Italy
[4] Technische Universität München, Germany

1.1 Introduction and System Overview

The example of use which was chosen as the Common Component Modeling Example (CoCoME) and on which the several methods presented in this book should be applied was designed according to the example described by Larman in [1]. The description of this example and its use cases in the current chapter shall be considered under the assumption that this information was delivered by a business company as it could be in the reality. Therefore the specified requirements are potentially incomplete or imprecise.

The mentioned example describes a *Trading System* as it can be observed in a supermarket handling sales. This includes the processes at a single *Cash Desk* like scanning products using a *Bar Code Scanner* or paying by credit card or cash as well as administrative tasks like ordering of running out products or generating reports. The following section gives a brief overview of such a Trading System and its hardware parts. Its required use cases and software architecture are described later in this chapter.

The Cash Desk is the place where the *Cashier* scans the goods the *Customer* wants to buy and where the paying (either by credit card or cash) is executed. Furthermore it is possible to switch into an express checkout mode which allows only Costumer with a few goods and also only cash payment to speed up the clearing. To manage the processes at a Cash Desk a lot of hardware devices are necessary (compare figure 1).

Using the *Cash Box* which is available at each Cash Desk a sale is started and finished. Also the cash payment is handled by the Cash Box. To manage payments by credit card a *Card Reader* is used. In order to identify all goods the Customer wants to buy the Cashier uses the *Bar Code Scanner*. At the end of the paying process a bill is produced using a *Printer*. Each Cash Desk is also equipped with a *Light Display* to let the Costumer know if this Cash Desk is in the express checkout mode or not. The central unit of each Cash Desk is the *Cash Desk PC* which wires all other components with each other. Also the software which is responsible for handling the sale process and amongst others for the communication with the *Bank* is running on that machine.

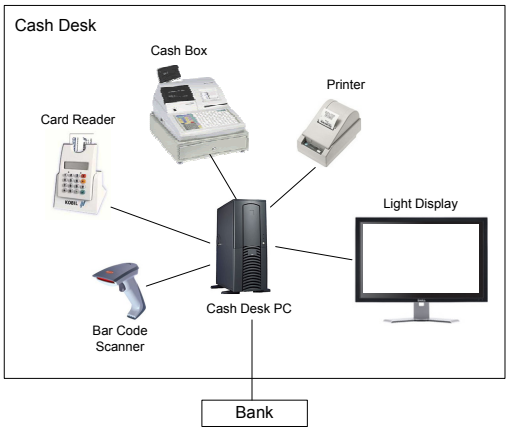

Fig. 1. The hardware components of a single Cash Desk.

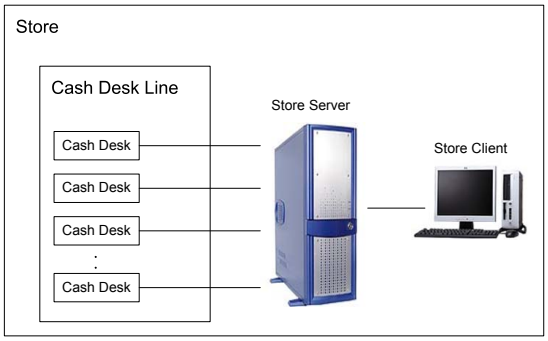

Fig. 2. An overview of entities in a store which are relevant for the Trading System.

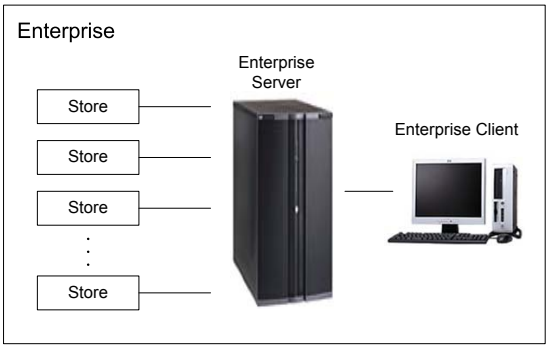

Fig. 3. The enterprise consists of several stores, an enterprise server and an enterprise client.

A *Store* itself contains of several Cash Desks whereas the set of Cash Desks is called *Cash Desk Line* here. The Cash Desk Line is connected to a *Store Server* which itself is also connected to a *Store Client* (compare figure 2). The Store Client can be used by the manager of the Store to view reports, order products or to chance the sales prices of goods. The Store Server also holds the *Inventory* of the corresponding Store.

A set of Stores is organized in an *Enterprise* where an *Enterprise Server* exists to which all Stores are connected (compare figure 3). With the assistance of an *Enterprise Client* the Enterprise Manager is able to generate several kinds of reports.

1.2 Functional Requirements and Use Case Analysis

In this section the considered use cases of the Trading System are introduced which are depicted in figure 4 with the involved actors. Each use case is described using a uniform template which includes a brief description of the use case itself, the standard process flow and its alternatives. Moreover, information like preconditions, postconditions and the trigger of the use cases are given. In the description of the use cases the codes in the squared brackets refer to extra-functional properties in section 1.3.

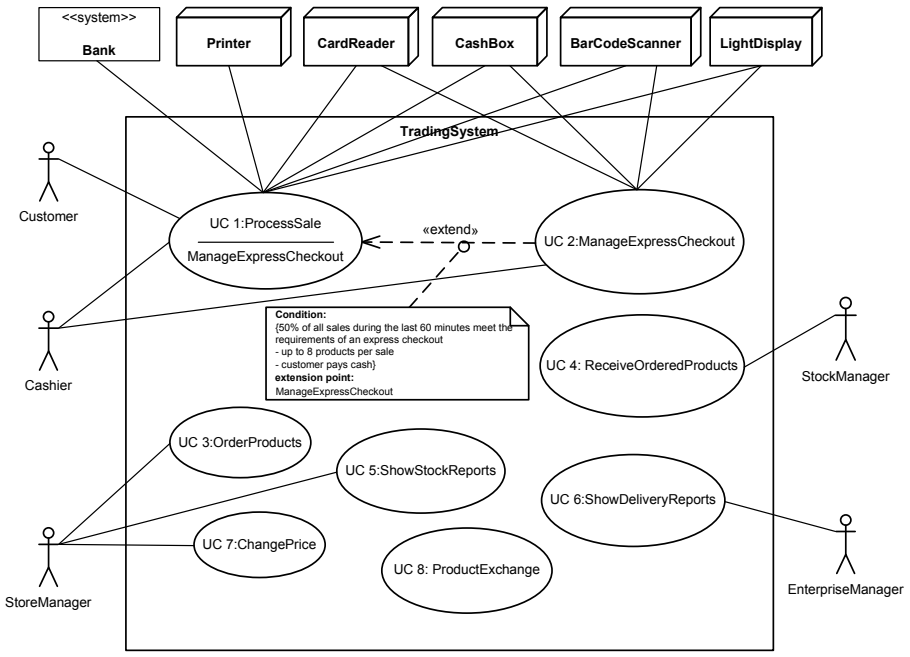

Fig. 4. An overview of all considered use cases of the Trading System.

UC 1 - Process Sale

Brief Description At the Cash Desk the products a Customer wants to buy are detected and the payment - either by credit card or cash - is performed.

Involved Actors Customer, Cashier, Bank, Printer, Card Reader, Cash Box, Bar Code Scanner, Light Display

Precondition The Cash Desk and the Cashier are ready to start a new sale.

Trigger Coming to the Cash Desk a Costumer wants to pay his chosen product items.

Postcondition The Customer has paid, has received the bill and the sale is registered in the Inventory.

Standard Process

1. The Customer arrives at the Cash Desk with goods to purchase. [arr1]
2. The Cashier starts a new sale by pressing the button *Start New Sale* at the Cash Box. [t12-1]
3. The Cashier enters the item identifier. This can be done manually by using the keyboard of the Cash Box [p13-1, t13-1] or by using the Bar Code Scanner [p13-2, t13-2].
4. Using the item identifier the System presents the corresponding product description, price, and running total. [t14-1]
   The steps 3-4 are repeated until all items are registered. [n11-2]
5. Denoting the end of entering items the Cashier presses the button *Sale Finished* at the Cash Box. [t15-1]
   (a) To initiate cash payment the Cashier presses the button *Cash Payment* at the Cash Box. [p15-1,t15a-1]
      i. The Customer hands over the money for payment. [t15a1-1]
      ii. The Cashier enters the received cash using the Cash Box and confirms this by pressing Enter. [t15a2-1]
      iii. The Cash Box opens. [t15a3-1]
      iv. The received money and the change amount are displayed [t15a4-1], and the Cashier hands over the change. [t15a4-2]
      v. The Cashier closes the Cash Box. [t15a5-1]
   (b) In order to initiate card payment the Cashier presses the button *Card Payment* at the Cash Box. [p15-2, t15b-1]
      i. The Cashier receives the credit card from the Customer [t15b1-1] and pulls it through the Card Reader. [t15b1-2]
      ii. The Customer enters his PIN using the keyboard of the card reader and waits for validation. [t15b2-1]
      The step 5.b.ii is repeated until a successful validation or the Cashier presses the button for cash payment. [t15b2-2, n15b2-1]

6. Completed sales are logged by the Trading System and sale information are sent to the Inventory in order to update the stock. [t16-1]
7. The Printer writes the receipt and the Cashier hands it out to the Costumer. [t17-1]
8. The Customer leaves the Cash Desk with receipt and goods.

Alternative or Exceptional Processes

- *In step 3: Invalid item identifier if the system cannot find it in the Inventory.* [p13-4]
  1. The System signals error and rejects this entry. [t13-3]
  2. The Cashier can respond to the error as follows:
     (a) It exists a human-readable item identifier: [p13-5]
        i. The Cashier manually enters the item identifier. [t13-4]
        ii. The System displays the description and price. [t14-1]
     (b) Otherwise the product item is rejected. [p13-6]
- *In step 5.b: Card validation fails.* [p15b2-2]
  1. The Cashier and the Customer try again and again.
  2. Otherwise the Cashier requires the Customer to pay cash.
- *In step 6: Inventory not available.* [p16-1]
  The System caches each sale and writes them into the Inventory as soon as it is available again. [t161-1]

UC 2 - Manage Express Checkout

Brief Description If some conditions are fulfilled a Cash Desk automatically switches into an express mode. The Cashier is able to switch back into normal mode by pressing a button at his Cash Desk. To indicate the mode the Light Display shows different colors.

Involved Actors Cashier, Cash Box, Light Display, Card Reader

Precondition The Cash Desk is either in normal mode and the latest sale was finished (case 1) or the Cash Desk is in express mode (case 2).

Trigger This use case is triggered by the system itself.

Postcondition The Cash Desk has been switched into express mode or normal mode. The Light Display has changed its color accordingly.

Standard Process

1. The considered Cash Desk is in normal mode [p2-1] and just finished a sale which matches the condition of an express checkout sale. Now 50% of all sales during the last 60 minutes fulfill the condition for an express checkout.
   (a) This Cash Desk, which has caused the achievement of the condition, is switched into express mode. [t21a-1]

(b) Furthermore the corresponding Light Display is switched from black into green to indicate the Cash Desk's express mode. [t21b-1]
(c) Paying by credit card is not possible anymore. [t21c-1]
(d) The maximum of items per sale is reduced to 8 and only paying by cash is allowed. [t21d-1]

2. The Cash Desk is in express mode [p2-2] and the Cashier decides to change back into normal mode.
   (a) The Cashier presses the button *Disable Express Mode*. [t22a-1]
   (b) The color of the Light Display is changed from green into black color. [t22b-1]
   (c) Cash and also card payment is allowed and the Costumer is allowed to buy as much goods as he likes. [t22c-1]

UC 3 - Order Products

Brief Description The Trading System provide the opportunity to order product items.

Involved Actors Store Manager

Precondition An Overview over the Inventory is available and the Store Client was started.

Trigger The Store Manager decided to buy new product items for his store.

Postcondition The order was placed and a generated order identifier was presented to the Store Manager.

Standard Process

1. A list with all products [n3-1] and a list with products running out of stock are shown. [n3-2, p3-1, t31-1]
2. The Store Manager chooses the product items to order and enters the corresponding amount. [t32-1]
3. The Store Manager presses the button *Order* at the Store Client's GUI. [t33-1]
4. The appropriate suppliers are chosen and orders for each supplier are placed. An order identifier is generated for each order and is shown to the Store Manager. [t34-1, t34-2, t34-3]

UC 4 - Receive Ordered Products

Brief Description Ordered products which arrive at the Store have to be checked for correctness and inventoried.

Involved Actors Stock Manager

Precondition The Store Client was started and the part Inventory of the Trading System is available.

Trigger The ordered products arrive at the Store.

Postcondition The Inventory is updated with the ordered products.

Standard Process

1. Ordered products arrive at the stock attached by an order identifier which has been assigned during the ordering process. [n4-1]
2. The Stock Manager checks the delivery for completeness and correctness. [p4-1, t42-1]
3. In the case of correctness, the Stock Manager enters the order identifier and presses the button *Roll in received order*. [t43-1]
4. The Trading System updates the Inventory. [t44-1]

Alternative or Exceptional Processes

- *In step 2: Delivery not complete or not correct.* [p4-2]
  The products are sent back to the supplier and the Stock Manager has to wait until a correct and complete delivery has arrived. This action does not recognized by the System.

UC 5 - Show Stock Reports

Brief Description The opportunity to generate stock-related reports is provided by the Trading System.

Involved Actors Store Manager

Precondition The reporting GUI at the Store Client has been started.

Trigger The Store Manager wants to see statistics about his store.

Postcondition The report for the Store has been generated and is displayed on the reporting GUI.

Standard Process

1. The Store Manager enters the store identifier and presses the button *Create Report*. [t51-1]
2. A report including all available stock items in the store is displayed. [t52-1]

UC 6 - Show Delivery Reports

Brief Description The Trading System provides the opportunity to calculate the mean times a delivery from each supplier to an considered enterprise takes.

Involved Actors Enterprise Manager

Precondition The reporting GUI at the Store Client has been started.

Trigger The Enterprise Manager wants to see statistics about the enterprise.

Postcondition The report for the Enterprise has been generated and is displayed to the Enterprise Manager.

Standard Process

1. The Enterprise Manager enters the enterprise identifier and presses the button *Create Report*. [t61-1]
2. A report which informs about the mean times is generated. [t62-1]

UC 7 - Change Price

Brief Description The System provides the opportunity to change the sales price for a product.

Involved Actors Store Manager

Precondition The store GUI at the Store Client has been started.

Trigger The Store Manager wants to change the sales price of a product for his store.

Postcondition The price for the considered product has been changed and it will be sold with the new price now.

Standard Process

1. The System presents an overview over all available products in the store. [t71-1]
2. The Store Manager selects a product item [t72-1] and changes its sales price. [t72-2]
3. The Store Manager commits the change by pressing ENTER. [t73-1]

UC 8 - Product Exchange (on low stock) Among Stores

Brief Description If a store runs out of a certain product (or a set of products; "required good"), it is possible to start a query to check whether those products are available at other Stores of the Enterprise ("providing Stores"). Therefore the Enterprise Server and the Store Servers need to synchronize their data on demand (one scheduled update per day or per hour is not sufficient). After a successful query the critical product can be shipped from one to other Stores. But it has to be decided (using heuristics to compute the future selling frequency),

whether the transportation is meaningful. For example, if the product is propably sold out at all Stores within the same day, a transportation does not make sense.

Expressed in a more technical way one Store Server is able to start a query at the Enterprise Server. The Enterprise Server in turn starts a query for products available at other Stores. As the Enterprise Server does not have the current global data for Stores at any time (due to a write caching latency at the Store Servers) the Enterprise Server has to trigger all Store Servers to push their local data to the Enterprise Server.

Involved Actors This use case is not an end-user use case. Only servers are involved.

Precondition The Store Server with the shortage product is able to connect to the Enterprise Server.

Trigger This use case is triggered by the system itself.

Postcondition The products to deliver are marked as incoming or unavailable, respectively, in the according Stores.

Standard Process

1. A certain product of the Store runs out.
2. The Store Server recognizes low stock of the product. [t82-1]
3. The Store Server sends a request to the Enterprise Server (including an identification of the shortage products, and a Store id) [t83-1]
4. The Enterprise Server triggers all Stores that are "near by" (e. g. ¡300 km) the requiring store, to flush their local write caches. So the Enterprise Server database gets updated by the Store Server. [t84-1, t84-1]
5. The Enterprise Server does a database look-up for the required products to get a list of products (including amounts) that are available at providing Stores. [t85-1]
6. The Enterprise Server applies the "optimization criterion" (specified above) to decide, whether it is meaningful to transport the shortage product from one store to another (heuristics might be applied to minimize the total costs of transportation). This results in a list of products (including amounts) per providing store that have to be delivered to the requiring Store. [t86-1]
7. The Store Server, initially sending the recognition of the shortage product, is provided with the decision of the Enterprise Server. [t87-1]
   (a) The required product is marked as incoming. [t87-2]
8. The Store Server of a near by Store is provided with information that it has to deliver the product. [t88-1]
   (a) The required product is marked as unavailable in the Store. [t88-2]

Alternative or Exceptional Processes

- The Enterprise Server is not available: The request is queued until the Enterprise Server is available and then is send again.
- One or more Store Servers are not available: The Enterprise Server queues the requests for the Store Servers until they are available and then resend them. If a Store Server is not available for more than 15 minutes the request for this Server is canceled. It is assumed, that finally unavailable Store Servers do not have the required product.

Extension on use case 8 - Remove Incoming Status

Brief Description If the first part of use case 8 (as described above) has passed, for moved products an amount marked as incoming remains at the Inventory of the Store receiving the products. An extension allows to change that incoming mark via a user interface at the Store Client if the moved products arrive at a Store.

Precondition The Inventory is available and the Store Client has been started.

Trigger The moved products (according to UC8) arrive at the Store.

Postcondition For the amount of incoming products the status "incoming" is removed in the Inventory.

Standard Process

1. The products arrive at the stock of the Store.
2. For all arriving products the Stock Manager counts the incoming amount.
3. For every arriving product the Stock Manager enters the identifier and its amount into the Store Client.
4. The system updates the Inventory.

Alternative or Exceptional Processes

- If the entered amount of an incoming product is larger than the amount accounted in the Inventory, the input is rejected. The incoming amount has to be re-entered.

1.3 Extra-Functional Properties

The following table includes CoCoME's extra-functional properties in terms of timing, reliability, and usage profile related information. They can be seen as guiding values when conducting QoS-analysis with CoCoME. The results from different methods can be compared more adequately if they are based on the same extra-functional properties.