

CSSE 374 – Software Architecture and Design I

Homework 2

Objective

To apply what you have learned about UML domain modeling by producing a set of system sequence diagrams and requisite operations contracts, and by beginning a preliminary Logical Architecture for the BrickBuster Video System (BBVS).

Due Date

5 p.m., Tuesday, Week 3

Tasks

1. Read through the attached Domain Model, scope statement, requirements, and use cases. Develop the system sequence diagrams (SSD) for the relevant elements in the Domain Model. While there are not a lot of detailed Use Cases, use your creative imagination to come up with reasonable sequences for your SSDs. The objective of this exercise is to give you exposure to the process of reasoning about behavior at the analysis level.
2. From the SSDs you produce, identify *at least three* operations (methods) that need some level of detail to ensure constraints are met or rules are followed. For each operation that you identify, give a sentence or two describing why it warrants additional detail. Write the operations contracts (OC) for each of these.
3. In preparation for creating a Logical Architecture in Homework 3, begin choosing application domain classes. From the Domain Model, SSDs, and Operations contracts, identify key classes from the application domain layer and organize them into packages. Give a UML Package Diagram showing the organization and a paragraph or two justifying it. You do not need to show dependencies between packages at this time.

Submitting Your Work

Please submit your SSDs, Operations Contracts, and Package Diagram as a single document to the appropriate Angel Homework dropbox.

Steve's Section: You can submit either a **pdf** file **or** a **Word** document. Name your document *HW3-Analysis*, with the appropriate extension (*pdf*, *doc*, or *docx*). Please include your name on the first page of the document.

Shawn's Section: Please submit a **pdf** file with a cover page containing your Name, Assignment Title, Date, and Campus Mail number. Please name the document: <your last name>HW2-Analysis.pdf (e.g., Bohner-HW2-Analysis.pdf).

BrickBuster Video System (BBVS)

Scope: The BrickBuster Video System (BBVS) will provide a streaming video-on-demand service for Internet customers, available on PC's, iPhones, XBOX's and other devices. This new system will replace the company's previous brick-and-mortar video store system - hence the name of the new system!

Customers: The customers at present come to their local store to browse for videos and for related services, such as ordering a video that's not in the store. They pay a rental fee for each video, take it home, watch it as many times as they like over a period of a few days, then return it on time to avoid paying a late fee. Each customer has a name, address, and phone number. They must show a photo ID to get a free membership with the store. They then get a membership card, which they present with their payment when they rent videos. Payments can be made using cash, debit card, or credit card. If a video is returned late, or is lost, the extra fee for this is assessed the next time the customer rents a video.

Video store: Has the videos to be rented, as well as the records of each local customer-member. The store includes retail terminals for ringing-up sales. Each video has a title and a unique ID that is scanned during the sales transaction.

Shelves: Each video in the store has one or more copies, all conveniently located together on a shelf area designated for that particular video. The shelf actually has just the cases, and the customers bring the cases to the checkout counter, where they are filled with the proper DVD's. Each store has a recommended shelf layout, showing the store personnel where the videos are supposed to be, for efficient stocking.

Provisioning: The store gets regular deliveries of new videos, via trucks from the warehouses or directly from DVD suppliers. The store system has information about each shipment, showing where the new DVD's are supposed to go, and what they replace.

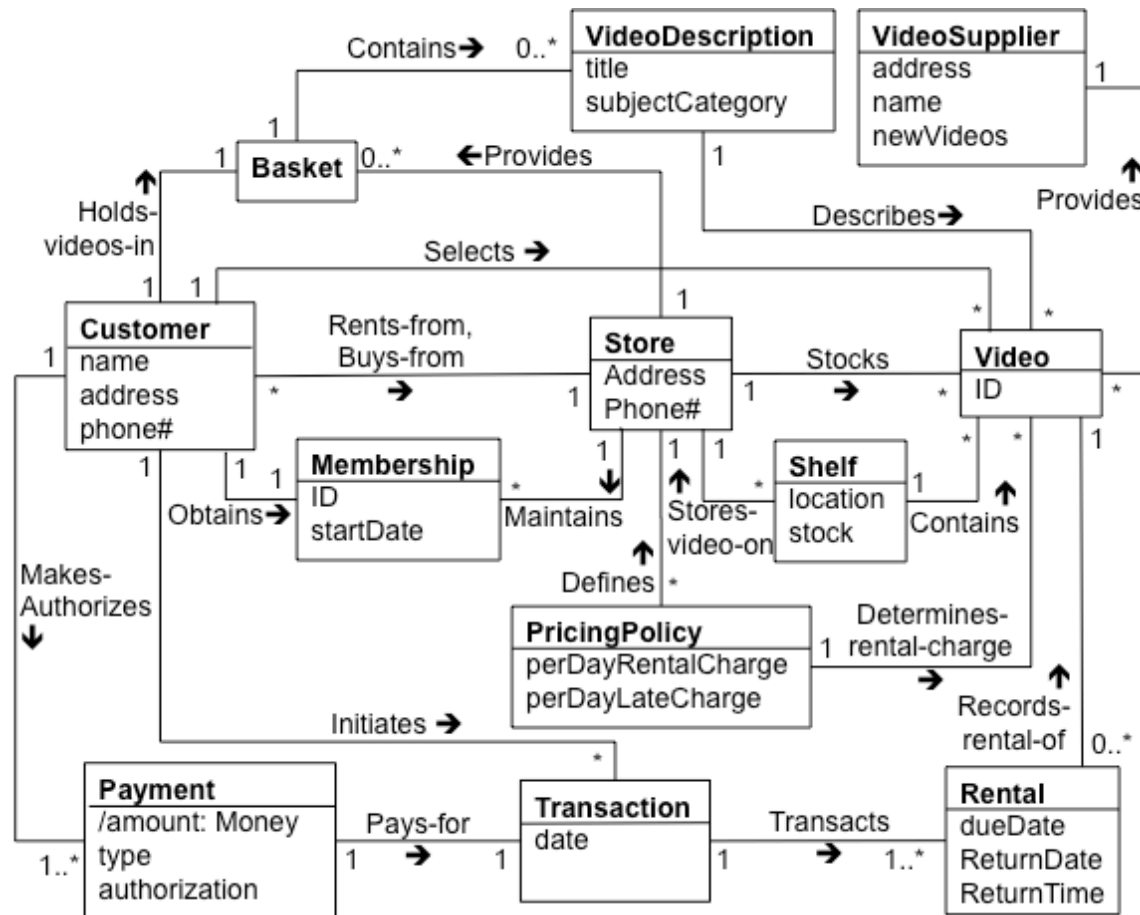
Unprovisioning: The old DVD's that the store wants to get rid of are largely sold to customers, and removed from inventory as they are sold. Annually the store does a physical inventory that removes stolen or lost videos from the headquarters system.

Headquarters servers: The store inventory is kept remotely, and indicates which actual copies of which videos are supposed to be in-store, and which are checked-out to which customer. Periodic inventory checks are made to determine shrinkage.

Kiosk: The store also has a kiosk from which customers can order videos to be delivered later. This kiosk has access to a remote database of all available videos.

Store employees: Include a store manager and store associates. The manager has access to employee records and store financial data. All employees can ring-up sales at the counter, and do inventory operations.

A Domain Model for BrickBuster Video System (Homework 1):



The Domain Model depicted above (a relatively detailed version of Homework 1) is intended to provide a basic framework from which the SSDs, Operations Contracts, and Logical Architecture packages/components can be derived. Note that the model is incomplete – largely addressing the Use Cases provided but, omitting things like rental returns, customer services, and the like. Conceptual classes represented in the Domain Model, provide the basic domain objects used in the BBVS. In the figure above, the Customer is a key class that many of the other classes revolve around. The Customer Selects Videos, Buys-from or Rents-from the Store those Videos, and Initiates Transactions, including Making and Authorizing Payments.

New Videos are Provided by Video Suppliers while older releases are sold to customers. The Customer selects from Videos (in the store) from the stock on Shelves, and holds them a Basket – actually, like in a store, these just contain the empty case with the Video Description. The selected videos are then rented or purchased in the rental transaction. Hence, Transaction also plays a central role - it is Paid-for by a customer Payment, and it Transacts the Rental of the Videos. The Rental also is key - it Records-rental-of Videos. The Store is in

multiple activities, in that it is where the Customer Rents-from, and it also Stocks the Videos.

Note that we did not include Shopping itself as a class, though this certainly could have been done. In the model, we also did not include other parties or nouns of interest which were described in the Scope description, above.

Use Cases (from Homework 1):

In getting these use cases from the customer, we tried to make them separate from the way the customer now does business. That is, we wanted them to be "how they wanted the new system to work." At the same time, we wanted to keep the use cases general enough that they might apply to more than one way of running a video business.

UC1: Customer rents videos

Preconditions: Customer has a membership, has selected videos they want to rent, and made the system aware of their choices.

Actor: Customer (if self-service or remote), or store associate (if in a store).

Main flow:

1. Actor indicates to rent first item (e.g., clicking "rent" on a networked device, or scanning it physically in a store).
2. System verifies immediate availability, waits for actor to make next option.
3. Actor indicates they are done selecting.
4. System shows total, prompts for payment.
5. Actor selects method of payment, entering additional data if needed (e.g., credit card number).
6. System verifies the payment has gone through, schedules the goods for rental (e.g., sets up a window to click on to view the video remotely, or tells the store clerk where to find the DVD).

Alternate flows (among many):

- 2a. System tells actor that the video is not currently available, and provides information on when it will be.
- 3a. Actor buys additional items, in the same way, if desired, returning to step 3 after each.
- 6a. System rejects method of payment, asks actor for alternative.

Postcondition: Rental transaction is complete.

UC2: System lets customer select videos to rent

Preconditions: None

Actor: Customer

Main flow:

1. "Store" shows videos available for rental. (This could be in stages, such as indicating areas of the store for customer to browse, or responding to a search request.)
2. Customer picks a video to rent.
3. System shows items in "basket," suggests checkout.

Alternate flows:

- 1a. Item is not in stock. "Store" shows when it is available, if possible.
- 2a. Customer continues in this loop, selecting multiple videos.

Postcondition: Customer has products ready to rent.