

Requirements Modeling and Analysis in OO

Yilong Yang

Department of Computer Science and Information
Faculty of Science and Technology
University of Macau
yylonly@gmail.com

June 18, 2019

Contents

- 1 Overview
- 2 Use Case Diagram
- 3 System Sequence Diagram
- 4 Contract of System Operation
- 5 Conceptual Class Diagram
- 6 Conclusion

Contents

- 1 Overview
- 2 Use Case Diagram
- 3 System Sequence Diagram
- 4 Contract of System Operation
- 5 Conceptual Class Diagram
- 6 Conclusion

Requirements Modeling

- Requirements errors are one of the causes leading failings in software projects.
- Careful requirements modeling along with systematic validation helps to reduce the uncertainty about target systems.
- The goal of requirements validation is to construct the consistent requirements for the needs of target users.

Requirements Modeling

However, this process is complicated, and it can be hard to produce a correct and complete requirements specification. The complexity is due to the following interrelated attributes:

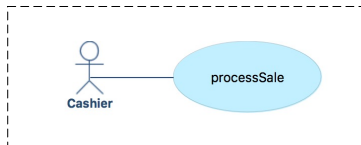
- 1) the complexity of application domains and business processes
- 2) the uncertainty of clients and domain experts about their needs
- 3) the lack of the understanding of system developers about application domains
- 4) the difficulties of the understanding between system developers and clients

Requirements Modeling

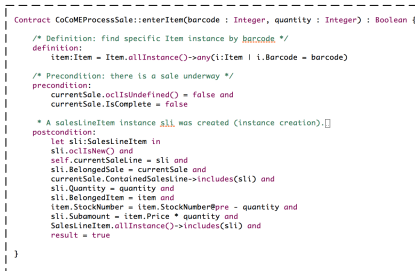
The Unified Modeling Language (UML) is a de facto standard for requirements modeling and system design. For requirements modeling and analysis, UML provides

- Use case diagram
- System sequence diagram
- Contract of system operation
- Conceptual Class diagram

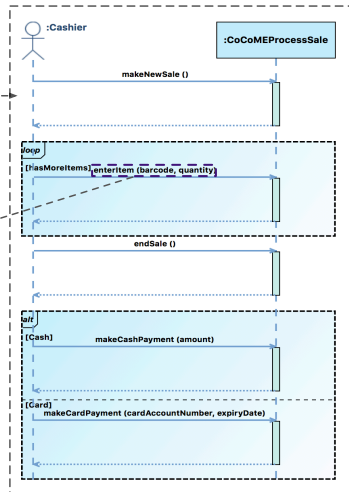
Requirements Model in UML



1. Use Case Diagram

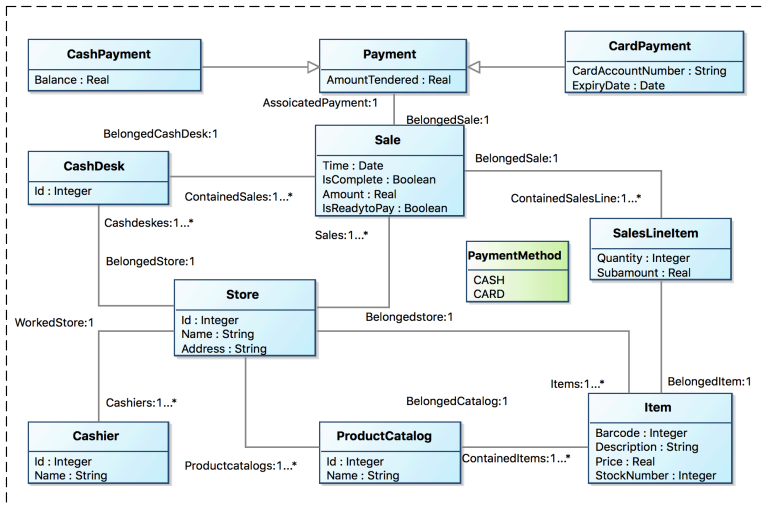


3. Contracts of System Operations



2. System Sequence Diagrams

Requirements Model in UML



4. Conceptual Class Diagram

Contents

- 1 Overview
- 2 Use Case Diagram**
- 3 System Sequence Diagram
- 4 Contract of System Operation
- 5 Conceptual Class Diagram
- 6 Conclusion

Use Case Diagram

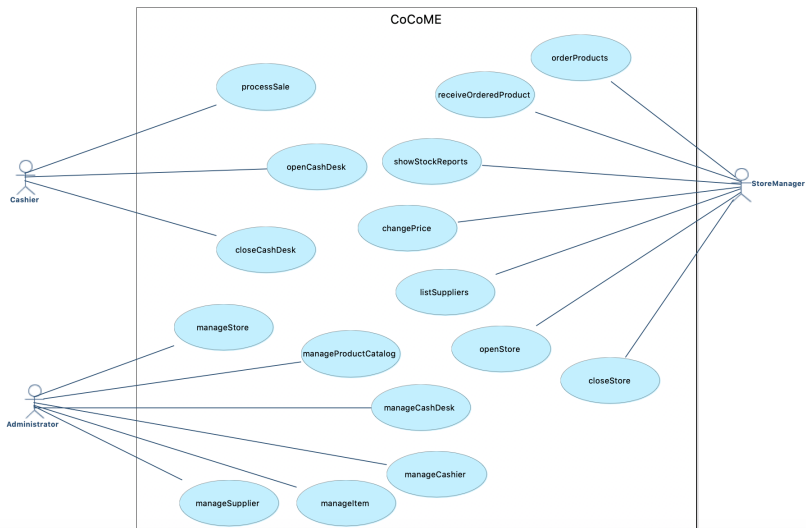
- A use case diagram captures domain processes as use cases in terms of interactions between the system and its users.
- It helps customers and domain experts specify functional requirements of the target system.

Use Case Diagram

It contains:

- a set of use cases for a system
- the actors represented a type of users of the system or external systems that the system interacts with
- the relations between the actors and these use cases, and relations among use cases.

Use Case Diagram - CoCoME



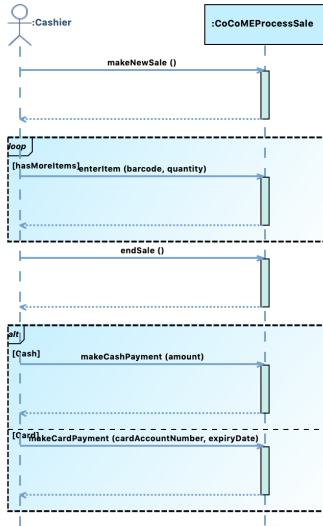
Contents

- 1 Overview
- 2 Use Case Diagram
- 3 System Sequence Diagram**
- 4 Contract of System Operation
- 5 Conceptual Class Diagram
- 6 Conclusion

System Sequence Diagram

- A system sequence diagram describes a particular domain process of a use case.
- It contains the actors that interact with the system, the system and the system events that the actors generate, their order, and inter-system events.
- It helps customers to find system operations and provides the sequences to interact with the prototype for requirements validation.

System Sequence Diagram



System Sequence Diagrams vs Sequence Diagram

Compared with the sequence diagram in design models, a system sequence diagram

- treats all systems as a black box
- contains system events across the system boundary between actors and systems
- without object lifelines and internal interactions between objects.

Contents

- 1 Overview
- 2 Use Case Diagram
- 3 System Sequence Diagram
- 4 Contract of System Operation**
- 5 Conceptual Class Diagram
- 6 Conclusion

Contract of System Operation

The contract of a system operation specifies the conditions:

- that the state of the system is assumed to satisfy before the execution of the system operation, called the *pre-condition* of the system operation
- the system state is required to satisfy after the execution (if it terminated), called the *post-condition* of the system operation

Contract of System Operation

Typically,

- the pre-condition specifies the properties of the system state that need to be checked when system operation is to be executed,
- the post-condition defines the possible changes that the execution of the system operation is to realize.

System State in OO

- A state of an object-oriented system is about the existing objects together with their properties/states and relations/links.
- The state is an object diagram defined by a conceptual class diagram plus the input and returns parameters of the operations.

System State in OO

The changes of system states are classified into

- i) new objects created (together with initial values of attributes and links of associations),
- ii) attributes of existing objects (in the current state) modified,
- iii) new links among existing objects formed, and iv) existing objects and/or links are removed.

Contract of System Operation

//Signature

```
Contract CoCoMEProcessSale::enterItem  
  (barcode : String, quantity : Real) : Boolean {
```

//Definition Section

definition:

//Find Object

```
item:Item = Item.allInstance()->any(i:Item | i.Barcode = barcode)
```

//Pre-condition Section

precondition:

```
currentSale.oclIsUndefined() = false and
```

```
currentSale.IsComplete = false and
```

```
item.oclIsUndefined() = false and
```

```
item.StockNumber > 0
```

//Post-condition Section

Contract of System Operation

```
postcondition:  
  //Create an Object  
  let sli:SalesLineItem in  
  sli.oclIsNew() and  
  //Add Links  
  self.currentSaleLine = sli and  
  sli.BelongedSale = currentSale and  
  currentSale.ContainedSalesLine->includes(sli) and  
  sli.BelongedItem = item and  
  //Modify Attributes  
  sli.Quantity = quantity and  
  sli.Subamount = item.Price * quantity and  
  item.StockNumber = item.StockNumber@pre - quantity and  
  //Add an Object  
  SalesLineItem.allInstance()->includes(sli) and  
  result = true  
}
```

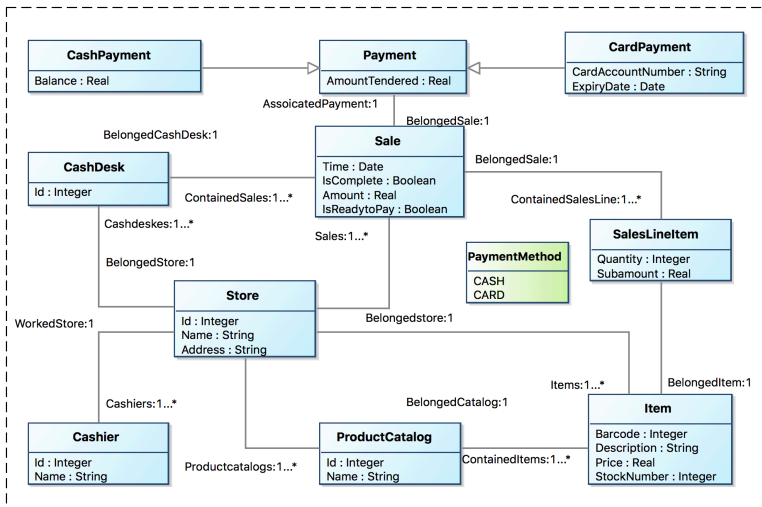
Contents

- 1 Overview
- 2 Use Case Diagram
- 3 System Sequence Diagram
- 4 Contract of System Operation
- 5 Conceptual Class Diagram**
- 6 Conclusion

Conceptual Class Diagram

- A conceptual class diagram is a concept-relation model, which illustrates abstract and meaningful concepts and their relations in the problem domain.
- The domain concepts are specified as classes, the relations of the concepts are specified as the associations between the classes, and the properties of the concepts are specified as the attributes of the classes.

Conceptual Class Diagram



Conceptual Class Diagram vs Class Diagram

Compared with the class diagram in design models,

- a conceptual class diagram does not contain operations in the classes.
- It focuses on specifying the domain concepts but not for how to encapsulate operations.

Case Studies

- ATM - Automated Teller Machine
- CoCoME - Supermarket System
- LibMS - Library Management System
- LoanPS - Loan Processing System

RM2PT



Conclusion

The Unified Modeling Language (UML) is a de facto standard for requirements modeling and analysis. It provides

- Use case diagram
- System sequence diagram
- Contract of system operation
- Conceptual Class diagram