

BCLS Notes

Beyond the Classroom Lecture Series

Volume 2 Number 2 Fall 2006

BCLS Notes are
produced for the benefit
of our students by the
Faculty of School of
Business, Public
Administration, and
Information Sciences,
Long Island University



Copyright © 2006 by Long Island University

Business Modeling with UML

Virginia de Paula, PhD

Computer Science Department

School of Business, Public Administration and Information Systems

Brooklyn Campus

Long Island University

virginia.depaula@liu.edu

Abstract

In Information System Analysis and Design, two groups of people work to construct the software that matches the needs of an organization. In the first moment of the software development, business analysts and systems analysts work together trying to establish a common vocabulary that will reduce communication problem between the stakeholders. Therefore, business process analysts perform business modeling and should communicate with the system analysts and software engineers to develop the new business architectures. In the present work, we show how business analysts and systems analysts (software engineers) can clearly integrate their roles in software development by using UML (Unified Modeling Language) – a graphical language to visualize, specify, construct, and document software-intensive systems.

1. Introduction

In Software Development, the stakeholders involved in the process of constructing a software can be classified into two main categories: those who understand the information system being automated – the domain, and those with the technical skills to turn the information system in a software that should be loyal representation of the real system besides adding business value to the organization.

Business modeling is particularly useful to understand business processes, allowing the construction of the corresponding software system.

Usually, there are three basic reasons you might need to model a business [1]:

- To re-engineer a business: the goal is to re-think how the business operates and interacts with the outside world;
- To improve a business process: the goal is to streamline how the business works, and to enhance its competitiveness;
- To automate a business process: the goal is to have a model of the current business to understand the environment in which a software system will function.

Business process analysts usually perform business modeling. However, when the goal is to develop a software system, the responsibility for modeling is in hands of the requirements analysts, which, along with business analysts, validate the system requirements. It is worth saying that requirement analysis is part of the software development process. Therefore, a requirement analyst is a specialized software engineer.

This document is organized as following. Section 2 presents an overview of the Unified Modeling Language (UML) with emphasis in Use-Cases diagrams. Section 3 introduces the existing extension for Business Modeling. In Section 4, we present two case studies to illustrate the UML extensions for Business Modeling. In Section 5, we point out some conclusions.

2. UML

Unified Modeling Language (UML) is a graphical modeling language. As said it is a *language* and it has the very particular characteristic to have diagrams as its notation [4]. It is worthy saying that each diagram element has a very well defined semantics, which guarantees the meaning and preciseness of the diagram.

UML provides different diagrams to deal with each step in a software development process, allowing business analysts to have an abstract perspective of the software system and developers to have a different and more technical point of view. Therefore, from the very beginning, the two groups can clarify requirements and discuss using the same terminology and illustration.

The diagrams provided by UML are:

2.1. Use Case diagrams;

- 2.2. Class diagrams;
- 2.3. Object diagrams;
- 2.4. Activity diagrams;
- 2.5. Sequence diagrams;
- 2.6. Communication diagrams;
- 2.7. Component diagrams;
- 2.8. Packet diagrams;
- 2.9. Deployment diagrams.

Use cases, Class, Object, Activity, and Sequence are the most used diagrams to improve communication between the stakeholders involved in the definition of the software system. Considering this document is an introduction to business modeling, we give more attention to Use Cases diagrams. Activity and Sequence diagrams do not need extensions to be used for business modeling. We will illustrate their use in Section 4, when we present the Case Study.

2.1 Use Case Diagrams

Use Case diagrams are very useful to clarify requirements and establish a common vocabulary to improve communication between stakeholders. A Use Case Diagram can include a set of *cases* along with the interaction of the users and/or entities with the cases (system). An entity is any subdivision of the organization that needs to interact with the specific software system being developed. Users and entities that interact with the system are called *actors*.

The notation of use case diagrams is like as following:

- Use cases are represented by ellipses;
- Actors are represented by a “doll” or by rectangle illustrating another subsystem, a department or division, etc;
- A frame around the use cases symbolizes the system boundaries.

Figure 1 illustrates the use cases diagram notation for the magazine circulation business process [4]. The use cases diagrams can be accompanied by some complementary information such as: name, brief description, performance goals, benefit/value, workflow/flow of events, special requirements, extension points, relationships, and activity diagrams.

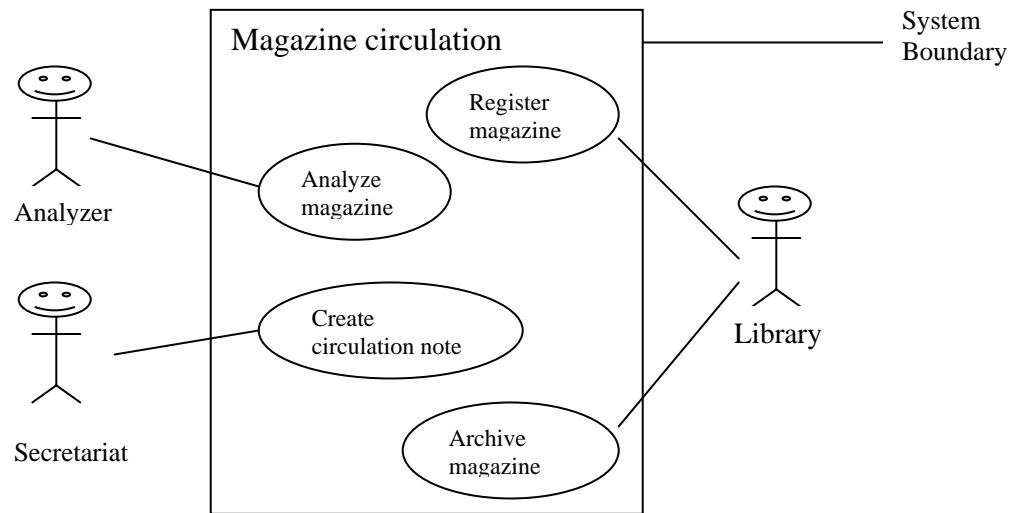


Figure 1: Use cases of the magazine circulation business process

Use cases diagrams are usually presented in conjunction with a use case description to show the external system behavior in a limited working situation from the point of view of a user.

A specific use case may depend on other use cases. UML provides some stereotypes to clearly specify dependencies. The `<<include>>` stereotype is used when one use case should always happen in conjunction with another one. The `<<extend>>` stereotype is used when optionally a use case happens in conjunction with another one.

Using the example of an ATM machine (Figure 2), any operation (withdraw, get balance, etc) should ask for security authentication. In this case, the `<<include>>` stereotype can be used.

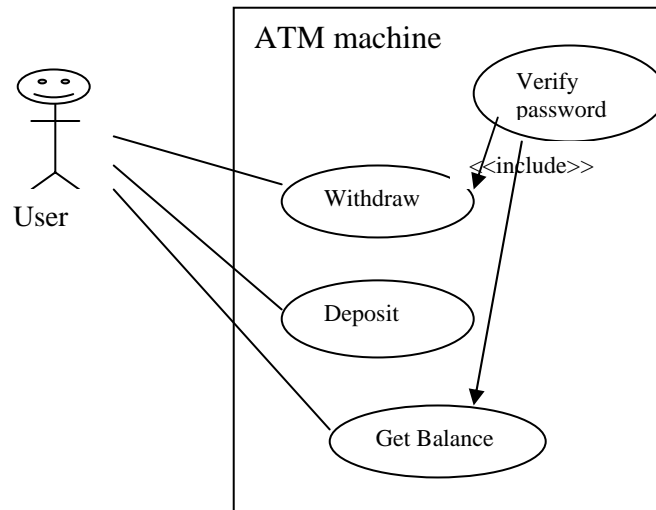
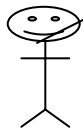


Figure 2: Use cases of the ATM machine business process

3. UML Extension for Business Modeling

One of the main UML extensions to model business is the business use case diagram. Business use cases describe business processes and we can say that to actually understand a business, it is necessary to know who the business interacts with, i.e., who puts demands on it and/or who is interested in its output. Different types of “interactors” are represented as business actors [2]. The graphical element to represent a business actor is:



Business use cases are represented by:



Besides the business use cases, the official UML extension for Business Modeling [3] proposes the inclusion of some Business objects:

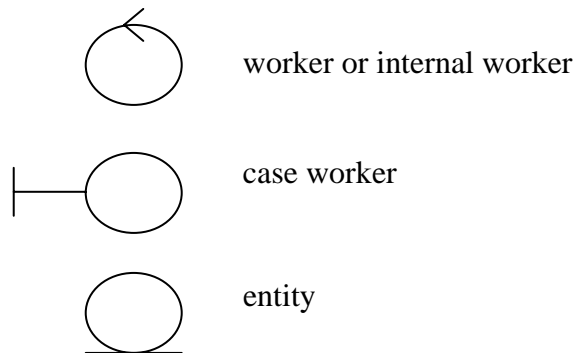
- worker
 - case worker
 - internal worker
- entity

A *worker* is a class to represent a human that acts within a system. A class is used in object-orientation to describe *things* that can be classified as being of the same category because they have common characteristics and behavior. Examples of classes: faculty, student, chair, etc.

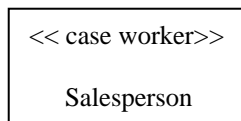
A *case worker* is a worker who interacts directly with actors outside the system, while a *internal worker* interacts with other workers and entities inside the system.

Entities represent objects that worker access, inspect, manipulate, produce, and so on.

The following special icons are proposed to represent the business modeling extension.



In UML rectangle are also used to represent a class and class stereotypes can be shown with keywords in guillemets within the usual class symbol. For example:

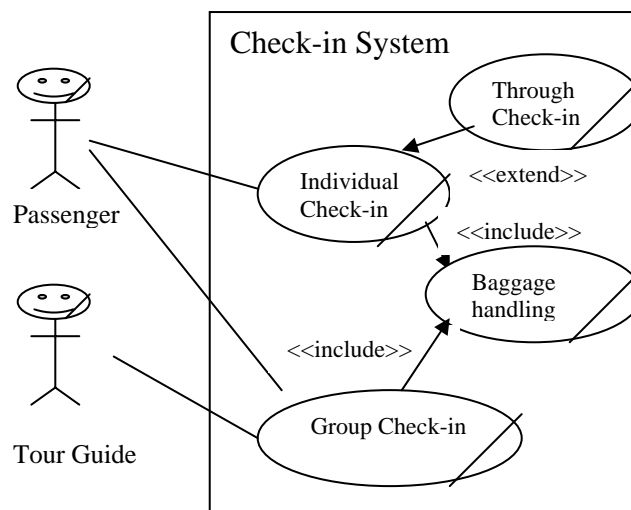


4. Case Study

4.1. Check-in system

The check-in system [2] can be used for any type of transportation: airplane, train, bus, etc. The check-in can be done individually or in groups. In this case, can be done by the passenger or by the tour guide responsible for the group. If the passenger is part of a group, but is checking in individually, he or she should provide this information.

An initial use case diagram is presented below.



5. Conclusion

UML is a very popular modeling language between software developers. As a graphical language with a formal semantics associated to each graphical element,

it brings the easiness of reading figures and diagrams along with a precise interpretation of the diagrams.

Communication between stakeholders involved in the development of software for a specific domain has been an issue for several years. System analysts have been successfully using UML, specially Use Cases diagrams to find common terminology between stakeholders, improving the software development process and making software development and maintenance less expensive.

References

[1] Baker, Bryon. “Business Modeling with UML: The Light at the End of the Tunnel”. In: The Rational Edge.

[2] Heumann, Jim. “Introduction to business modeling using the Unified Modeling Language (UML)”

URL: <http://www-128.ibm.com/developerworks/rational/library/360.html>

[3] Rational Software. UML Extension for Business Modeling – version 1.1. September 1997. URL: http://umlcenter.visual-paradigm.com/umlresources/exte_11.pdf

[4] Oestereich, Bernd. Developing Software with UML – Object-oriented Analysis and Design in Practice. Second Edition. Addison Wesley. 2002.