Journal of Knowledge Management, Economics and Information Technology

ISSN 2069 – 5934
ScientificPapers.org

http://www.scientificpapers.org

UML IN BUSINESS ADMINISTRATION

Daniel Ionita, Romano Britanica University, Bucharest, Romania

Abstract: *The article elaborates weather UML, primarily used in software engineering, can be a useful tool in business modeling and administration. By analyzing the advantages the modeling language has to offer we find that UML is visual and object oriented and that it is useful in expressing structure, interaction and behavior as well. With its help managers and business people can build models and diagrams to help put things into perspective. "Case Study 1" shows UML can be used as an analysis tool in business modeling to help increase the complexity and depth of the event or project that is being developed. "Case Study 2" attempts to prove that UML can also be efficiently used in finding solutions to newly appeared problems in a business environment. Despite the practicality of the Unified Modeling Language there is still some criticism brought to it. Some programmers consider it to be hard to learn and some developers claim that it is too abstract. The article concludes that despite the minor drawbacks; due to its adaptability and complex visual models, it is a very useful tool that adds value to the modeling of business structures and processes.*

Keywords: *UML, Unified Modeling Language , Business Administration, Business Modeling, Diagrams*

What is UML?

UML or in full Unified Modelling Language is a standardized modelling language used in the development of software engineering. To be more precise, it offers a set of graphical notation methods (diagrams) that help create visual schematics for software structures.

UML uses techniques from data modelling; object oriented modelling, business modelling and component modelling. It can be used with all processes throughout the software development life cycle and across different implementation technologies. It is useful in visually expressing structure, processes and behaviour.

How is UML related to business administration?

As the industrial era ended a new era arose. Today we live in the era of information and technology. The success of many businesses today is determined by the correct flow and synergy of informational processes with physical processes. Every manager of today must ask themselves a couple of questions. Do the information systems effectively support their way of working? Do the systems adapt easily to change? Is information used as an important strategic resource in the business? Is the information adequate and correct?

To answer these questions managers need to turn to business modelling. A model is a simplified view of a complex reality. It is a means to creating abstraction, allowing them to eliminate irrelevant details and focus on one or more important aspects at a time. Business modelling helps managers to better understand the key mechanism of an existing business, to act as the basis for improving the current business structure and operation, to show the structure of an innovative business concept, to experiment with a new business concept, to copy or study a concept used by a competitive or competing company, to identify profitable opportunities etc.

Despite the fact that UML was created initially for software development, it also turned out to be a very useful tool in business modelling. UML has been successfully applied to the modelling of just about any system one can think of, from data structures to embedded real-time systems, to XML (Extensible Mark-up Language) schemas, and to real-world organizations from family businesses to multinational enterprises.
So it's not surprising that business analysts find UML very handy for visualizing organizational processes.

It has been proven that by using UML as a business modelling language, managers can better manage complexity, reduce development time and improve the quality of business systems.

Why is UML so useful in business modelling?

There are five main reasons why UML is considered very useful in business modelling.

1. UML breaks the communication barrier

UML provides a common language for business administrators and software developers. By using symbols and diagrams business developers are now able to accurately communicate their desires to software developers. Even though this sounds simple, before UML was applied in business modelling, there was almost always some kind of disconnection between the design of the business and the design of the software.

2. UML is visual

UML can construct a clear visual model for businesses. With visual models it is easier for managers and business developers to see the business structure as a bigger picture, to notice possible opportunities, to understand and develop the flow of processes (virtual and physical), to see bottlenecks, to understand the flow of information and many others.

3. UML is object oriented

Objects are entities that parallel objects in the "real world": they have some properties (such as name and address), relate in various ways to other things around them, and will exhibit some sort of behaviour when acted upon. Object-oriented models can therefore very closely approximate actual business objects and systems, even to the extent of portraying how different parts of the system work together dynamically.

4. UML is able to express structure, interaction and behaviour

As business systems have become more and more automated, they have also become more dependent on software systems. Understanding how those systems work and interact and how to guide their successful evolution along with the evolution of business environment is critical to the success of a business. UML is useful here because it enables the user to display information from many different angles through the use of diagrams, expressing structure, interaction and behaviour.

Structure diagrams:
Class diagram: is a diagram used to describe the structure of a system by presenting the system's classes, their attributes, and the relationships among the classes.
Component diagram: is used to show how a system is split up into components and shows the dependencies among these components.
Object diagram: is also a way to depict structure, partially or complete, at a specific time in the presented system.

Behaviour diagrams:
Activity diagram: represents the business and operational workflows of components in a system. Activity diagrams are also used to represent the overall flow of control.
State machine diagram: depicts the various states that an object may be in and the transitions between those states and also what events cause those transformations.
Use case diagram: is a diagram used to show the functionality of a system in terms of actors, use cases and the dependencies between them.

Interaction Diagrams:
Communication diagram: shows the interactions between objects. They represent a combination of information taken from Class, Sequence, and Use Case Diagrams describing both the static structure and dynamic behaviour of a system.
Interaction overview diagram: is a type of activity diagram in which the nodes represent interaction diagrams.
Sequence diagram: is used to show how objects communicate with each other, depicted as a sequence of messages. The sequence diagram also presents the lifetime of those messages.
Timing diagrams: Is a specific type of interaction diagram, where the focus is on time limitations.

The problem with classical methods was that they did not allow the display of the dynamics that impact a system, thus key issues were prone to misinterpretations. UML on the other hand offers many options to describe dynamic real-world connection that are part of a modern business environment.

5. UML helps with customer orientation

The UML business technology focuses on business use cases, which empathizes on the connection with the customer. UML business use cases are built from the ground up, and explain how the customer interacts with the business and the way the business interacts with the customer.
For example, below we have a rather simple use case diagram. A customer requests a loan at a bank. Through the use case diagram we can see the entire process and the actors that participate in it. Their actions are depicted through use cases with the necessary dependencies between them.

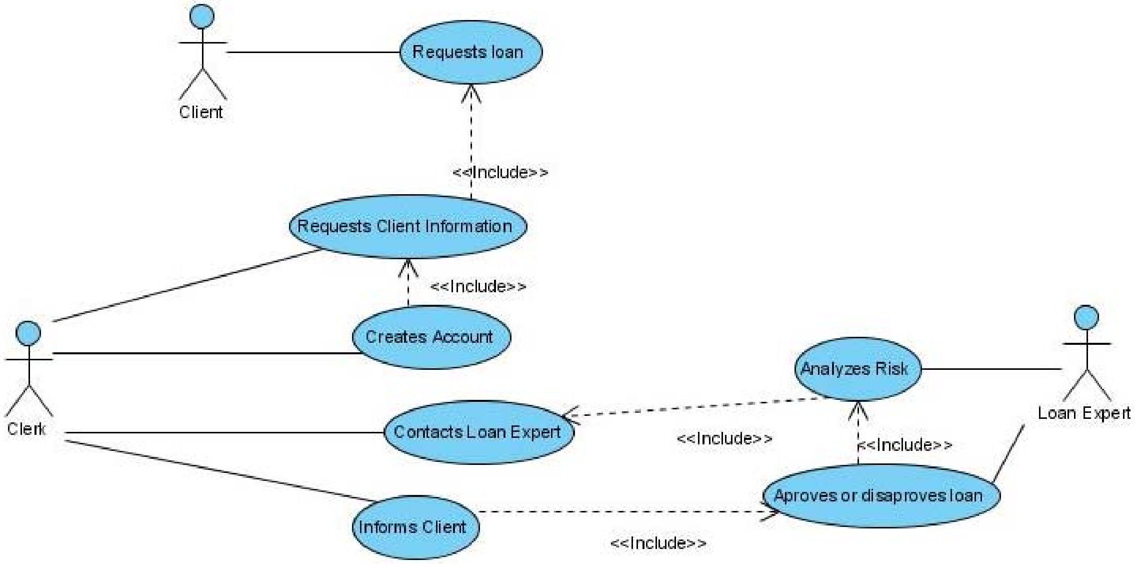


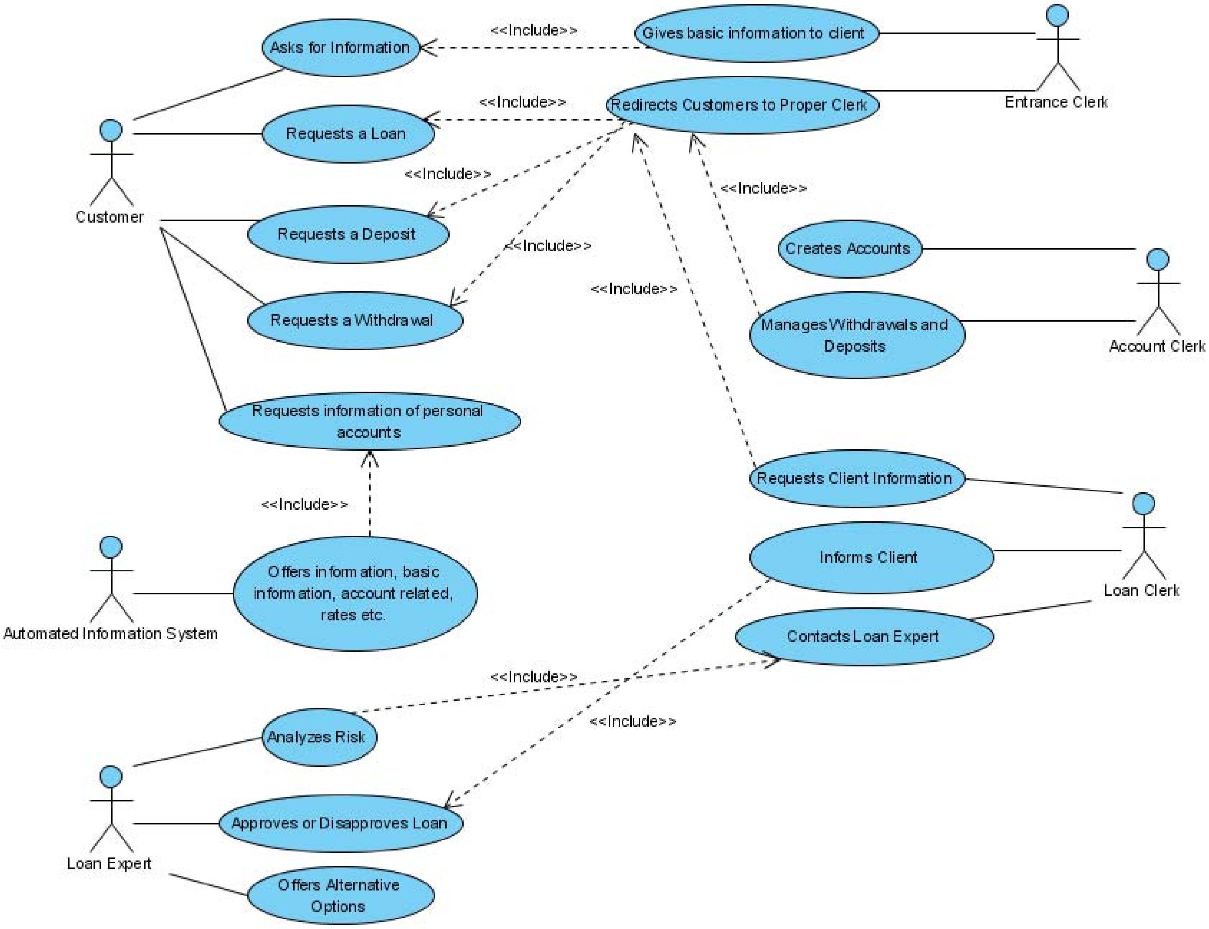
Figure 1: Case diagram

In other words, UML helps better organize and develop business processes, which are the means for delivering value to customers. So the success of an organization hinges on the performance of its business processes (whether they are automated or not).

Case Study 1

In order to prove the use of diagrams in business modelling, let us take the use case diagram from above and develop it further. For example our customer may have different requests other than a loan. A customer may request to make a deposit or a withdrawal, or he might request information on his accounts or rates. Logic leads us to think that one clerk may be overwhelmed by all these tasks. According to that we decide to appoint 3 types of clerks each with specific responsibilities: an entrance clerk to give basic information to clients and to redirect them further to a proper clerk, account clerks that handle the creation of new accounts and the management of existing accounts and also loan clerks that mediate between the loan experts and the clients. Also a good idea may be to implement an automated information system that the customer could use to request information at any time.

We can now see that the use case diagram has taken totally different proportions and represents a more complex structure. We now have six actors instead of the previous three and many more activities with different dependencies and it is still far from representing the complexity of a real-world bank, but with time and effort the use case diagram can eventually cover the structure of any type of business.



Asks for Information
<<Include>>
Gives basic information to client
Entrance Clerk
<<Include>>
Redirects Customers to Proper Clerk
Requests a Loan
Customer
<<Include>>
<<Include>>
Requests a Deposit
<<Include>>
Creates Accounts
<<Include>>
Requests a Withdrawal
Manages Withdrawals and Deposits
Account Clerk
Requests information of personal accounts
Requests Client Information
<<Include>>
Informs Client
Loan Clerk
Offers information, basic information, account related, rates etc.
Automated Information System
Contacts Loan Expert
<<Include>>
<<Include>>
Analyzes Risk
Approves or Disapproves Loan
Loan Expert
Offers Alternative Options

Figure 2: Complex structure of case diagram

Next we might want to go into the depth of a single activity and analyze it in a more complex way. The loan request is a good example. We will first use an activity diagram to represent the flow of the activity and the possible decision nodes in the workflow.

The diagram begins with the customer requesting the loan. The entrance clerk will ask the customer whether he does or does not already own an account. If he does not have an account he is sent to an account clerk, which will create an account for out customer. After that he is sent to a loan clerk. The loan clerk has the responsibility of collecting and verifying the information needed by the loan expert to properly asses the risk of the respective loan. This information is sent to the loan expert who decides whether to approve or disapprove the respective loan, according to the calculated risk. He may also offer an alternative, perhaps a smaller loan more proper to a higher risk. This information is than sent to the loan clerk which informs the client of the loan expert’s decision.

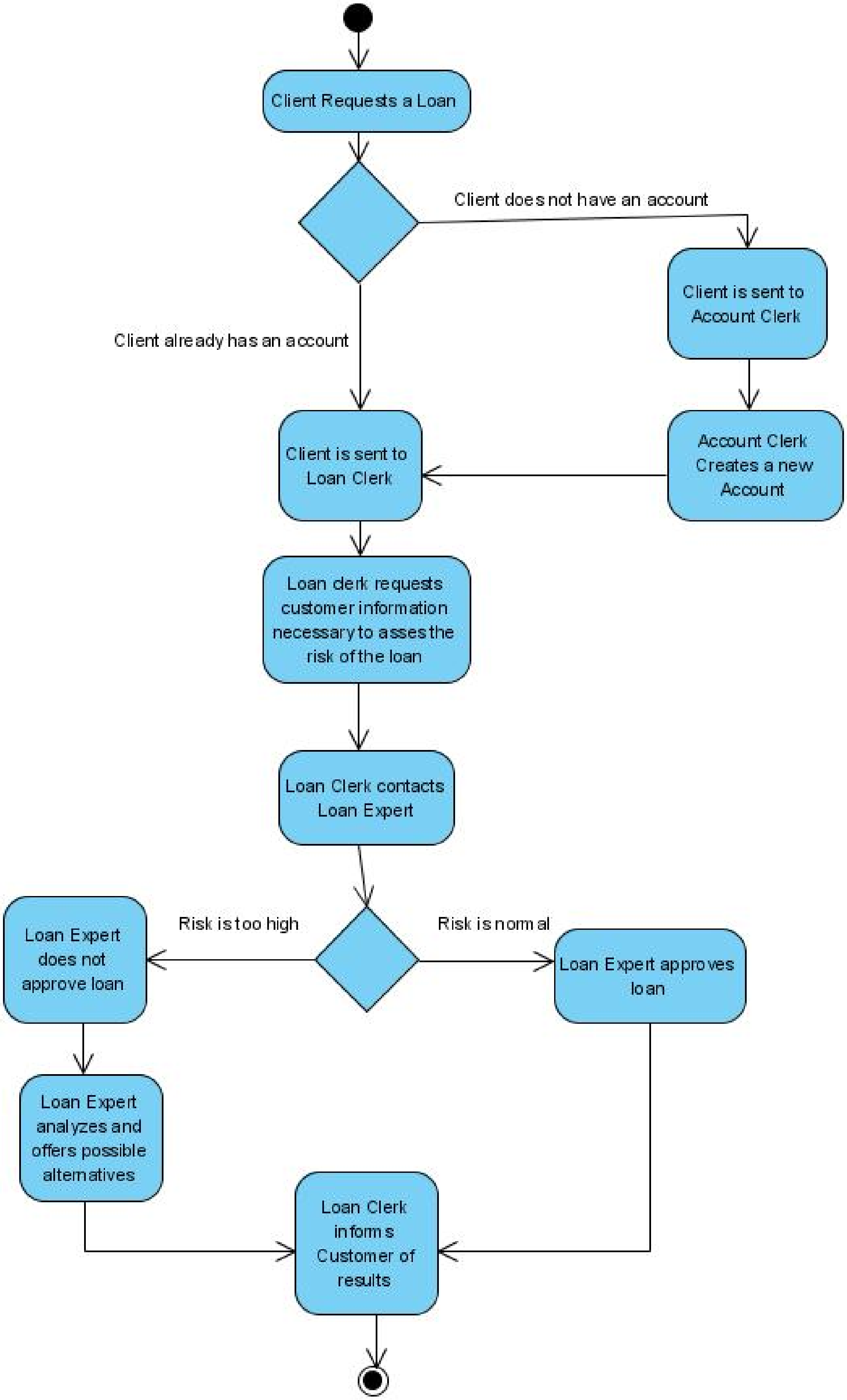

Client Requests a Loan
Client does not have an account
Client is sent to Account Clerk
Client already has an account
Account Clerk Creates a new Account
Client is sent to Loan Clerk
Loan clerk requests customer information necessary to asses the risk of the loan
Loan Clerk contacts Loan Expert
Risk is too high
Risk is normal
Loan Expert does not approve loan
Loan Expert approves loan
Loan Expert analyzes and offers possible alternatives
Loan Clerk informs Customer of results

Figure 3: Diagram of customer requesting

Furthermore we might also want to study the interactions between the objects involved in this activity. For this we may use a sequence diagram that can also show us the time life of objects and the order of the messages between the objects.

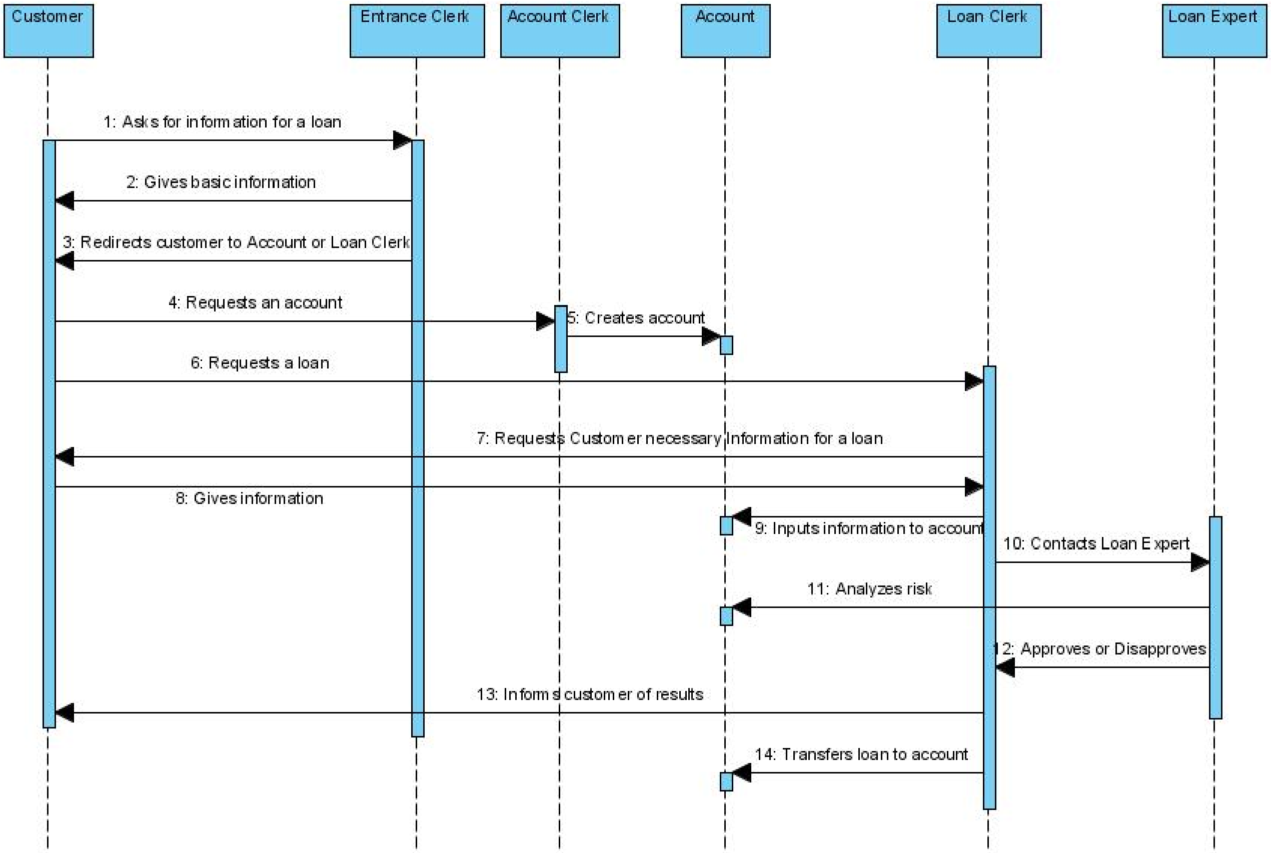

Customer
Entrance Clerk
Account Clerk
Account
Loan Clerk
Loan Expert
1: Asks for information for a loan
2: Gives basic information
3: Redirects customer to Account or Loan Clerk
4: Requests an account
5: Creates account
6: Requests a loan
7: Requests Customer necessary Information for a loan
8: Gives information
9: Inputs information to account
10: Contacts Loan Expert
11: Analyzes risk
12: Approves or Disapproves
13: Informs customer of results
14: Transfers loan to account

Figure 4: Diagram time life of objects

Case Study 2

On the second case study we will study the usefulness of UML diagrams in decision making and in dealing with exceptional circumstances. The next activity diagram represents a fictive “transport of merchandise to customer” diagram. In this business process the company also takes the responsibility of providing the transport. In this context the company has two options that depend on whether or not the transport is by land only, they either contact company Y that is presumably cheaper but cannot offer transport overseas or Contact Company X that provides quick delivery but charge more. After that is decided the merchandise is passed on to the transport company with the attached invoice. Than on arrival the customer may or may not accept the merchandise, if the merchandise is defective it is send back to the company for replacement. Of course if the customer accepts the merchandise, they are expected to make the respective payment.

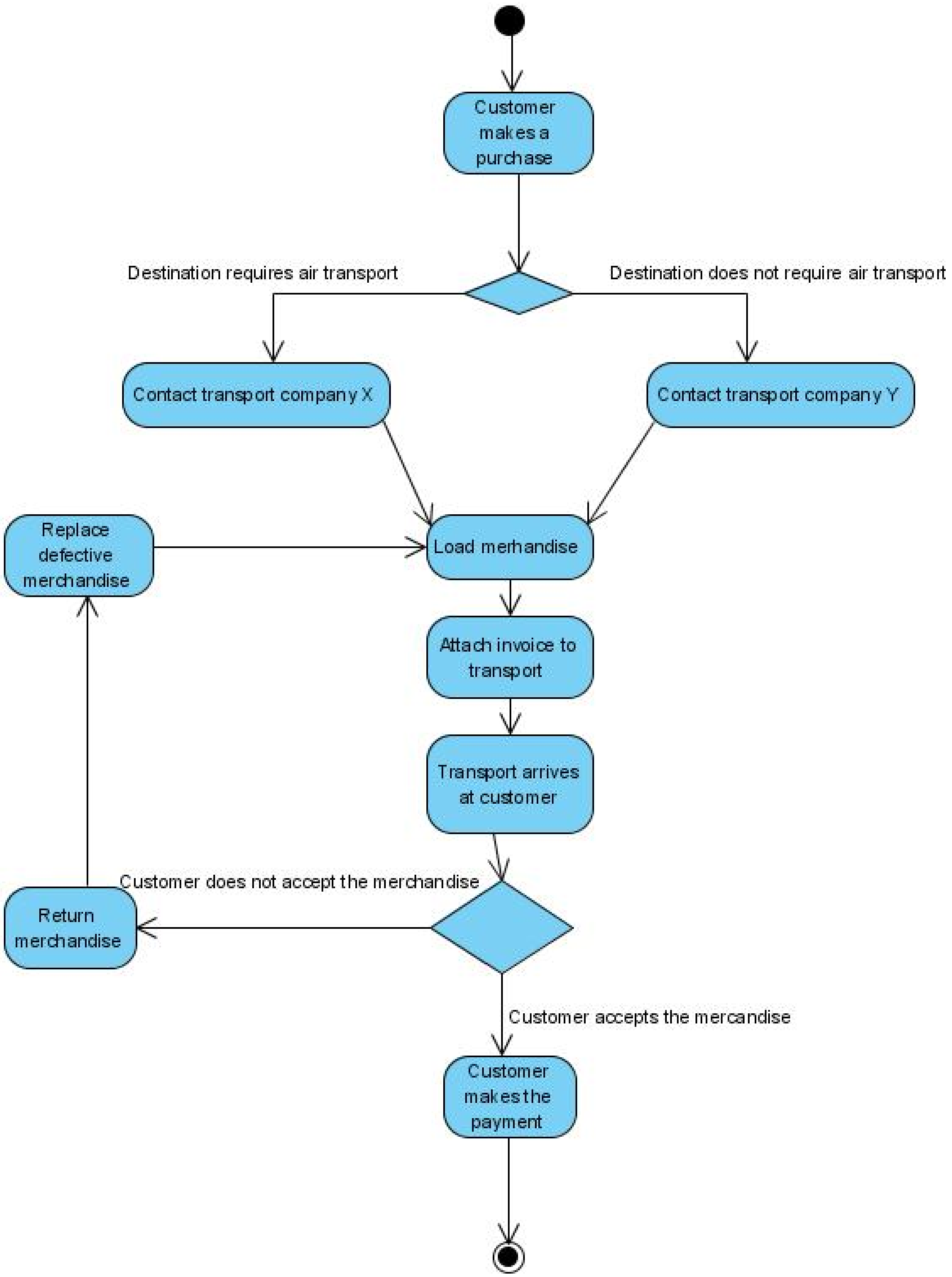

Customer makes a purchase
Destination requires air transport
Destination does not require air transport
Contact transport company X
Contact transport company Y
Replace defective merchandise
Load merhandise
Attach invoice to transport
Transport arrives at customer
Customer does not accept the merchandise
Return merchandise
Customer accepts the mercandise
Customer makes the payment

Figure 5: "Transport of merchandise to customer" diagram

At one point when the merchandise arrives at the customer, there is an error in the invoice, and by law the merchandise despite intact cannot be accepted, so following the process the entire transport is sent back to the provider. On this account with the help of the activity diagram the process is reinvented this time the invoice is sent by fax only after the customer accepts the merchandise. The sacrifice of this decision is that the company needs to more keep in contact with the customer until the transport arrives so that they may send the invoice. The new process is represented by the second activity diagram:

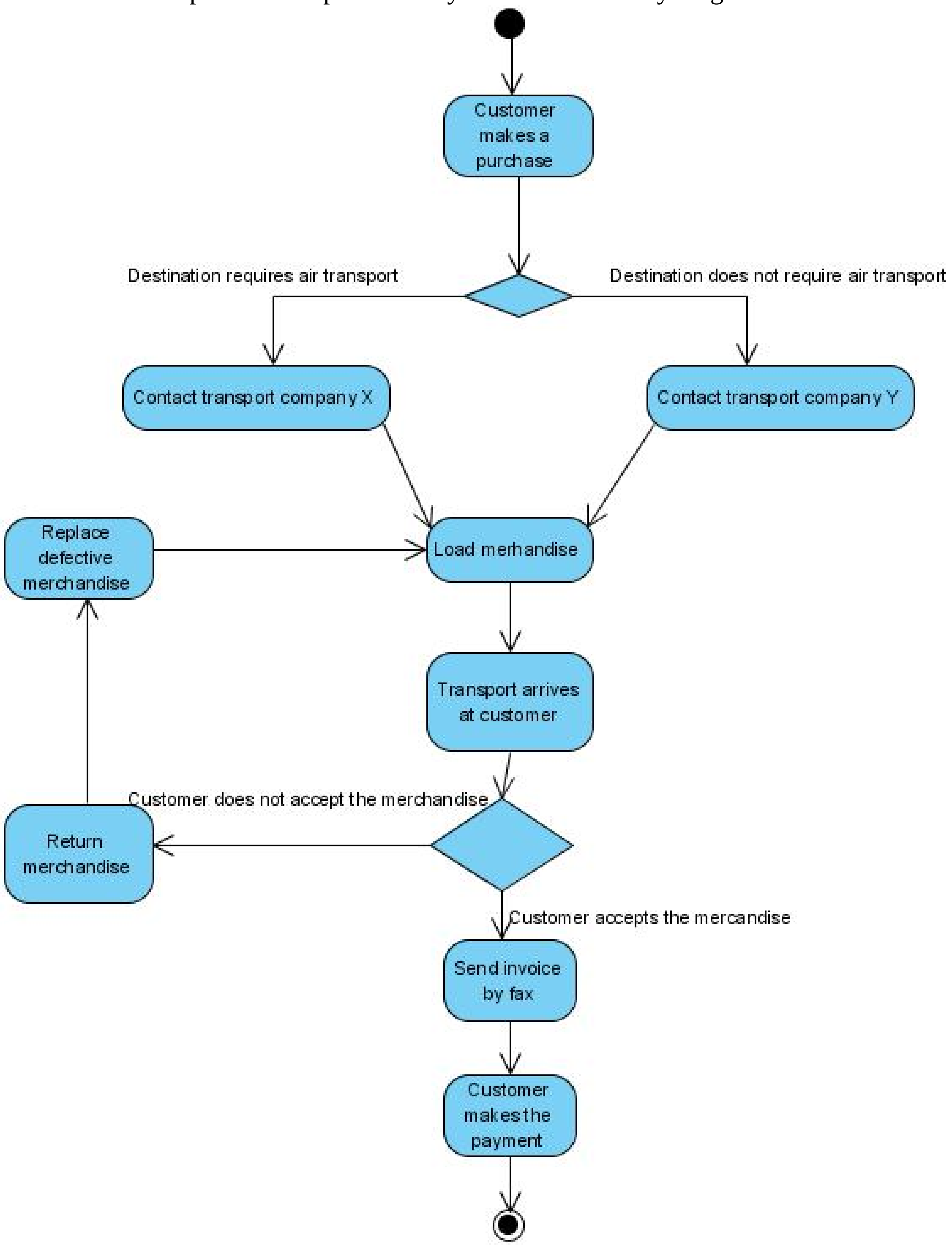

Customer makes a purchase
Destination requires air transport
Destination does not require air transport
Contact transport company X
Contact transport company Y
Replace defective merchandise
Load merhandise
Transport arrives at customer
Customer does not accept the merchandise
Return merchandise
Customer accepts the mercandise
Send invoice by fax
Customer makes the payment

Figure 6: Activity diagram

Criticism

Despite all the praise brought to the UML language and all the positive views on UML in business modelling, there are also those who criticize both UML as a modelling language and UML used in business development.

- Firstly UML is seen by many as a **hard to learn** or even incomprehensible modelling language, many software developers find it hard to understand and some do not even know the basics of UML. In practice, people often draw diagrams with the symbols provided by their CASE tool, but without the meanings of those symbols.
- **Linguistic incoherence**. The extremely poor writing of the UML standards themselves -- assumed to be the consequence of having been written by a non-native English speaker -- seriously reduces their normative value. In this respect the standards have been widely cited, and indeed pilloried, as prime examples of unintelligible geek-speak.
- **UML and implementation language mismatch**. UML is an object oriented modelling language, thus a business developer might tend toward solutions that reside at the intersection of the capabilities of UML and the implementation language. This problem is particularly poignant if the implementation language does not adhere to orthodox object-oriented doctrine, as the intersection set between UML and implementation language may be that much smaller.
- **UML is too abstract**. Some developers claim that UML is sometimes too abstract and that sometimes during the development process they might raise and stretch the abstraction level beyond real world possibilities or that they might make the software much more complex than actually necessary.
- **UML is not necessary for all businesses, only for complex automated business systems.**

Conclusion

Personally I find UML to be a very useful tool both in the development of software and in the modelling of a business because of the necessity to visualize the business in its development and because it is a good way to identify possible problems and even solve them before they become a threat. Despite all the criticism brought to it I believe that UML in business administration is more of an asset than a liability.

Bibliography

http://en.wikipedia.org/wiki/Unified_Modeling_Language
Bryon Baker - Business Modelling with UML: The Light at the End of the Tunnel
Hans-Erik Eriksson and Magnus Penker - Business Modelling with UML