

Activity Diagrams

What it is and How to Use

Author: Maria Ericsson

Email: mariae@rational.com

1 Introduction

In its basic form, an activity diagram is a simple and intuitive illustration of what happens in a workflow, what activities can be done in parallel, and whether there are alternative paths through the workflow.

Activity diagrams as defined in the Unified Modeling Language [UML1.3] are derived from various techniques to visually illustrate workflows, see for example [Johansson]. Much of the basis for the definition of the activity diagram notation is found in [Martin & Odell].

In the Rational Unified Process [RUP], we talk about how you can use activity diagrams to visualize the workflow of a business use case. A complete workflow description will have a basic flow, and one or several alternative flows. This workflow has a structure that we can define textually, using informal if, if-then-else, or do-until statements of various kinds. For a simple workflow with a simple structure such textual definitions may be quite sufficient, but in the case of more complex structures, activity diagrams help to clarify and make more apparent what the workflow is.

Historically, activity diagramming techniques have mostly been used in the business process modeling domain, but this article will also briefly discuss how you can use it in the system modeling domain.

2 Basic Activity Diagram Notation

As common for most notations, the activity diagram notation has some elements that are necessary for you to understand if you want to be “conversant” about activity diagrams. Those elements are presented in this section. Next section on Advanced Notation talks about additional goodies you may or may not find use for. A basic activity diagram can have the following elements:

- *Activity states*, which represent the performance of a step within the workflow.
- *Transitions* that show what activity state follows after another. This type of transition is sometimes referred to as a completion transition, since it differs from a transition in that it does not require an explicit trigger event, it is triggered by the completion of the activity the activity state represents.

- *Decisions* for which a set of *guard conditions* are defined. These guard conditions control which transition of a set of alternative transitions that follows once the activity has been completed. You may also use the decision icon to show where the threads merge again. Decisions and guard conditions allow you to show *alternative threads* in the workflow of a business use case.
- *Synchronization bars*, which you can use to show parallel subflows. Synchronization bars allow you to show *concurrent threads* in the workflow of a business use case.

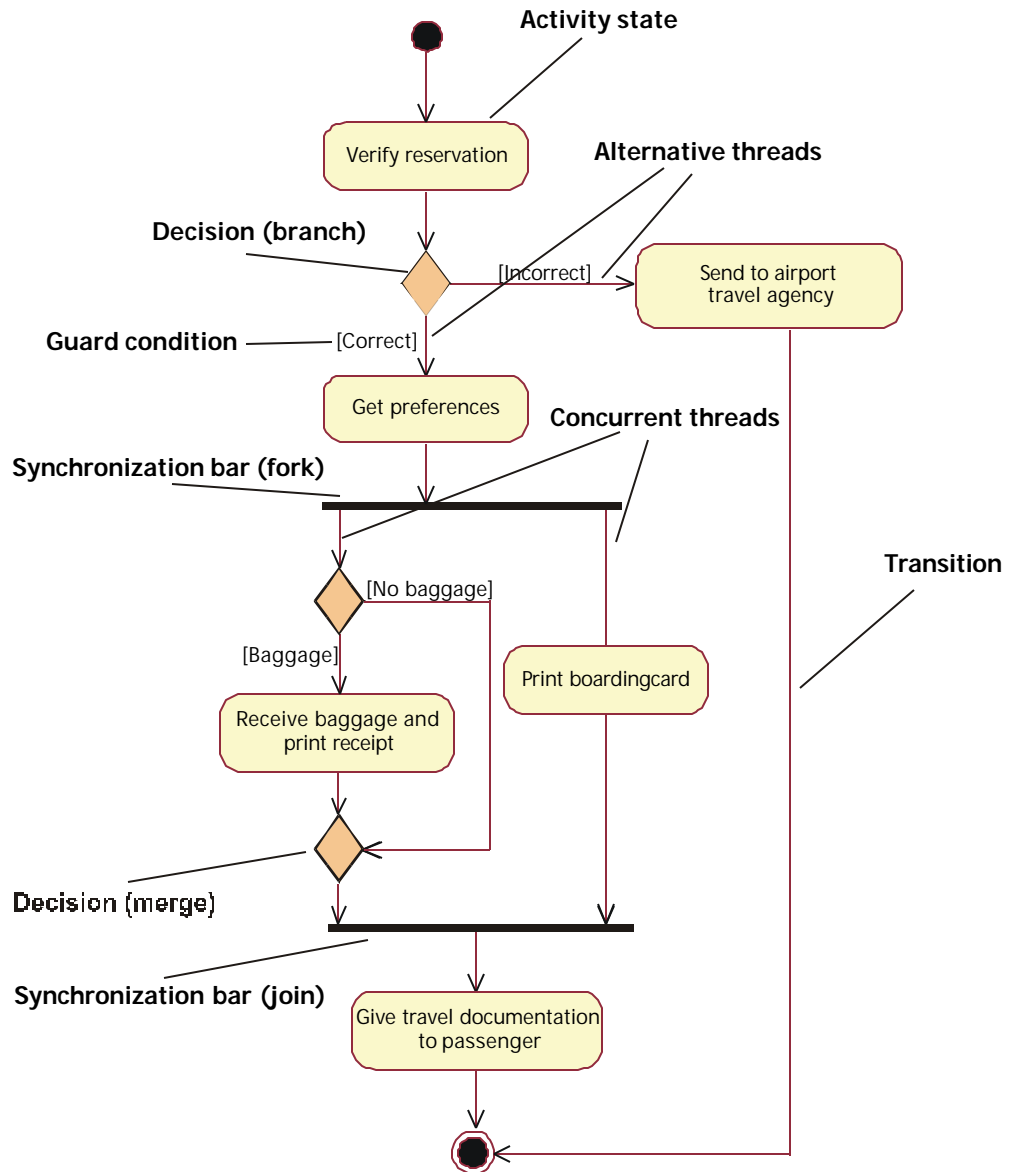


Figure 1. An activity diagram for the business use case Individual Check-In in the business use-case model of airport check-in.

3 Advanced Notation

In more complex examples, you would often make use of the following constructs:

- Conditional threads
- Nested activity diagrams
- Partitions

3.1 Conditional Threads

In section 2, we said that you describe which one of a set of alternative threads to follow by defining guard conditions for each thread.

Guard conditions can also be used to show that one of a set of concurrent threads is conditional. For example, in the individual check in example from figure 1, the passenger checking in might be a frequent-flyer member. In that case, you need to award the passenger frequent flyer miles.

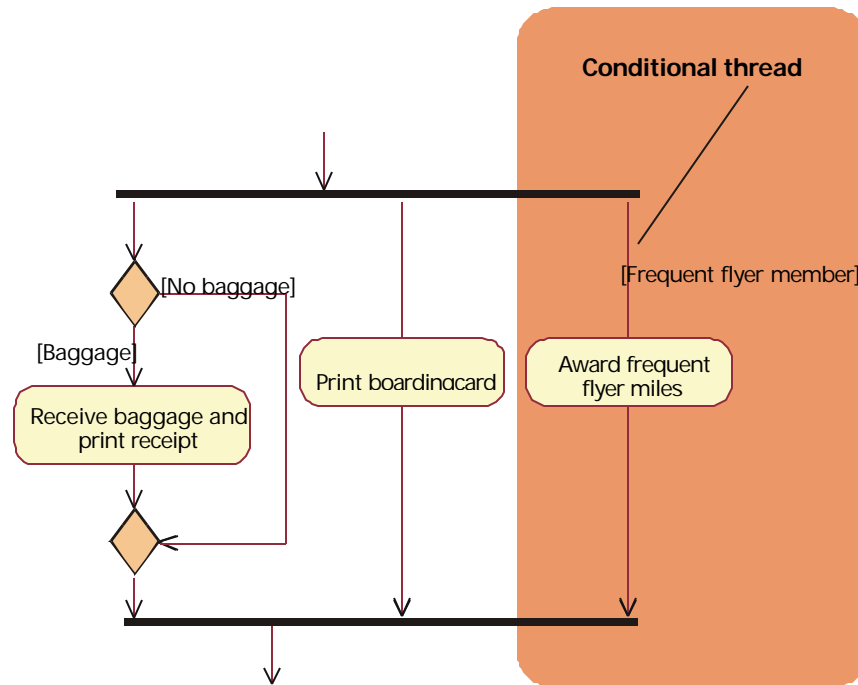


Figure 2. Awarding frequent flyer miles is a conditional thread in the Individual Check-In workflow.

3.2 Nested Activity Diagrams

An activity state may reference another activity diagram, which shows the internal structure of the activity state. Another way to say this is that you can have nested activity graphs. You can either show the sub-graph inside of the activity state (figure 3), or let the activity state refer to another diagram (figure 4).

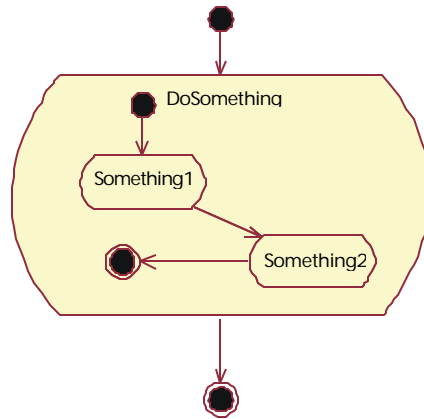


Figure 3. A nested activity graph shown within an activity state.

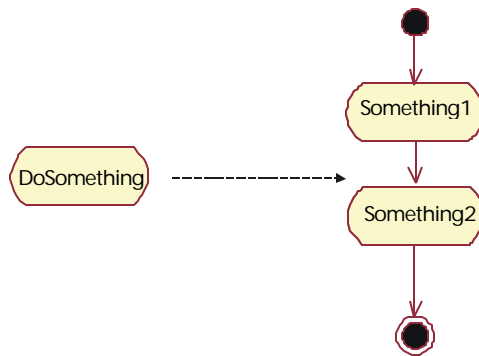


Figure 4. Alternatively, put the subgraph in a separate diagram and let the activity state refer to it.

3.3 Partitions

The contents of an activity diagram may be organized into *partitions* using solid vertical lines. A partition does not have a formal semantic interpretation, but is in business modeling often used to represent an organizational unit of some kind.

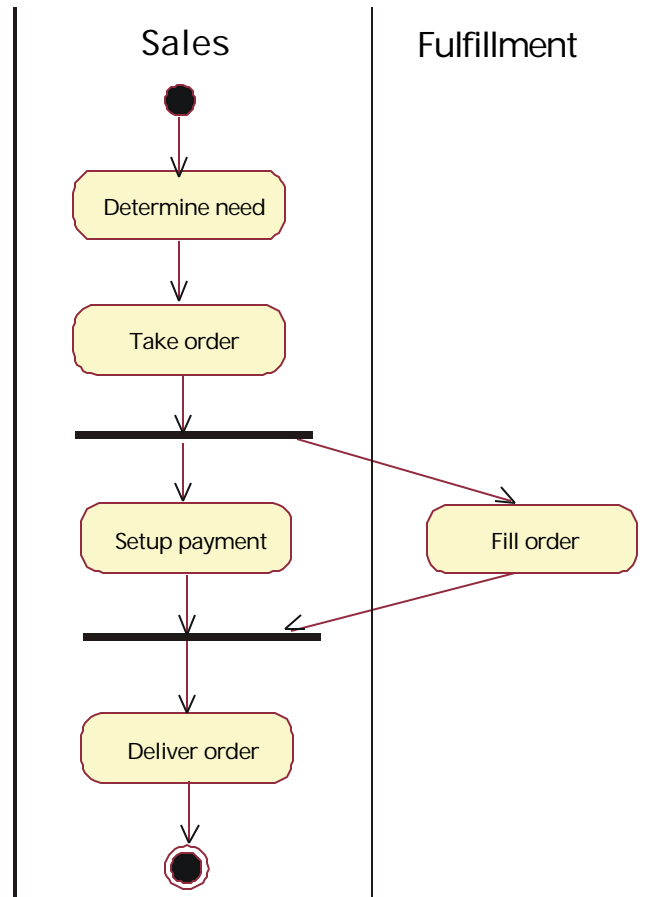


Figure 3. An activity diagram illustrating the workflow of a business use case that represents a (generic) sales process. In this example, the partitions represent departments in the organization.

4 Documenting Business Use Cases

Which comes first, the activity diagram or the textual description of the workflow? This is somewhat dependent how you are used to working, and whether you “think graphically” or not. Some prefer to outline the structure visually in a diagram first, and then develop the details in the text. Others rather start with a bulleted list of the activity states first and agree on those (like a step-by-step outline to the use case), then define the structure using a diagram.

A very valid question is also whether you really need both the textual document and the diagram.

To understand what an activity diagram adds to the understanding of a workflow, we present a sample workflow description, and then an activity diagram for that workflow. This example is a proposal process, taken from an organization that sells telecom network solutions, individually configured to each customer. We have simplified the example by removing the detailed text in most of the subsections, but tried to keep enough so that you can understand the structure of the workflow. The full text of this example can be found in [RUP].

Basic Workflow

1.1 Initial Contact

This process starts with an initial contact between the Customer and The Company. This may happen in one of the following ways:

< details excluded >

1.2. Initial Opportunity Work

< details excluded >

1.2.1 Gather Preliminary Customer Requirements

< details excluded >

1.2.2 Create Sales Plan (optional)

< details excluded >

1.2.3 Perform Opportunity Analysis

< details excluded >

1.3. Create Proposal Project Plan

< details excluded >

1.4. Create Delivery Project Plan

< details excluded >

1.5. Prepare a Quote

< details excluded >

1.6. Compile Additional Information

< details excluded >

1.7. Analyze and Finalize the Proposal

< details excluded >

1.8. Present the Proposal

< details excluded >

1.9. Obtain Customer Decision

< details excluded >

2. Alternative Workflows

2.1. Business Opportunity Rejected

If in 1.2., it turns out the business opportunity is rejected, the following actions may be taken:

< details excluded >

2.2. Unable to Meet Customer Requirements

If in Perform Opportunity Analysis or Prepare a quote, The Company is unable to suggest a solution to the customer requirements, then the following actions may happen:

< details excluded >

2.3. Critical Information Not Known

If at any point in the Proposal Process The Company identifies some critical information not known or available then he does one of the following:

< details excluded >

2.4. New/Incomplete or Incorrect General Customer Profile

If The Company determines that the general customer profile is inaccurate for some reason, the following actions may be taken.

< details excluded >

An activity diagram for the workflow could look as follows. We use basic notation only in this diagram. Activity states correspond to sections in the workflow description:

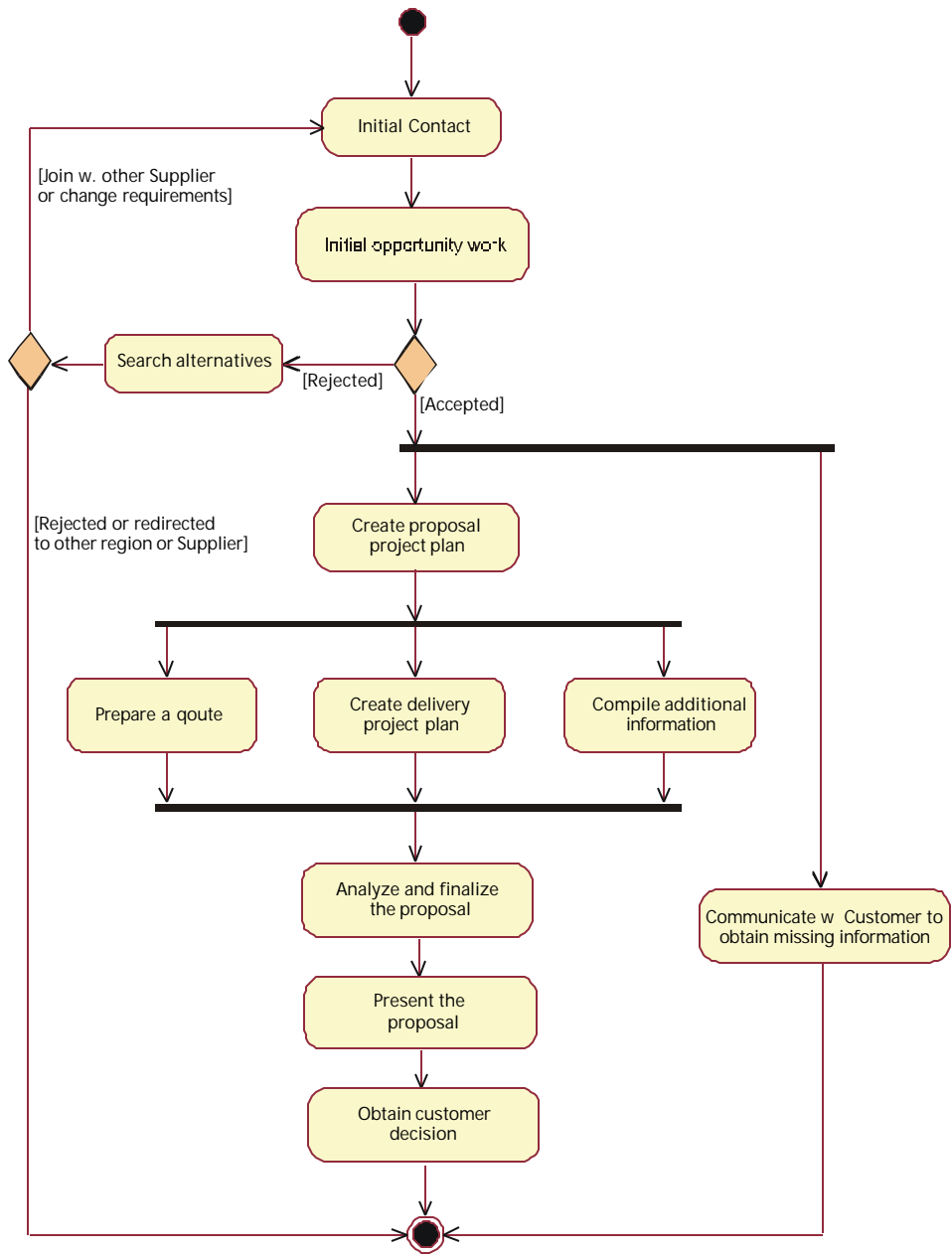


Figure 4. An activity diagram for the business use case Proposal Process.

The activity state 'Initial opportunity work' consists of three sub-steps that can be done in parallel. This is illustrated in a sub-graph to this activity state, see figure 5.

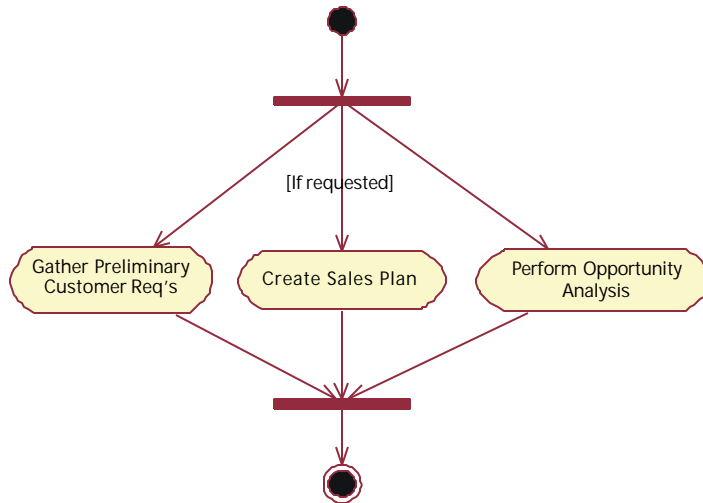


Figure 5. Sub-diagram to the activity state 'Initial opportunity work'. Creating a sales plan is optional, which is indicated by a guard condition on the incoming transition.

5 Documenting Business Use-Case Realizations

You are documenting business use-case realizations, rather than business use cases, if you are using partitions and the partitions are coupled to classes (business workers mainly) in the business object model.

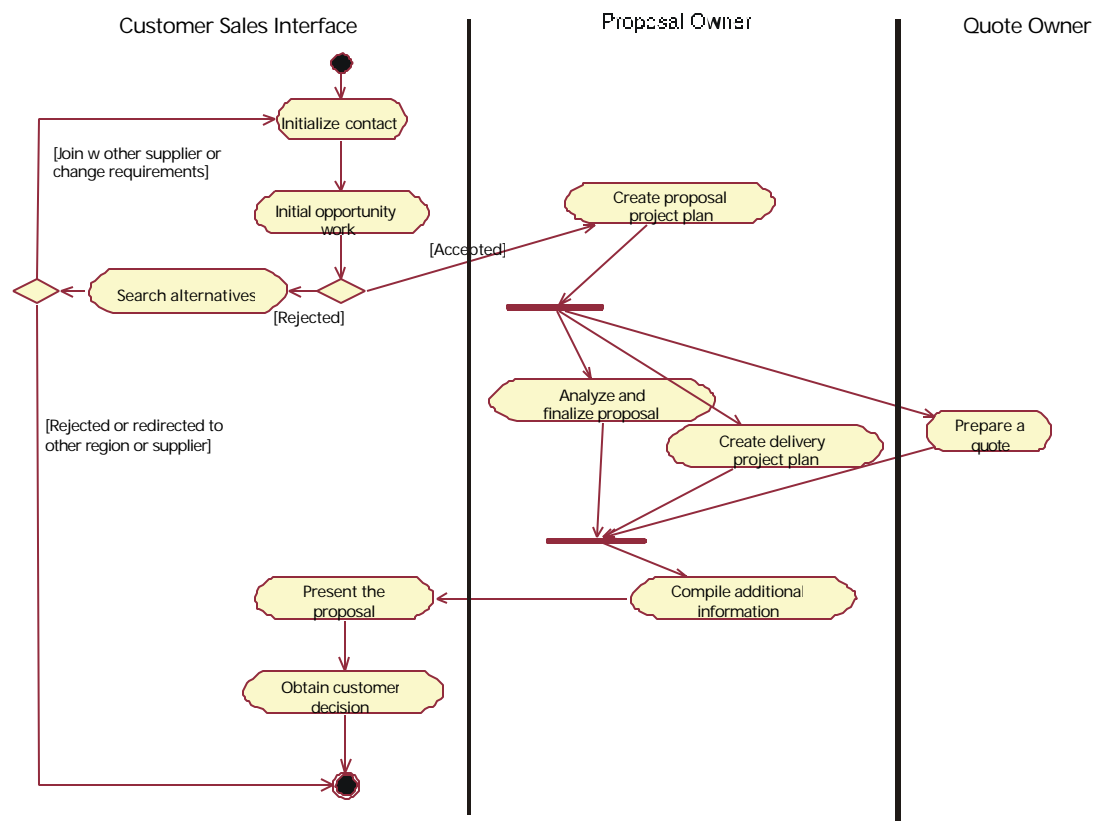
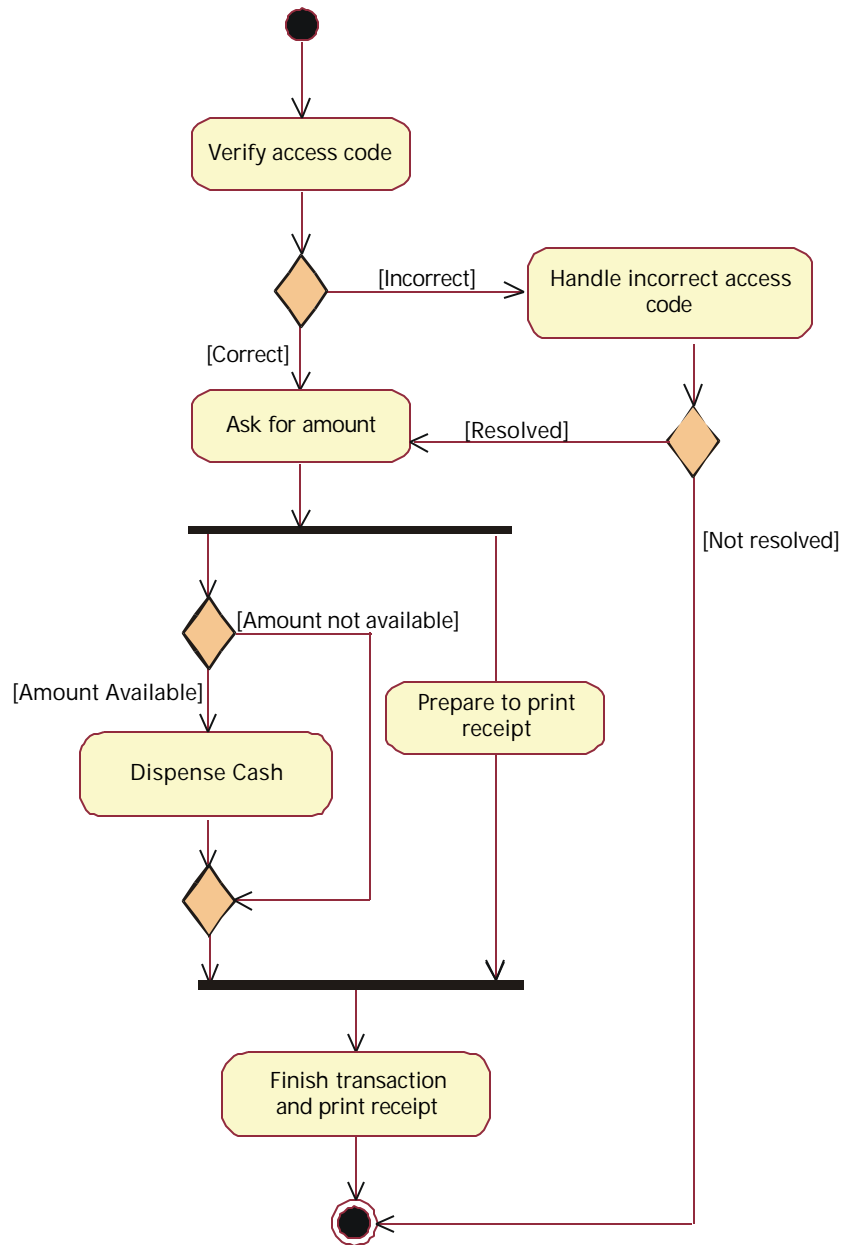


Figure 6. The same workflow as presented in figure 5, but here the activities are organized in partitions.

Compared to a sequence diagram, which could be perceived to have a similar purpose, an activity diagram with partitions focuses on how you divide responsibilities onto classes, while the sequence diagram help you understand how objects interact and in what sequence.

6 Just for Business Modeling?

Just like you would use an activity diagram to show the structure of a workflow, you could also use it to show the structure of a flow of events of a system use case.



A simplified activity diagram for the use case Withdraw Money in the use-case model of an automated teller machine (ATM).

In the first stages of identifying objects and classes based on the use cases (use-case analysis), activity diagrams can be useful when exploring responsibilities of analysis classes. You might use the activity diagram technique to draw a first sketch of class responsibilities, a sketch that you then throw away.

7 Discussion

Activity is defined as an ongoing non-atomic execution within a state machine. A structure that has a duration with possible interruption points. Activities can be modeled by nested states, by a submachine reference, or by an activity expression.

Activity state represents the execution of a pure computation with substructure. The normal use of an activity state is to model a step in the execution of a procedure or algorithm.

An activity can represent a fairly large procedure (with some substructure), as well as something relatively small. In the RUP, we have chosen to focus on describing how you can use activities and activity diagrams to describe use cases and use-case realizations, but the concept itself as defined in the UML is more general than that.

Action states are like activity states, except they are atomic. An action state cannot have substates, and it cannot be terminated by an external event. Action states are intended for short bookkeeping operations, while activity states are for computations of any duration or complexity. Action states have no special notation. They can be shown as a state with an entry action, or as an activity state.

8 References

[RUP] – *Rational Unified Process* version 5.1.1

[UML1.3] – *OMG UML Specification v. 1.3 alpha R2 draft* 1999, available at <http://uml.shl.com/>.

[UMLref] – Jim Rumbaugh, et al. 1998. *UML Reference Manual*. Addison Wesley Longman.

[Jacobson] – Ivar Jacobson, Maria Ericsson, and Agneta Jacobson 1994. *The Object Advantage - Business process reengineering with object technology*. Addison-Wesley.

[Johansson] – H. Johansson, P McHugh, J. Pendlebury, and W. Wheeler III 1993. *Business Process Reengineering. Breakpoint Strategies for Market Dominance*. Chichester: John-Wiley and Sons.

[Martin & Odell] Martin, J. and Odell, J. 1996. *Object Oriented Methods: a Foundation, the UML Edition*. Englewood Cliffs, NJ: Prentice Hall.