

Activity Diagrams: What They Are and How to Use Them

Maria Ericsson



Maria Ericsson is Senior Process Developer at Rational Software.

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 Language¹ are derived from various techniques to visually illustrate workflows, see for example². And much of the basis for the definition of the activity diagram notation is found in³.

In the Rational Unified Process⁴, 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.

The purpose of this paper is to show how you can use activity diagrams within the Rational Unified Process for business modeling as well as system modeling. Activity diagrams are often mentioned almost as a synonym to business modeling. For a more complete introduction to what business modeling is we refer to⁵, and for details to⁶.

The reader of this article is assumed to be familiar with the basics of the Unified Modeling Language.

Basic Activity Diagram Notation

As is 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. The next section talks about additional goodies you may find useful. A basic activity diagram can have the following elements: ([Figure 1](#))

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 can be referred to as a completion transition. It differs from a transition in that it does not require an explicit trigger event; it is triggered by the completion of the activity which 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 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.

Advanced Notation

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

- Conditional threads
- Nested activity diagrams
- Partitions

Conditional Threads

Guard conditions can be used to show that one of a set of concurrent threads is conditional. For example, in the individual check-in example from [Figure 2](#), the passenger checking in might be a frequent-flyer member. In that case, you need to award the passenger frequent flyer miles.

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](#)).

To show the sub-graph inside of the activity state is convenient if you want to see all details of the workflow in one and the same diagram. But if there is any level of complexity in the workflow presented, this can make the diagram hard to read.

To simplify the workflow graph, you may instead choose to put the sub-graph in a separate diagram, and let the activity state sub-graph details refer to that diagram.

Partitions

The contents of an activity diagram may be organized into *partitions* (swimlanes) 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 5](#)).

Documenting Business Use Cases

Background: A business use-case model describes the processes of a business and their interactions with external parties like customers and partners. The processes of the business are represented as business use cases, and the external parties are represented as business actors.

Describing a business use case includes among other things giving it a name, a brief description, defining its performance goals, and its workflow. The most time-important and time-consuming aspect to describe is the workflow.

Which comes first, the activity diagram or the textual description of the workflow? This is somewhat dependent on

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 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 valid question is also whether you really need both the textual document and the diagram. The activity diagram technique does allow you to write brief descriptions of each activity state, which should make the textual specification of the workflow obsolete. Here you need to be sensitive to your audience and the format in which they expect the specification.

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 ([Figure 6](#)). 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 in⁴.

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 Customerz 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 >

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 it 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 is shown in [Figure 6](#). We use basic notation only in this diagram. Activity states correspond to sections in the workflow description:

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 7](#).

An activity state can represent a fairly large procedure (with substructure), as well as something relatively small. If you are using activity diagrams to define the structure of a workflow, you should not attempt to explore several levels of activity graphs down to their most "atomic" level. This will most probably make the diagram (or set of diagrams, if you are using separate sub-graphs) very hard to interpret. You should aim at having one diagram that outlines the whole workflow, where a few of the activity states have sub-graphs.

Documenting Business Use-Case Realizations

Background: A business use-case realization describes how a particular business use case is realized within the business object model, in terms of collaborating business workers and business entities. A business worker represents a set of responsibilities typically carried by one individual. A business entity represents a "thing" that is created, managed, or used. The realization of a business use case can be described textually, but is more commonly explained with diagrams - collaboration diagrams, sequence diagrams, activity diagrams, or a combination. Which diagram type you choose depends on the complexity of the workflow and also where in the process you are.

You are using the activity diagram to document 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 8](#)).

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 helps you understand how objects interact and in what sequence. Activity diagrams give focus on the workflow, while sequence diagrams give focus on the handling of business entities. Activity diagrams and sequence diagrams could be used as complementary techniques, where a sequence diagram shows what happens in an activity state.

Just for Business Modeling?

Background: The use-case model is a model of a system's intended behaviors. A use case tells the story of how a user (represented as an actor in the model) can use the system to achieve a particular purpose. Describing a use case among other things includes giving it a name, a brief description, and defining the flow of events of the use case.

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 ([Figure 9](#)). 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.

Summary

This paper has given you an overview of:

- Basic and more advanced elements of the activity diagram notation. Basic elements of activity diagrams are activity states, transitions, decisions, and synchronization bars.
- How activity diagrams allow you to show concurrent threads, and alternative threads, as well as conditional threads in a workflow.
- How you can use activity diagrams in business modeling. You can illustrate the workflow of a business use case. You can describe how a business use case is realized by business workers and business entities.
- How you can use activity diagram in system modeling. You can illustrate the flow of events of a use case. You can define how a use case is realized by analysis classes.

About the Author

Maria Ericsson is Senior Process Developer at Rational Software, part of the team that develops the Rational Unified Process. She started working in the field of software engineering and object technology in 1990 at Objectory AB, and co-authored Ivar Jacobson's *The Object Advantage — Business Process Re-engineering with Object Technology*. She has just recently relocated back to her roots in Sweden after four and a half years with Objectory and Rational in the USA. She can be reached at mariae@rational.com.

References

1. *OMG UML Specification v. 1.3 alpha R2 draft*. 1999. Available at <http://uml.shl.com/>.

2. Johansson, H.; McHugh, P.; Pendlebury, J.; and Wheeler, W. III. 1993. *Business Process Reengineering. Breakpoint Strategies for Market Dominance*. Chichester: John Wiley and Sons.
3. Martin, J. and Odell, J. 1996. *Object Oriented Methods: a Foundation, the UML Edition*. Englewood Cliffs, NJ: Prentice Hall.
4. *Rational Unified Process* version 5.1.1
5. Kruchten, P. 1998. *The Rational Unified Process — an Introduction*, Addison Wesley Longman.
6. Jacobson, Ivar; Ericsson, Maria; and Jacobson, Agneta. 1994. *The Object Advantage — Business Process Reengineering with Object Technology*. Addison-Wesley.

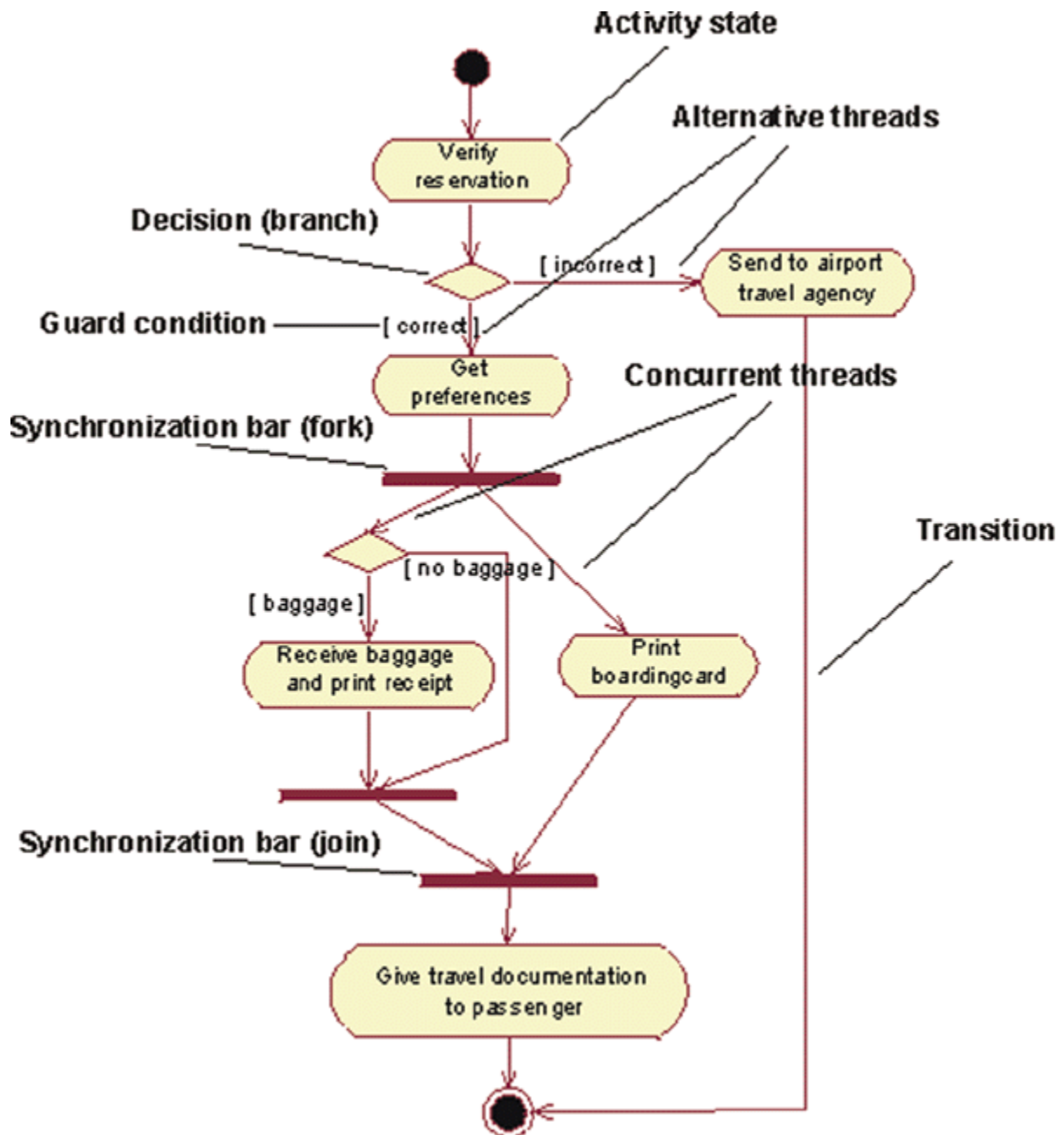


Figure 1: An activity diagram for the business use case Individual Check-In in the business use-case model of airport check-in.

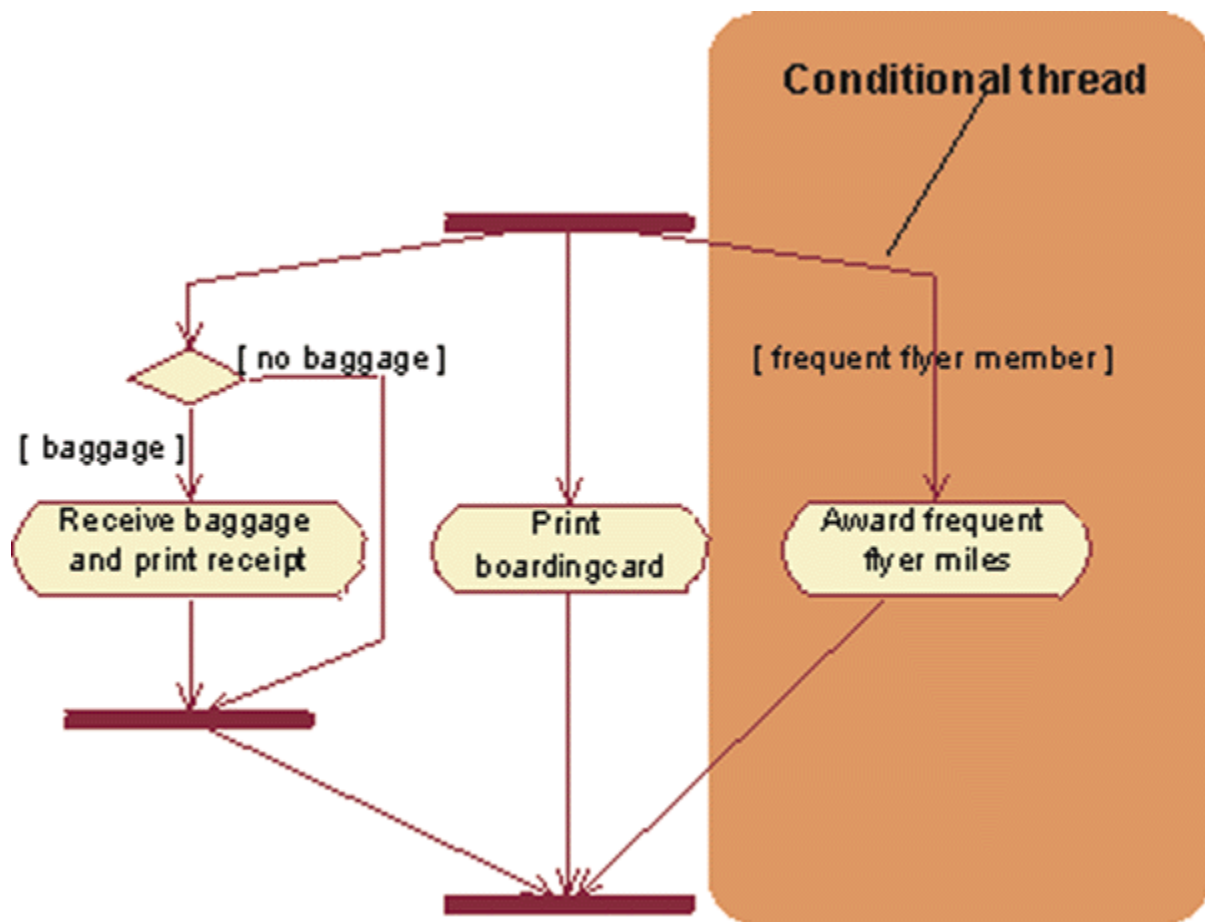


Figure 2: Awarding frequent flyer miles is a conditional thread in the Individual Check-In workflow.

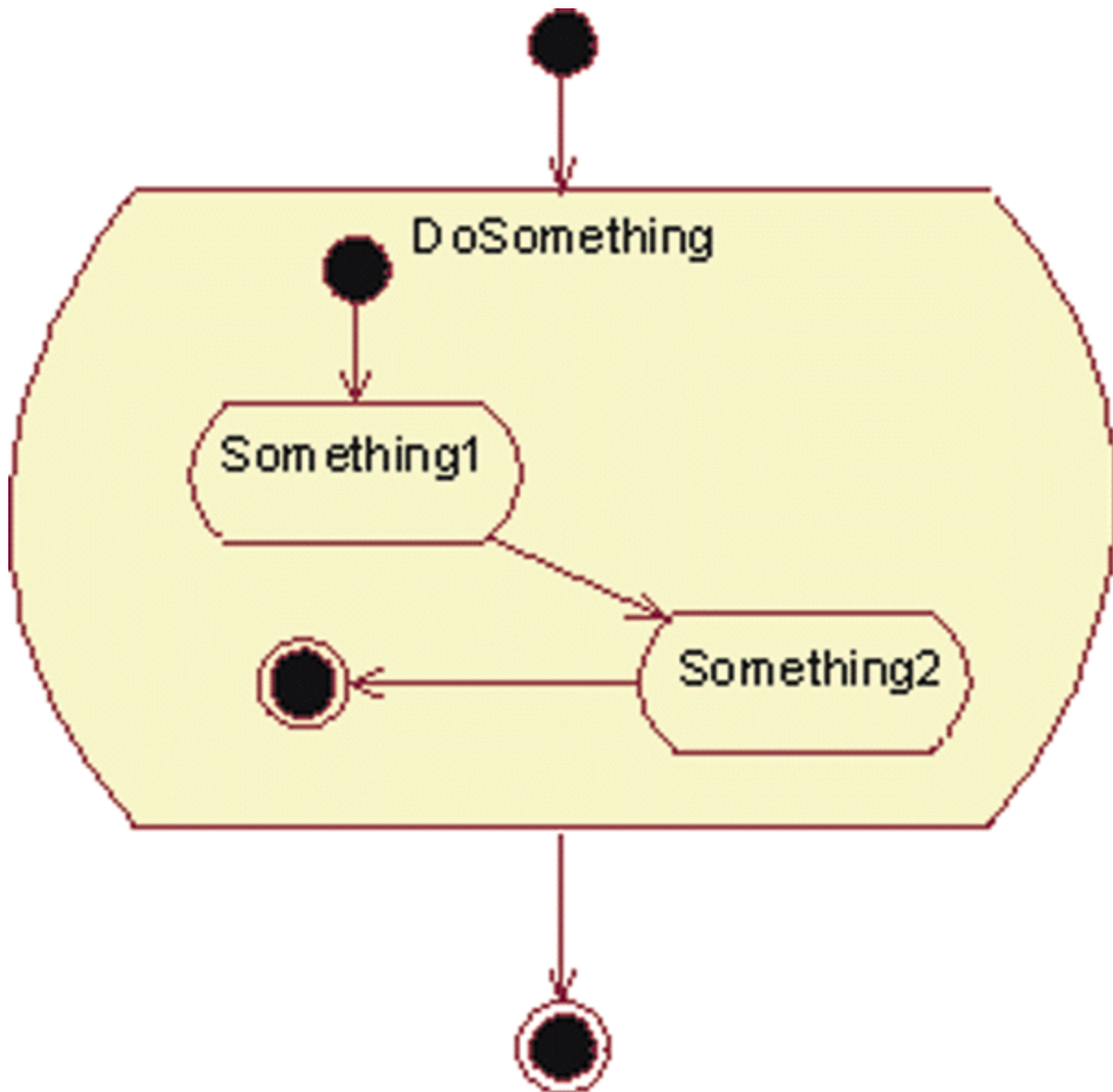


Figure 3: A nested activity graph shown within an activity state.

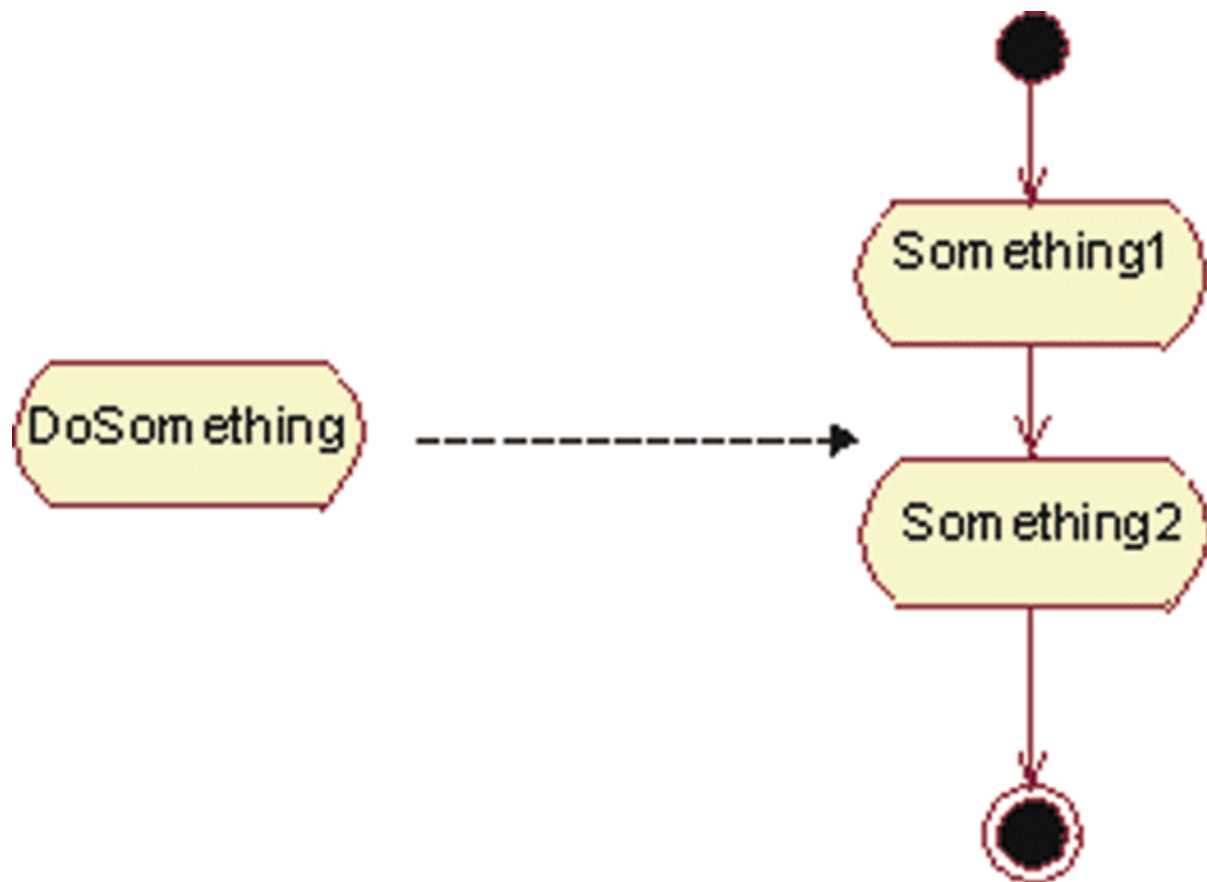


Figure 4: Alternatively, put the sub-graph in a separate diagram and let the activity state refer to it.

Sales

Fulfillment

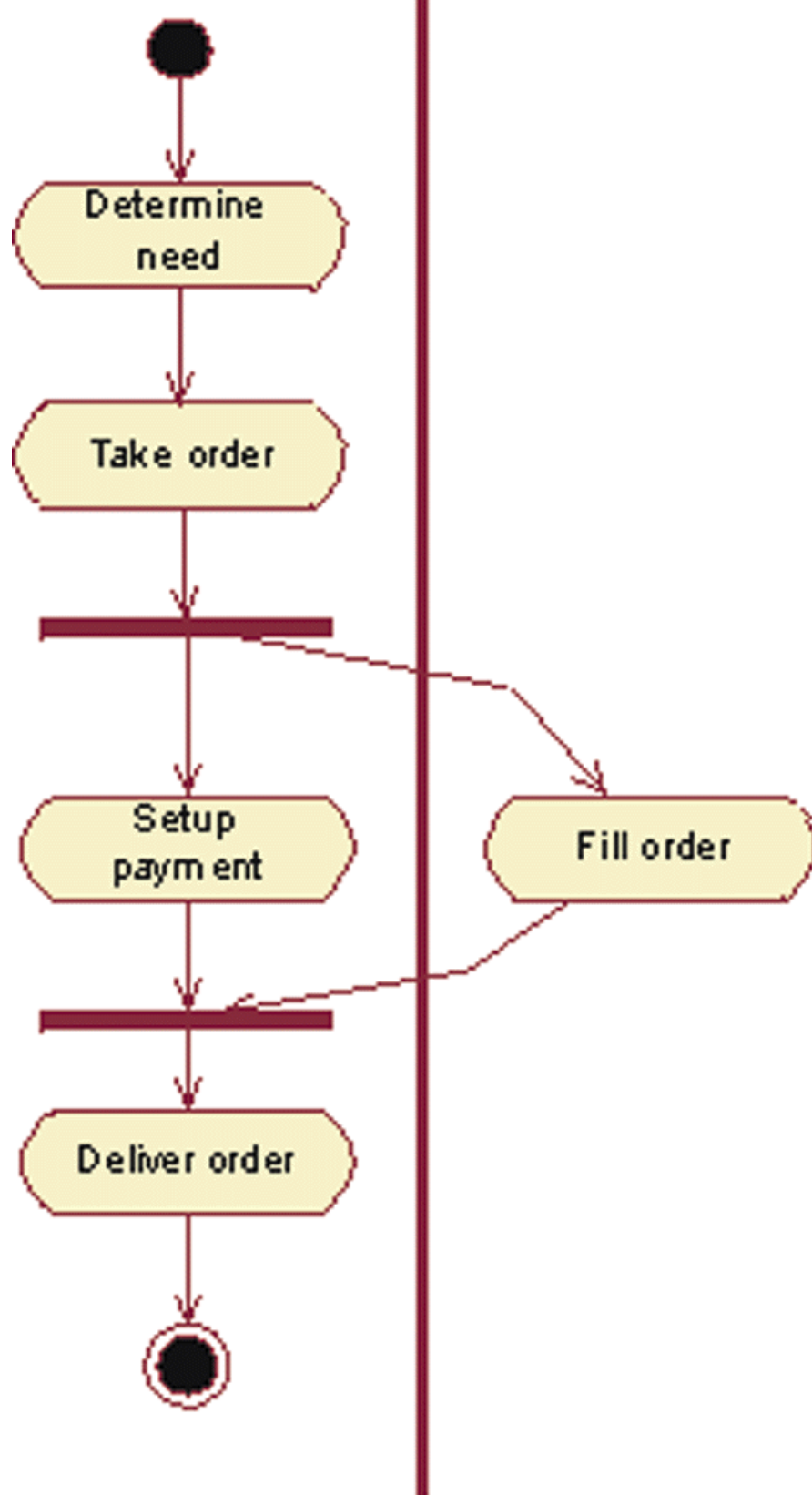


Figure 5: An activity diagram illustrating the workflow of a business use case that represents a (generic) sales process. In this example, the partitions represent departments in the organization.

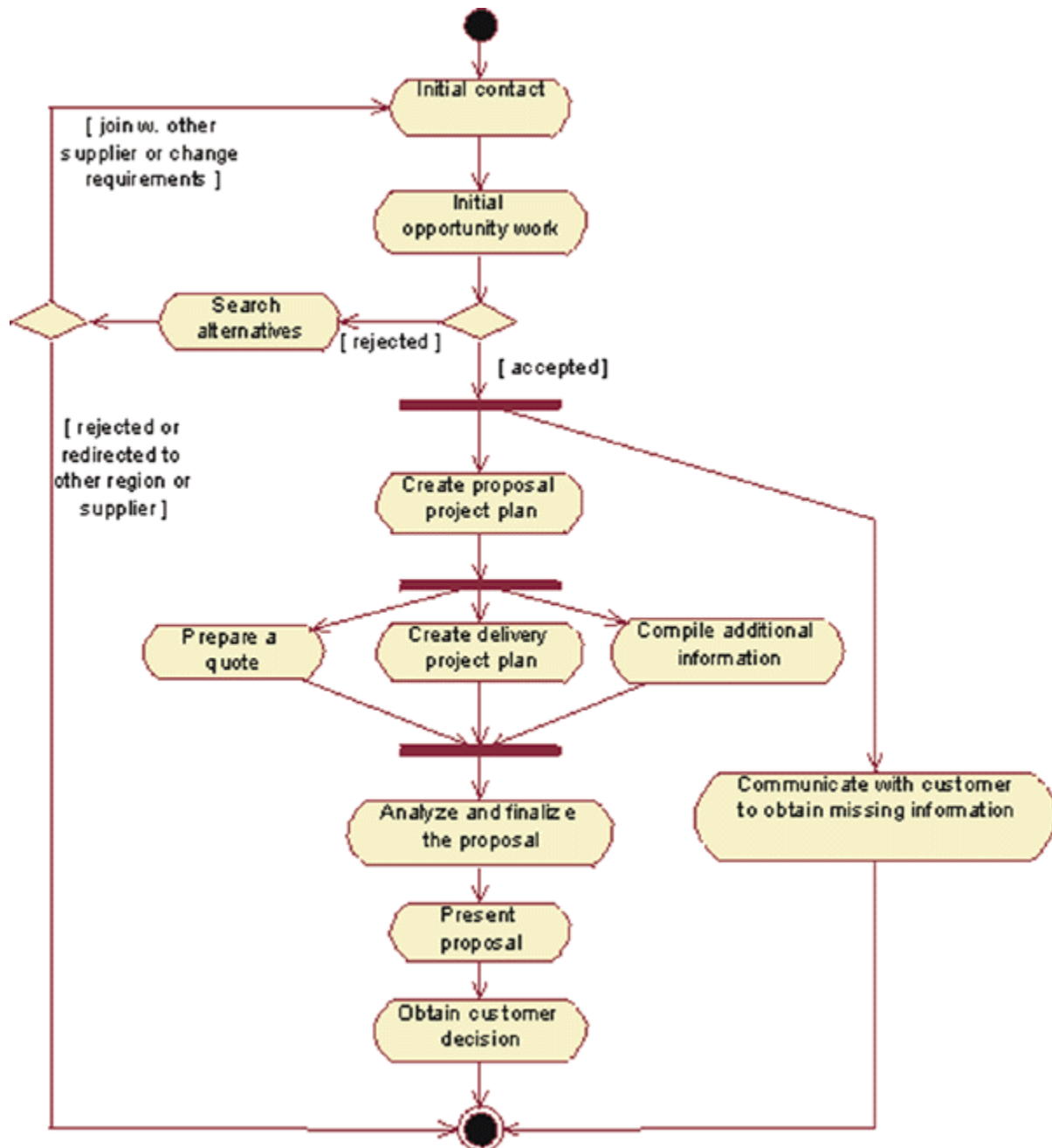


Figure 6: An activity diagram for the business use case Proposal Process.

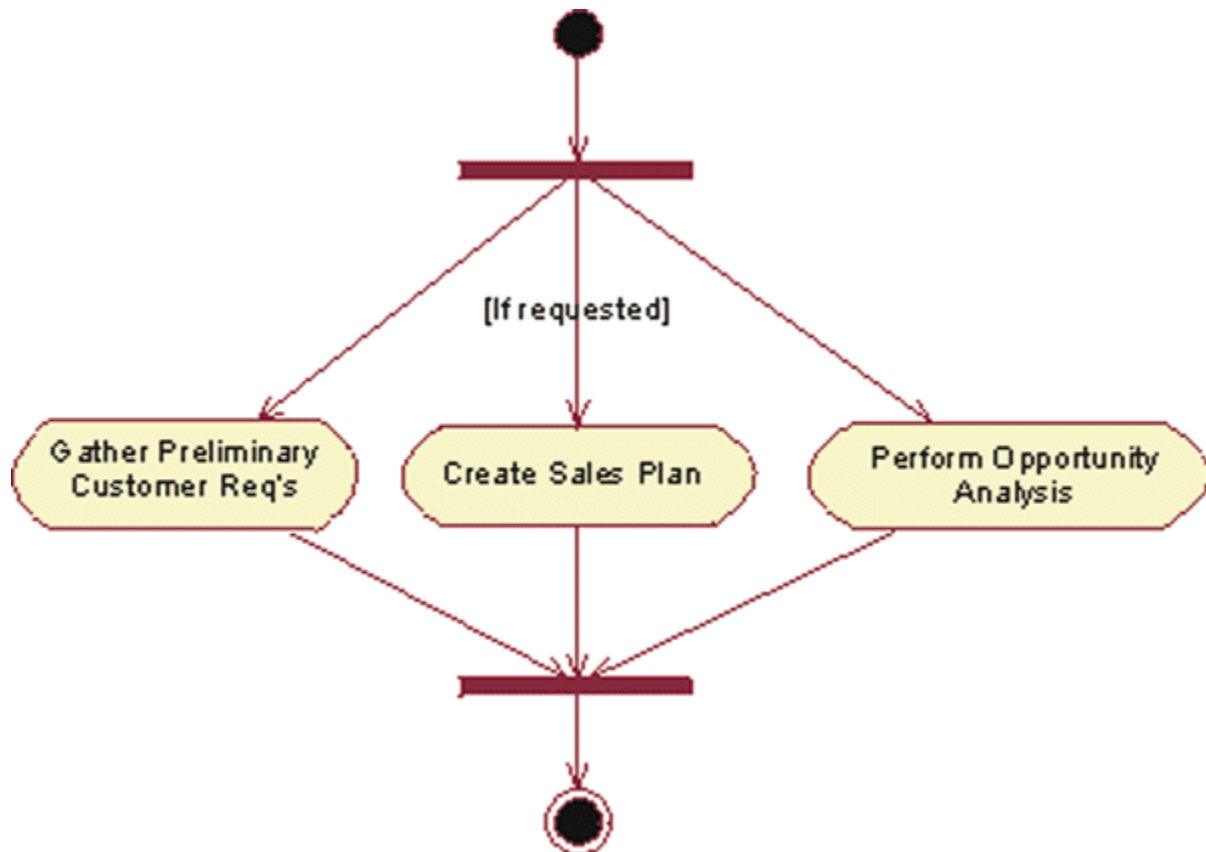


Figure 7: Sub-diagram to the activity state 'Initial opportunity work'. Creating a sales plan is optional, which is indicated by a guard condition on the incoming transition.

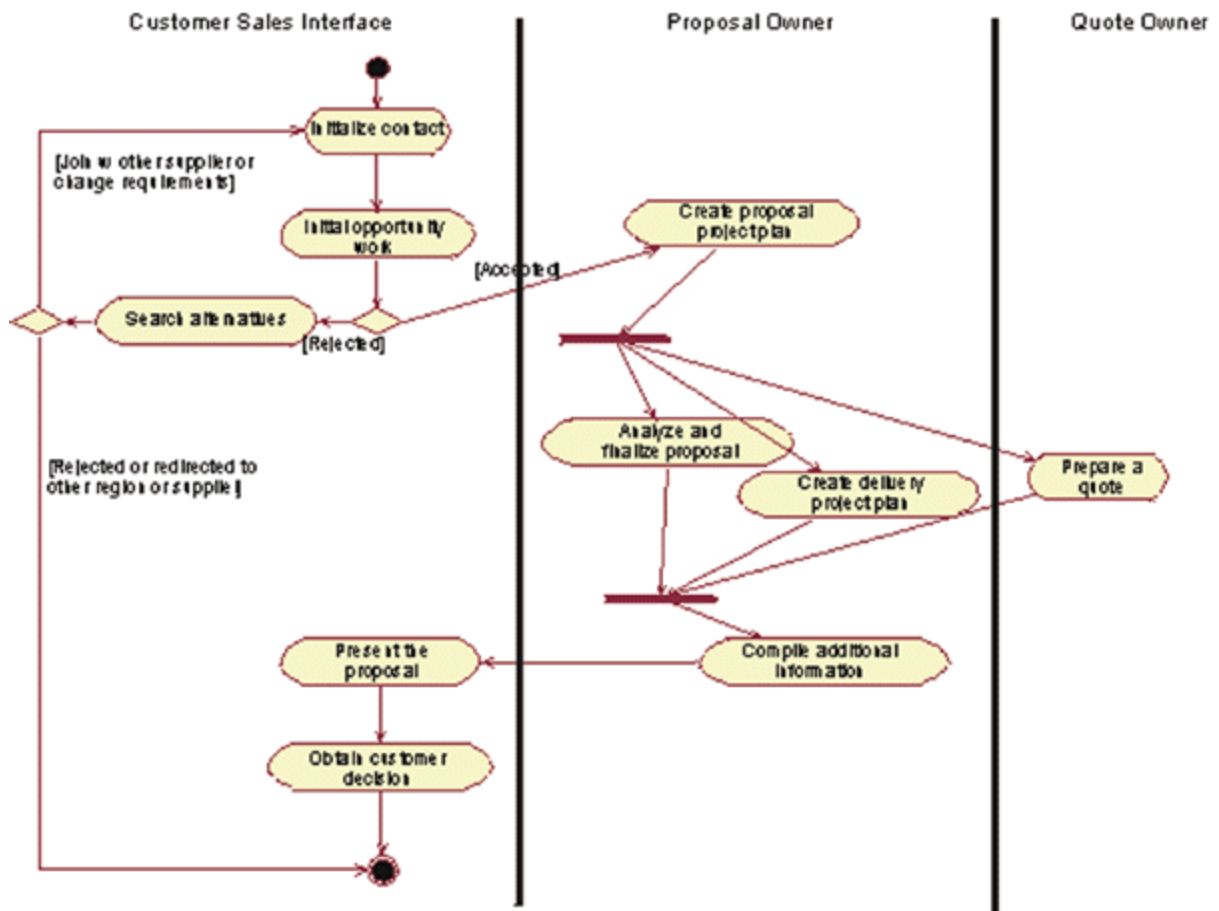


Figure 8: The same workflow as presented in Figure 6, but here the activities are organized in partitions.

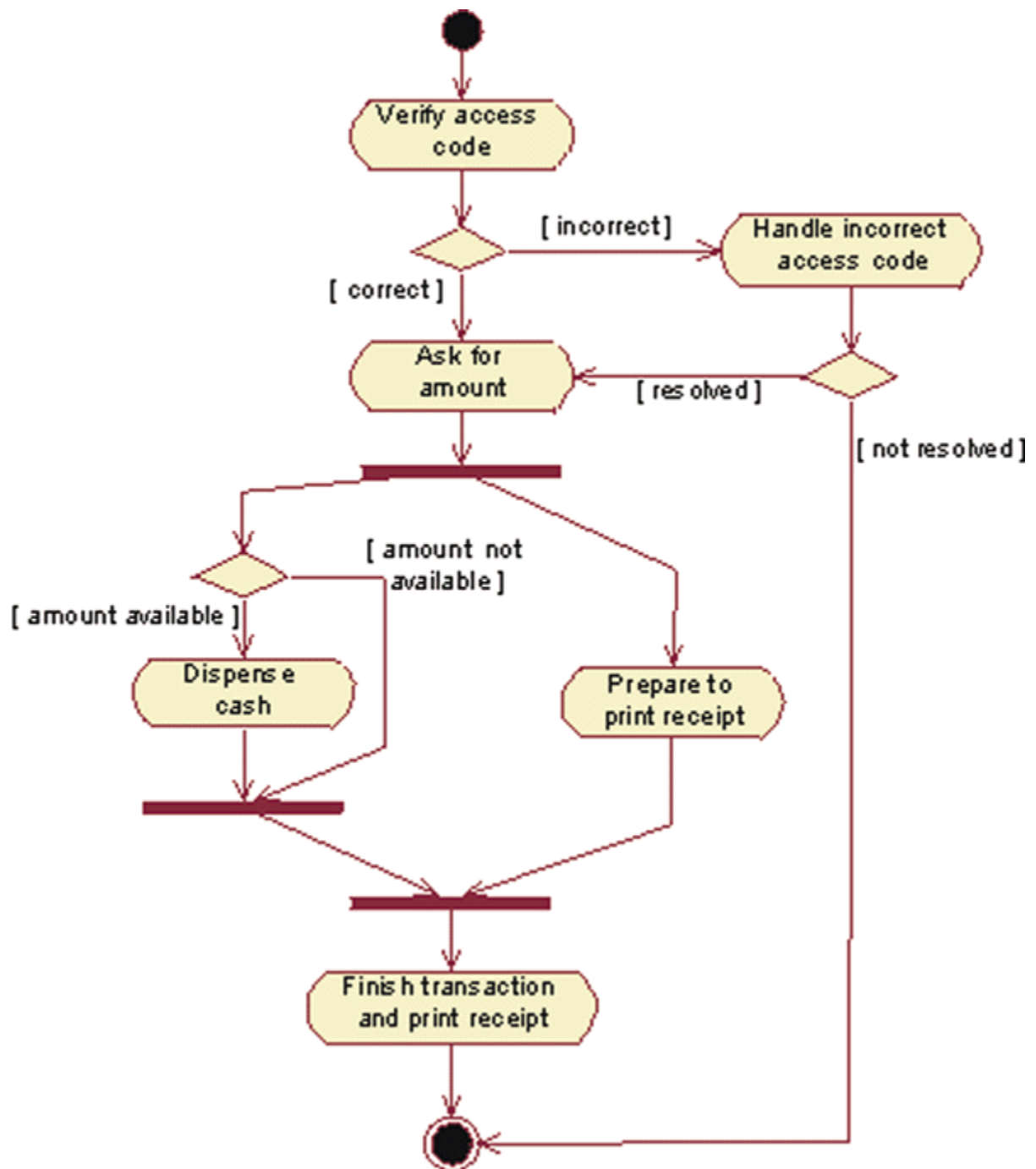


Figure 9: A simplified activity diagram for the use case Withdraw Money in the use-case model of an automated teller machine (ATM).