Class 13

Learning Objectives

- Understand control-flow patterns in workflows
- Be able to model a control-flow in YAWL

Readings

This class is the first in a series of seven (13 to 20) which will introduce you to the control-flow, resource, and data-flow perspective of a workflow model. The control-flow is concerned with defining the tasks that make up a workflow, the order in which they are to be executed, and possible branching and joining points in the workflow. Control-flow is essentially Petri-net and workflow net based (which you covered in the previous classes), although the YAWL graphical diagrams have some more advanced options.

In the following classes, we will define the resource and data perspective for workflows. Resources are typically people in the organization and the roles they play. The resource perspective is concerned with who is allowed to or who is supposed to carry out the tasks in a workflow. Finally, the data perspective deals with the information that is associated with each workflow instance. For example, a workflow that makes a travel reservation will include data about the traveller, flights, hotels, billing and accounting, and authorization. This data will need to be captured in the workflow.

All seven classes (13 to 20) rely on the same set of readings: Chapters 2, 3, 8, and 10 of the textbook, and the support reading by Bratosin (2009) on defining data structures for YAWL. We will try to guide your reading to focus on the topic of each chapter, but you will likely end up reading some material multiple times, as it is hard to really separate the control-flow, resource, and data perspectives.

Chapter 2 introduces the YAWL graphical notation (language) and uses examples shown in the YAWL editor. Chapter 8 focuses specifically on the functionality of the editor, it serves almost as a user manual for the editor. You may read these chapters in either order. When reading chapter 8 first, it becomes clearer how the language constructs in chapter 2 will actually be modeled. Reading chapter 2 first makes it easier to understand what is being demonstrated in the YAWL editor in chapter 8.

At the end of the notes for this class you will find a brief introduction to the YAWL software. You should install this first, before doing the readings, as you'll need the software for the review exercises.

You may also wish to look at the following tutorial video on the course web-site:

- Video Y1: YAWL Editor Control Flow 1

Chapter 2

Chapter 2 is the core chapter of the textbook (and of the entire course) and you should read it carefully and understand it thoroughly. It is a very long chapter but you have already read Section 2.3 for the previous class. Of this chapter, for this class you should read the following sections:

- Section 2.1
- Section 2.2.1

- Section 2.3 (you may wish to re-read this)
- Section 2.4 (**without** Section 2.4.3)

Control flow is the sequence in which tasks of a workflow are supposed to be executed. The simplest control flow if of course a sequence (one after another), but of course that also takes the longest time. Some tasks may be done in parallel (at the same time) if enough resources are available to do them at the same time (this is where the resource perspective will come into play later). This is called an AND branch or split, as you may want to do A and B. At other times, you want to do one thing, but not another. This is called an XOR split. XOR stands for "exclusive or", meaning either or. An OR split ("A or B") means either A or B, or both, whereas an XOR split ("A xor B") means either A or B, but not both. The final split is what YAWL calls the "thread-split", in which case the following tasks can be performed multiple times. For example, to process a travel reservations, one might have to get a manager's authorization for each of the multiple flights on the reservation. These are the basic ideas and Section 2.2.1 has a lot more detail, as well as some tricky variations on these. Remember that all of these, even the tricky variations, are based on real workflows that the YAWL people looked at over 10 years.

Once you understand the various control-flow patterns that a workflow diagram should be able to model, you should be able to see how the YAWL diagram elements can be used to do just that. Section 2.4 introduces you to the YAWL notation, which is quite similar to workflow nets you looked at in the previous class. Figure 2.20 on page 58 gives an overview. The OR-split and OR-join are added by YAWL and the cancellation region is a useful feature that you may have found missing in workflow nets. Cancellation regions are based on the formal idea of reset nets in Section 2.3.3, but it is not important that you understand the formalism in that subsection; it is only necessary that you understand the general concept of a cancellation region as that set of tasks which can be cancelled by some other task. Section 2.4 also describes how YAWL workflow nets are executed, based on the underlying Petri nets. This is why you played around with WoPeD and Petri nets in the previous classes, as it is a gentle introduction to these more advanced YAWL nets.

Chapter 3

Chapter 3 is really a (very brief) addendum to chapter 2 in that it defines more precisely the semantics (meaning) of the OR-join. **Do not read** sections 3.2.2 and 3.4.

The problem with the OR-join is that it is not quite clear what it means. It is frequently used and makes intuitive sense ("I want to continue with task X when tasks A or B are finished"), but translating this intuitive meaning into something that computer software can execute is difficult and ambiguous. For example, what happens when task A is done, but B is not done yet? Technically, "A or B" is true, so we could continue. But what do we then do once B also finishes? Waiting for B is also an option but do we even know that B will finish or is it possible that it gets cancelled? You should read Chapter 3 to at least get a sense of the problem (Section 3.3) and the solution that the YAWL software has adopted (Sections 3.2.1). Start by reading 3.1, then 3.3 and then come back to read Section 3.2.1.

Chapter 8

From reading chapter 7 in class 8 you know that YAWL consists of the workflow engine (the software that executes the workflows) and the workflow editor (the software with which you design the workflows) on your computer. This chapter introduces the YAWL Editor version 2 which is the core

modelling component of the YAWL System. The chapter demonstrates how to create and analyze the control-flow of a YAWL process, how to specify the involved data, how to link organizational resources to process tasks. Again, for now we'll just focus on the control-flow and leave the data and resources for later. The editor enables workflow designers to graphically define complex models, and to analyze and export these models to the Engine.

You can think of chapter 8 as a bit of a user's manual for the workflow editor. Of course YAWL also comes with a proper user's manual, so you may wish to have a look at that as well; it is installed on your computer in the YAWL menu under documentation. For this class, read the following sections:

- Section 8.1
- Section 8.2
- Section 8.5

If you wish to read the user's manual (it's installed on your computer with the YAWL software), the relevant sections in there are 3.1, 3.2, 3.3, 4.1, 4.2, 4.3, 4.4.

The YAWL editor is a software tool to draw the YAWL diagrams (and later add the resource and data definitions). The workflow definitions (diagrams) you create in the editor are saved on your computer and will need to be uploaded to the YAWL workflow engine through a web interface. This is what Figure 8.1 tries to explain (in somewhat technical terms). Section 8.2 then provides a brief introduction how to use the editor (look at the YAWL user's manual for more details, but it's a pretty simple diagramming software, so you should also be able to figure this out for yourself with a bit of playing around). This section also describes how timer-outs can be set, i.e. firing a task automatically after a given amount of time is up. Finally, when you have an XOR split (e.g. “either A or B but not both”) in the control flow, you need to specify a condition that is used to decide which of the following tasks (A or B) to perform. Figure 8.5 shows one such condition. Notice that the condition requires some data to work on, so we won't be able to add these conditions until we have covered the data perspective in the next class.

Finally, Section 8.5 shows how the editor provides error messages to the user when the workflow is incomplete, or otherwise doesn't make sense. It's important that you look at the error messages. Some errors appear in the list because you're not done yet with drawing the entire process, and will disappear as you complete modelling the process. For example, when you have an AND-split without a corresponding AND-join, the editor will show an error. As soon as you put in the join, that error should go away. Other errors require your full attention, especially those that are the result of “Specification Analysis”.

After this chapter (perhaps with the help of the user manual) you should be able to produce YAWL diagrams for simple workflows, without data or resource aspects. Try this by working on the following review questions (which are mostly modelling exercises, so you become familiar with the editor and YAWL models).

Review Questions:

- Name and describe the eight groups of control-flow patterns and give examples for each group.
- Explain the idea of cancellation region.
- Chapter 8, Exercises 2, 3, 4, 5, 6, 7

At this point, we haven't covered the data perspective, so please ignore those parts of the exercises and focus on the diagram only. Exercises 2, 3, and 4 will be very simple without the data parts, while exercises 5, 6, and 7 cover somewhat realistic processes.

- Chapter 2, Exercises 1, 2, 3, 4, 5

  Exercise 1 will again test your understanding of the previous class, as it relates YAWL to Petri nets. Exercises 2, 3, and 4 are somewhat realistic processes. Again, model these without worrying about the data or resource aspects for now (and only model them in YAWL, you've done the Petri net models for the previous class' exercises). Exercise 5 asks you to identify the possible execution traces for a number of workflows. An execution is simply the order in which tasks are executed, from workflow start to workflow finish.

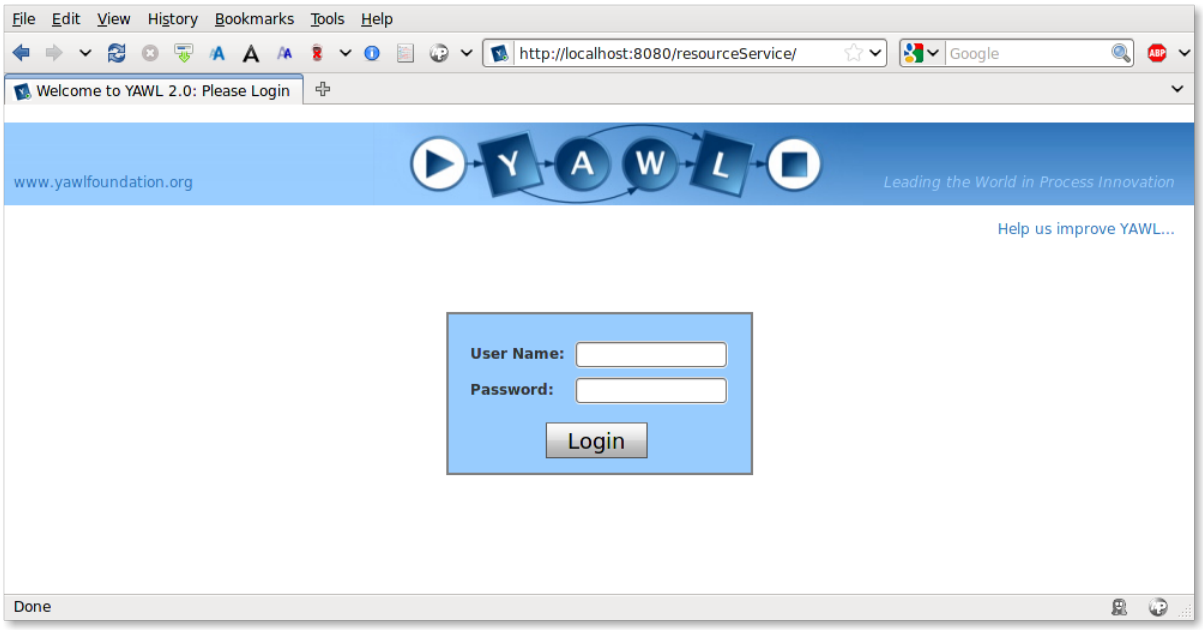
- Chapter 3, Exercise 3

  This exercise tests your understanding of what exactly an OR join means in YAWL and under what conditions a workflow can continue after an OR join.

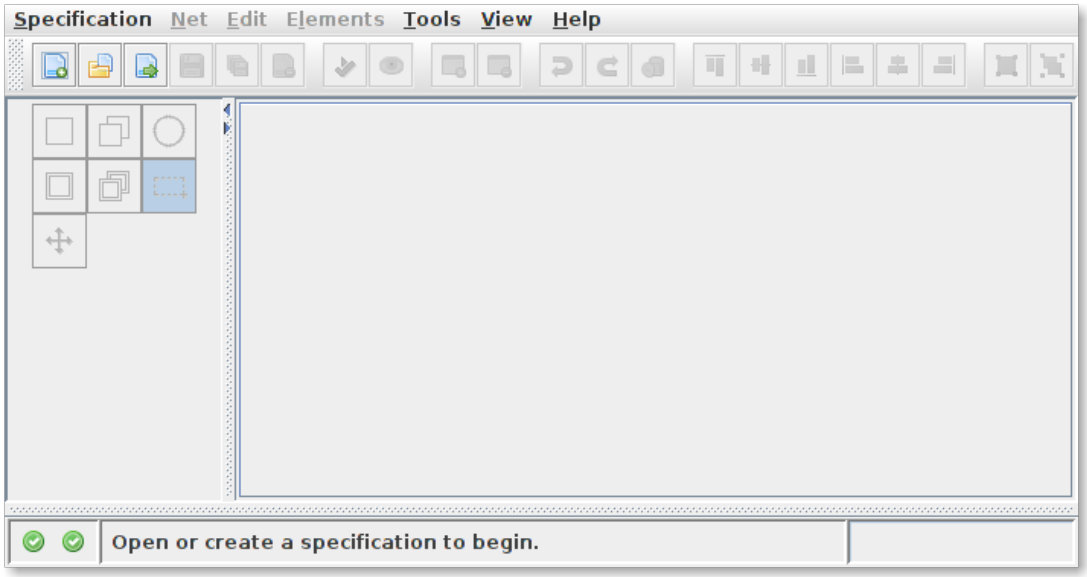
The YAWL Software

The YAWL software and the YAWL user manual are available from the course website. Download the version for your operating system (e.g. Windows, Mac, Linux) and install it. From reading chapter 7 you know that YAWL consists of the workflow engine (the software that executes the workflows) and the workflow editor (the software with which you design the workflows) on your computer. You can accept all the default suggestions for the installation. Refer to Section 2.2. of the user manual.

After installation, you should see (in the Start Menu on Windows, in the YAWL folder in the Applications folder on a Mac) the option to start and stop the workflow engine, access the YAWL control center, and to start the workflow editor. Start the workflow engine. Next, access the YAWL control center. Use the user name “admin” and the password “YAWL” (both are case sensitive) to log in.



Next, start the YAWL editor:


Specification
Net
Edit
Elements
Tools
View
Help
Open or create a specification to begin.