Modeling Workflow with Petri-Nets

Lecture Objectives

In this class, you are going to learn:

- Basic Workflow Concepts
- Mapping Workflow onto Petri-Nets
  - Routing
  - Enactment

Why study workflow modeling?

• The success of a workflow system depends on the quality of the workflows put into it
  – Understand workflow modeling and analysis
• Use Petri nets as modeling tool
  – Can represent a process in a graphical, straightforward and formal manner
  – Allow detailed analysis

Workflow management concepts

A workflow definition is composed out of three parts:

• **Process definition:**
  a description of the process itself

• **Resource classification:**
  a classification of the resources to be used

• **Resource management rules:**
  how to map work onto resources
Process definition (1)

• A process definition specifies
  – *which* steps are required and
  – in what *order* they should be executed.
• Also known as: routing definition, procedure, workflow script
• **Examples**: purchase order, tax declarations, insurance claims process

Process definition (2)

A process definition consists of 3 elements:

• **Tasks** (step, process element)
  – A logical unit of work, e.g., typing a letter, stamping a document, checking personal data
• **Conditions** (state, phase, requirement)
  – A condition is used to determine the enabling of a task.
• **Subprocesses**
  – Use of previously defined processes
  – Allow frequently occurring processes to be *reused*
The Case (1)

- A case is the 'thing' which needs to be processed by following the process definition.
- Also known as: process instance, job, project
- Example: insurance claim, purchase order, tax return, complaint, loan application
- Each case has a unique identity
  – Possible to refer to the case in question

The Case (2)

- A case has a limited lifetime.
- Example: For an insurance claim, a case begins when the claim is submitted and disappears from the workflow system when the claim processing is completed.
- Between the appearance and disappearance of a case, it always has a particular state.
The State of a Case

Determined by 3 elements:

• **Case attributes** (a.k.a. case variables)
  – The logistic attributes of a case used to route the case.
  – Allow a task (under certain conditions) to be omitted from execution
  – Case variable value may change as the case progresses

• **Conditions** (a.k.a. phase, requirement)
  – Denote how far a case has progressed e.g., “order accepted”, “application refused”, “under consideration”
  – Specify requirements satisfied before a particular task is carried out

• **(Contents/application data)**
  – Contained in documents, files, databases
  – Not managed by the WFMS

The Task (1)

• A logical unit of work
• Examples: typing letter, assessing valuation report, filing a complaint...
• Three types of task:
  – **Manual**: performed by $\geq 1$ people without use of an application e.g., making a physical check
  – **Automatic**: performed without any human intervention i.e., a computer application carries out the task entirely based upon recorded data
  – **Semi-automatic**: Person + Application e.g., completion of a valuation report by an insurance assessor supported with a computer program
The Task (2)

- **Indivisible/Atomic**: always carried out in full
- **Rollback**: If anything goes wrong during the performance of a task, must return to the beginning of the entire task
- **Commit**: Once a task is completed, its effect cannot be undone
- Whether a task is atomic depends on the *context* of the task:
  - A task contracted out by a client to a supplier is considered “atomic” by the client
  - A supplier may split the same task set into smaller ones

Task, Work Item, and Activity (1)

- A task is a generic piece of work, NOT the performance of activity for one specific case
- **Work item**: combination of a case and a task which is just about to be carried out
  - Created when the state of a case allows it
  - An actual piece of job which may be carried out
- **Activity**: actual performance of a work item
  - When work begins upon the work item, it becomes an activity
- Both work item and activity are linked with a specific case
The Process

- Describes:
  - How a particular category of cases is executed.
  - Which tasks need to be carried out, and their ordering.
- Hence, a process is a procedure for a case type.
- In general, many different cases are handled using a single process.
  - A task may only be performed on some cases.
  - Conditions decide the ordering.
- Process = tasks + conditions
- Since each case has a finite lifetime (beginning and end), the corresponding process also has a beginning and end
Routing of Cases

- Describes the lifecycle of a case, i.e., which tasks need to be performed and in which order

- 4 basic constructions:
  - Sequential
    "first A then B"
  - Parallel
    "A and B at the same time or in any order"
  - Choice
    "A or B"
  - Iteration
    "multiple A's"

Enactment

- A work item can only be carried out once the state/condition of the cases allow it.
- However, the actual performance may requires more than this.
- **Triggering** of an activity:
  - **Resource initiative**: a work item is executed by a person, he has to take the assignment from his “in-tray”.
  - **External event**: arrival of a request message.
  - **Time signal**: reaching a time limit.

- Now, let us look at how all these concepts can be mapped onto the Petri net!
Lecture Objectives

In this class, you are going to learn:

- Basic Workflow Concepts
- Mapping Workflow onto Petri-Nets
  - Routing
  - Enactment

Mapping a process definition onto Petri nets

```
task

condition

case

case attributes

subprocess
```
Example: Complaint Handling Process

- An incoming complaint is recorded
- The client and the department affected are contacted (can be done in parallel)
- Afterwards, the data are gathered and a decision is taken
- Either (1) a compensation payment is made, or (2) a letter is sent.
- Finally, the complaint is filed.

Example: Modeling Complaint Handling as a Petri net
Example: Handling Multiple Complaints

Question: how to ensure that “collect” does not fire tokens from different cases?

Observations

- The Petri net has a start place (without incoming arcs) and an end place (without outgoing arcs)
- Condition $\rightarrow$ Place; Task $\rightarrow$ Transition
- A case can be represented by one or more tokens
  - The no. of tokens may fluctuate
- Each process should (1) be always possible to reach a state in which there is a token in the end, and (2) when there is a token in end, all others should disappear
  - Every case that begins at start will be completed properly
- Case attribute $\rightarrow$ modeled as a colored token
Lecture Objectives

In this class, you are going to learn:

• Basic Workflow Concepts
• Mapping Workflow onto Petri-Nets
  – Routing
  – Enactment

Routing of cases

• Sequential
  "first A then B"
• Parallel
  "A and B at the same time or in any order"
  – AND-split
  – AND-join
• Selective / Choice
  "A or B"
  – OR-split
  – OR-join
• Iteration
  "multiple A's"
Sequential routing

"First A then B"

Parallel routing

- **AND-split**: allows more than one task to be managed at the same time
- "A and B at the same time or in any order" (i.e., in parallel)
- **AND-join**: synchronizes parallel flows
- t1 and t2 are called *management tasks* which may or may not be "real" tasks
Example: Modeling Complaint Handling as a Petri net

Advantages of Parallel Routing

- Without the aid of a workflow system, a business process which executes cases manually uses **sequential routing**
  - Physical limitations e.g., accompanying document can only be in one place at a time
- Using a workflow system, tasks previously carried out sequentially can now be done in **parallel**
  - Achieve enormous savings
- Allowing parallel routing is one of the big merits of a workflow system!
Selective Routing: **Explicit** Choice of Tasks

- A choice between 2 or more tasks: "A or B"
- 4 management tasks: t1, t2, t3 and t4
- Choice is made immediately by t1 and t2

---

Selective Routing: **Implicit** Choice of Tasks

- Implicit choice: it depends on the "eagerness" of A and B!
- Choice NOT made immediately
Example: Modeling Complaint Handling as a Petri net

Selective Routing based on Case Attributes (1)

We use high-level Petri nets:
- tokens have values: case attribute values
- transitions determine the number of tokens produced: explicit OR-split

• Choice is explicit and may be based on logistic attributes
• No. of tokens produced is between 0 and 1
Selective Routing based on Case Attributes (2)

• Choice is explicit and may be based on logistic attributes
• Does this model act the same as in P.31?

Notations for Common Constructions

AND-split

Explicit OR-split

AND-join

OR-join
Iteration (While…Do…)  

- B may be executed 0 or more times.
- The “While…Do…” construct

Iteration (Repeat…Until)  

- B may be executed 1 or more times.
- The “Repeat…Until…” construct
Complaints handling

• A travel agency processes complaints with the following procedure:
  • An employee first registers every incoming complaint. After registration a form is sent to the customer with questions about the nature of the complaint. This is done by an employee.
  • There are two possibilities: the customer returns the form within two weeks or not. If the form is returned, it is processed automatically resulting in a report which can be used for the actual processing of the complaint.
  • If the form is not returned on time, a time-out occurs resulting in an empty report. Note that this does not necessarily mean that the complaint is discarded.

Complaints handling (2)

• After registration, i.e., in parallel with the form handling, the preparation for the actual processing is started.
  • First, the complaint is evaluated by a complaint manager. Evaluation shows that either further processing is needed or not. Note that this decision does not depend on the form handling. If no further processing is required and the form is handled, the complaint is archived.
  • If further processing is required, an employee executes the task ‘process complaint’ (this is the actual processing where certain actions are proposed if needed).
  • For the actual processing of the complaint, the report resulting from the form handling is used. Note that the report can be empty.
  • The result of task ‘process complaint’ is checked by a complaint manager. If the result is not OK, task ‘process complaint’ is executed again. This is repeated until the result is acceptable.
  • If the result is accepted, an employee executes the proposed actions.
  • After this the processed complaint is archived by an employee.
Lecture Objectives

In this class, you are going to learn:
• Basic Workflow Concepts
• Mapping Workflow onto Petri-Nets
  – Routing
  – Enactment
Triggers

- The workflow system cannot force things to happen in reality:
  - The arrival of an electronic message (EDI) which is needed to execute a task.
  - A resource which starts to work on a case.
  - The arrival of a paper document.
  - A phone call to confirm a purchase order.
- A workflow system is a reactive system, i.e. it is triggered by the environment.
- Some tasks require a trigger.

We identify four kinds of tasks:

- **Automatic**
  No trigger is required.

- **User**
  A resource takes the initiative.

- **External**
  A external event (message, phone call) is required.

- **Time**
  The task requires a time trigger.
The triggering concept can be modeled in terms of PN

\[ \text{trigger token} \]

\[ \text{O} \rightarrow \text{A} \rightarrow \text{O} \]

However, we will omit the extra place.

There is a subtle but important distinction between the following terms:

- **task**
  A logical step which may be executed for many cases.

- **work item** = task + case
  A logical step which may be executed for a specific case.

- **activity** = task + case + (resource) + (trigger)
  The actual execution of a task for a specific case.

Work items and activities are task instances.
## Process and Petri Net

<table>
<thead>
<tr>
<th>Process Definitions</th>
<th>Petri Net Terms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task</td>
<td>1 or more transition</td>
</tr>
<tr>
<td>Work item</td>
<td>Transition being enabled</td>
</tr>
<tr>
<td>Activity</td>
<td>Firing of a transition</td>
</tr>
</tbody>
</table>

### A process definition

A (sub)process has one input and one output place.
Implicit and Explicit OR-Split Revisited

Rollback and Commit

- A task is an atomic unit – always carried out in full
- Faults can happen during the performance of task:
  - the data operator found that certain data for carrying out a task are missing,
  - The application program crashes,
  - System error in the workflow system, etc.
- A Rollback is required: returning the workflow system to its state prior to the heart of the activity
  - Case attributes and conditions are restored to original values
  - Restart the activity
  - Can be more complicated if applications are involved
- After the activity has been successfully completed, a commit occurs and all changes made become definitive.
Example

- Model the workflow about complaint handling (P.36-37) again by adding triggers. Note the highlighted letters in the question.
Solution

Conclusions

• We have presented the precise definitions of workflow processes.
• It is possible to model workflow processes succinctly with high-level Petri nets.
• 5 constructs: AND-join, AND-split, OR-join, OR-split, iteration
• The difference between Explicit and Implicit Or-split
• It is important not just understand the definitions, but also try to apply Petri net to model real problems!
• In the next lecture, we investigate how resources are assigned to tasks.
Exercise: Travel Agency

• A travel agency organizes trips. To organize a trip the following tasks are executed.
• First the customer request is registered, then an employee searches for opportunities which are communicated to the customer. Then the customer is contacted to find out whether (s)he is still interested and whether more alternatives are desired.
• 3 possibilities: (1) the customer is not interested; (2) he wants more alternatives; (3) he selects a trip.
• If the customer selects a trip, then the trip is booked. In parallel (if desired) one or two types of insurance are prepared. A customer can take insurance for trip cancellation or/and for baggage loss.
• A customer can decide not to take any insurance, just one type of insurance, or both types of insurances.
• Two weeks before the start date documents are sent to the customer.
• The customer can cancel the trip at any time after completing the booking process (including insurance) and before the start date.
• Customers who are not insured for trip cancellation can cancel the trip.