Flexible Business Process Modelling via Dynamic Condition Response Graphs

Morten Marquard and Tijs Slaats

Joint work with
Thomas Hildebrandt, Soren Debois and Raghava Rao Mukkamala
Overview of Tutorial

• Background
• Introduction to DCR Graphs
• Hands-on Modelling Exercise
• Questionnaire
• Discussion
Exformatics

• Small Danish company (15 employees)
• Founded in 2003
• Provides software solutions for a large (40+) customer base, including:
  – Intellectual property, legal
  – Sales and delivery processes
• Develops IT systems for knowledge workers
Knowledge Workers need Flexible Workflow Systems

• Knowledge Workers:
  – Solve diverse problems
  – Are experts at what they do
  – Require freedom to make their own decisions

• However, rules do exist:
  – Laws
  – Business practices
State-of-the-Art

- Current Workflow systems:
  - Solve tasks in given order
  - The system is in control, not the user
Knowledge Workers need Flexible Workflow Systems

• Flexible Workflow Systems:
  – Based on describing rules directly instead of describing the flow of work
  – Offers users all possible choices that follow the rules, while still advising on best-practice
  – Are more easily adapted to change (new laws, changing business practices)
  – Require flexible workflow notations
DCR Graphs

• Such a declarative workflow notation
• Consists of events (tasks) and constraints (rules) between events
• Unconstrained events can happen at any time
• State represented as a marking consisting of executed, pending and included events
DCR Graphs by Example

We consider a basic expense claim example, with:

• 4 main activities:
  – Create Expense Claim
  – Approve Expense Claim
  – Reject Expense Claim
  – Payout Expense Claim

• And three roles:
  – Employee, can create an expense claim.
  – Manager, can approve or reject an expense claim.
  – Finance Department, can reject and payout an expense claim.
DCR Graphs by Example

- A claim should be created before it can be approved or rejected.
- A claim should be approved before it can be paid out.
- A claim should only be created once. (Every run of the workflow handles a single claim.)
- Once a claim has been rejected, it should not be paid out, unless it is approved again at some later point in time.
- If a claim is created, it should eventually be paid out, unless it is rejected.
- Payout should end the process.
Questions?
Hands-on Assignments
Discussion