Process Modelling

Notations, Methods and Enaction

Keith Phalp
Department of Electronics and Computer Science
University of Southampton
kp@ecs.soton.ac.uk

Invited talk at Bournemouth University, January 1997
Process Modelling

Capturing and describing a process for some purpose

Key feature of process modelling:
“many of the phenomena encountered must be enacted by a human rather than a machine”.
Overview

• Context. Importance of notation?
  – Many different approaches. (Choice).
  – Knowing the model is right.

• Modelling experience, and impact on methods.

• Using enactable models. An example.
Faith, Maths, Validation and Debugging

• Formal: Rigour: Need checking by experts
• Pragmatic: Typically diagrams:
  – Sacrifice rigour for understandability.
  – Users validate. Suffer from multiple interpretation
• Enactable. Visual. Try it out.
• Combinations.
  – Understandability and rigour.
  – Flexibility and familiarity.
  – Separation of concerns.
Business Processes: Experience

• Initial modelling must start with an easy to understand approach (diagrams).
• Even simple diagrammatic notations lead to complex models
  – which users find difficult to comprehend.
• Mechanisms to add structure to detailed models can help.
• Enactable notations do help
  – but users need to be shielded from them.
Putting it together

• Elicitation: Capture & describe processes using notations which users will understand (validate and feedback).

• Experimentation: Further understand process: Experiment with executable models / scenarios.

• Presentation / Discussion / Education - with understandable static and executable models.
A Spiral Method

- Present
- Present
- Present
- Present

Structure Structure Structure Structure Structure

Detail Detail Detail Detail

Current Process Process Scenarios Proposed Process

Time / Model Maturity
Detail, Structure, Present

- Establish detailed notation(s)
- Gather evidence (multiple sources).
- Produce model based on documentary understandable evidence.
- Analysis of detailed Modelling:
  - Run process scenarios (analyse)
  - include process data
  - Augment model with findings
  - Produce alternative process scenarios.
  - Validate process understanding

- Establish structured notation
- Impose structure
- Produce structured (understandable) models
- Highlight findings.
- Produce multiple viewpoints

- Choose presentation mechanism
- Present inconsistencies, findings, process scenarios.
- Use models to facilitate discussion.
- Use structured models, e.g., POSD models, models as guideline for reports.
A Route Through the Notations

User Facing
- DFDs
- RADs
- IDEF
- PWBS

Enactable Models
- CSP
- RolEnact
- Enact

Simulation and Support
- RolEnact Windows
- Model Based Slide Shows
- Other Support e.g. PWI

Paradigms (e.g. RolEnact)
- Interfaces

An illustrative route
Action

Action Role.Action
   Me(before $\rightarrow$ after)
End

Action Project_Manager.prepare_a_plan
   Me(estimate_received $\rightarrow$ plan_prepared)
End
Interaction

Interaction Role1.Interaction
   Me(before1 → after1)
   Role2(before2 → after2)
End
Selection

Selection Role1.Selection
  Me(before1 → after1)
  Role2(before2 → after2)
End

Automatically creates:
Me.Role2:=r,
r.Role1:=Me
Creation

Create Role1.Create
    Me(before1 → after1)
    new Role2
End

![Diagram showing the process of creation]
A Role: Director

Create Divisional_Director.newProject_Manager me(initial → manager_started)
    new Project_Manager
End

Interaction Divisional_Director.agree_TOR me(manager_started → initial)
    Project_Manager(initial → agreed_TOR)
End
The ‘state’ bar - indicating the current state of the Role
The ‘do’ Button
The list of actions the Role can enact

The abled action which the Role can enact in its current state
RolEnact for Designer

[Diagram of roles and tasks, showing interactions between Divisional Director, Designer, Project Manager, and Designer_Estimator]
Other Interfaces for RolEnact
Recap

• Modelling Story: Commonly used notations for elicitation and validation.
• Coherent route from user facing models to experimentation and simulation.
Extra Slides

RolEnact detail
Project Manager

Create  Project_Manager.newDesigners
  me(agreed_TOR → designers_started)
  new Designer
  new Designer_Estimator
End

Action  Project_Manager.write_TOR
  me( designers_started → TOR_written)
End

Action  Project_Manager.Prepare_plan
  me(estimate_received → plan_prepared)
End

Interaction  Project_Manager.send_plan
  me( plan_prepared → plan_sent)
  Designer_Estimator(sent_estimate → received_plan)
End

Interaction  Project_Manager.agree_delegate
  me( TOR_written → delegated)
  Designer(initial → delegated)
  Designer_Estimator(initial → delegated)
End

Action  Project_Manager.debrief
  me( design_received → project_completed)
End
Designer

Action Designer.choose_method
    me(delegated \rightarrow method_chosen)
End

Interaction Designer.ready_for_design
    me(method_chosen \rightarrow able_to_design)
    Project_Manager.Designer_Estimator(received_plan \rightarrow ended)
End

Action Designer.design
    me(able_to_design \rightarrow design_produced)
End

Action Designer.check_design
    me(design_produced \rightarrow assessing_design)
End
Designer

Action Designer.accept_design
    me.(assessing_design → accepted_design)
End

Action Designer.reject_design
    me(assessing_design → able_to_design)
End

Interaction    Designer.deliver_design
    me(accepted_design → design_sent)
    Project_Manager(plan_sent → design_received)
Designer_Estimator

Action Designer_Estimator.prepare_estimate
   me(delegated → estimated)
End

Interaction   Designer_Estimator.obtain_estimate
   me(estimated → sent_estimate)
   Project_Manager(delegated → estimate_received)
End