Supporting a Process-Oriented Requirements Method

Keith Phalp, Sherry Jeary, Lai Xu
{KPhalp, SJeary, lxu}@bournemouth.ac.uk
Bournemouth University
Outline for Today

• Process Models
• Moving from Process Model to Specification
• Some Simple Approaches to Help (in this case)
• A Model Driven Perspective
• Situational Enterprise Applications
• Mashups
• Existing Tools for Situational Mashups
• Round-up
• EU FP7 SOA4All
• Resources for Building Mashup Applications
The problem in a nutshell

• Customers want systems to support their business processes. (We can argue about the b word).
• Developers build systems for clients.
• Often the following happens:
  • “Oh dear. The system doesn’t meet the client’s needs”.
• It must be someone’s fault
  • “The customer didn’t tell us what they wanted”
  • “The developers didn’t understand the problem”
• This is a requirements problem, and very common.
• Often a lack of alignment (business and IT)
Terminology: Descriptions and phases

Environmental phenomena

Machine phenomena

The environment

The machine

Specification (interface)

Terminology: Descriptions and phases

analysis  specification  design

problem domain  interface  solution system
Role Activity Diagrams

- Initially, promoted by Praxis & Coordination Systems (Roberts), and the DTI sponsored IOPTClub.
- Variants and extensions, e.g., PROCESS project (Southampton Uni, 94-97) produced families of models (mapping to CSP) and enactable models (RolEnact 98).
- Recent resurgence of interest, with popular books by Keith Harrison Broninski and Martin Ould (both BCS).
- Still supported by many, e.g., see Venice Consulting (Martin Ould’s site), for much of interest.
  - www.veniceconsulting.co.uk
‘For an individual (or group) in the organisation to carry out their activities, they need to know what activities they must take part in, in what order those activities must take place, what other individuals or groups they must interact with, and which actions are dependent upon those interactions’.

Role Based Models

• ‘Role based models satisfy these requirements by grouping activities into ‘roles’, which describe the desired behaviour of individual groups, or systems’.

• ‘A role involves a set of activities which, taken together, carry out a particular responsibility or set of responsibilities’.
For now, for brevity, we have omitted states from the diagram.

No choice constructs shown here.

Have included a System Role.
A ‘Purist’ RAD approach
The importance of Interaction

- Activity (or activities) carried out at the same point as another activity in another role (or roles).
- A shared event.
- The consequence of an interaction is that all of the roles involved move from their current state to their next state.
- Act as points of synchronisation (or control) of the process.
- Interactions are synchronous.
ISSUES

- Process models and software models have different perspectives and languages.
- For say RAD to use case, they can be considered almost orthogonal.
  - Activities assigned to roles versus actors (roles) assigned to processes.
- Hence difficult to preserve mapping in moving from process model to software.
  - In addition, there may be information loss if our software constructs are not sufficiently powerful (rich enough).

SOLUTIONS (lots of different ones of course)

- A very thorough approach, which can be found in BSCP (Business Strategy Context Process) paper, moves from strategic view, using goal models, problem frames, and process models (Role Activity Diagrams).
- Present here aspects of a lightweight view (often used in teaching), more pragmatic and accessible (simple notational devices).
• This is against the purist approach, and a rather simplified (teaching) example.
• We (as software engineers) move towards specification.
• Need to ensure that we capture the system boundary (as with say a Yourdon Context Diagram).
• Need to ensure that, in moving to spec, we show cross boundary (problem to machine interactions).
• With a system RAD (usually will have different sub-system names) the interaction is between the roles (which will be actors) and the system role.
• This will correspond to use case associations.
• Hence, RAD acts as a way to consider the problem domain (inform requirements).
• RAD (with system roles) allows one to ‘discover’ or discuss the system boundary.
• Acts as a link between business view (intentions for system) and IT.

Practical
• Acts as a checklist for the specification.
• Gives a first cut list for the use case diagram communications.
• Of course the meat (of the use case) is in the descriptions.
Two sporting use cases

1. The match reached full-time
2. The referee blew his/her whistle
3. The ball crossed the goal-line
4. The goal was not given

Alternatives
4. The goal was given

Validation & Context. Someone who ‘knows the the game’. 
Suppose our event is now *Make smoothie*, which requires that when we have fruit. We actually have both apples and oranges.

For a use case we would be required to choose that the gaining of apples and oranges occurs in some arbitrary sequence. That is:

1. Fruit Finder get apples
2. Fruit Finder get oranges

However, in reality one might gather these fruits independently and in any, often unknown order.

**Also within the Use Case description the dependencies are unclear**

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

Interaction Keith.gives_pen
Me (has_pen → no_pen)
Sherry (no_pen → has_pen)
End

<table>
<thead>
<tr>
<th>Actor</th>
<th>Event</th>
<th>pre</th>
<th>post</th>
<th>Actor 2</th>
<th>pre</th>
<th>post</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keith</td>
<td>gives pen</td>
<td>has pen</td>
<td>no pen</td>
<td>Sherry</td>
<td>no pen</td>
<td>has pen</td>
</tr>
</tbody>
</table>
Considering dependencies

1. Client requests connection via Schedule
2. Scheduler acknowledges connection
3. Client sends network layout
4. Scheduler creates network handler
5. Scheduler registers network handler
6. Client starts executing its tasks

- Produces an enactment.
- States controlling which events can be invoked.
- Allows stakeholders to experiment with behaviour.
• **Style 3 (contd.)**
  
  • *The patient stands next to the doctor.*
  • *He puts the prescription in his pocket.*
  
  • Who is “he”? Whose pocket is “his”? Write proper nouns / names instead:
    
    • *The doctor puts the prescription in the patient’s pocket.*
    • *The GP puts the prescription in the customer’s pocket.*
  
  • This sentence is at fault because it uses synonyms (GP for doctor and customer for patient). Only use the agreed language of the domain since a synonym does not convey the same meaning.
One in-house approach has versions of RADs (environment, shared and machine RAD), which are then transformed (via rules) to UML models (class diagram), activity diagrams and use cases.

Similar idea to the system roles.

More rigorous (but requires sets of models)

Allows for direct transformation (stronger alignment).

Well received by our final years (but then they are Computing and BIT students – so amenable to software models and MDA).

Rules can be applied as guidelines too (written), or codified.

However, not clear that this is sufficiently accessible, which leads to our work on accessible models (part of VIDE).
VIDE Project

- **VIDE – Visual Model Driven Development**
- European Commission funded €2,298,436 ($2.95 million).
- To make model driven development fulfil business needs.
- To create novel tools to support model driven development, and to provide an end-to-end development process supported by visual notations.
- 10 partners including: SAP, 2 divisions of Fraunhofer, Softeam and Rodan (tool developers).
- BU focus specifically on accessibility of notation, particularly at the CIM level.
- Simple mechanisms (scrapbook, bloops) and wizards to support transformations (e.g., BPMN to Class diagram).
Design stage - initial opportunity scenario
Prototype implementation using Eclipse
Recap: The Story so far

• Role models (RADs) help to inform requirements.
• Issues in moving to specification (from RADs to use cases) can be mitigated by simple notational additions.
• For model driven approaches guidance becomes transformation rules (from RADs to UML models).
• Require an accessible (alignment focussed) MDA approach.
• The VIDE (MDA tool) is helpful and accessible, but large investment is required in tooling etc, and very much an application based approach (even though framework based).
• Need to consider how other technologies and approaches can help in this space, particularly for the SME.
• *Process oriented mashups may provide the equivalent to VIDE but from a different, web services, perspective.*
Outline for Today

- Process Models
- Moving from Process Model to Specification
- Some Simple Approaches to Help (in this case)
- A Model Driven Perspective
- Situational Enterprise Applications
- Mashups
- Existing Tools for Situational Mashups
- Round-up
- EU FP7 SOA4All
- Resources for Building Mashup Applications
Characteristics of traditional enterprise systems

- Closed
- Centralized control
- Mission critical
Enterprise information system landscape
An enterprise needs to be able to *quickly and cost-effectively change* how it does business and who it does business with (suppliers, business partners, customers), etc.
“Yes, we can!”
Empowering the Non-Technical Business Users

Situational Enterprise Applications in the Application Universe

- Air traffic control, NASA
- Enterprise Applications: ERP, CRM, SCM
- Situational Enterprise Applications: perimeter ERP, perimeter CRM, perimeter SCM
- Email, Calendar
Examples of Situational Applications

• Excel → spreadsheets
• Access → simple databases
• Visual basic for applications → small office-based applications
• ......
Key Spreadsheet Properties

- End users control programs
- Accessible “programming”
- Easy to modify
Spreadsheet on the Web

Software Systems Research Centre
Supporting Process Oriented Requirements
Situational Web Application Characteristics

- Easy to create
- Solves local problem for creator
- Focus on fast deployment
- Power to the local people
- Feeds
- Web services
- Composition environment
- Execution environment
- Social software aspects
Examples of Situational Applications

- Excel → spreadsheets
- Access → simple databases
- Visual basic for applications → small office-based applications
- ......
- Mashups → data aggregation applications
Outline for Today

- Process Models
- Moving from Process Model to Specification
- Some Simple Approaches to Help (in this case)
- A Model Driven Perspective
- Situational Enterprise Applications
- Mashups
  - Existing Tools for Situational Mashups
  - EU FP7 SOA4All
  - Resources for Building Mashup Applications
Example: Google maps
A “mashup” is a lightweight tactical integration of multi-sourced applications or content into a single offering (Gartner).

Web application hybrid that combines data from more than one source into a single integrated tool (Wikipedia).

Mashups are loosely coupled distributed systems... to the extreme. The developers of the individual components do not know each other and possibly do not even know that their application is being used as a component by another application (Wilde).
Minimal Mashup Architecture

Presentation/Interactivity
- HTML/XHTML
- CSS
- JavaScript
- Ajax

Web Service (API)
- XMLHttpRequest
- XML-RPC
- SOAP
- REST

Data
- XML
- RSS/Atom
- JSON
- KML
<table>
<thead>
<tr>
<th>Category</th>
<th>Example APIs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mapping</td>
<td>Google Maps, Yahoo Maps</td>
</tr>
<tr>
<td>Photo</td>
<td>Flickr, Picasa Web Albums</td>
</tr>
<tr>
<td>Search</td>
<td>Google Search, Yahoo Search</td>
</tr>
<tr>
<td>Video</td>
<td>Youtube, Yahoo video search</td>
</tr>
<tr>
<td>e-Commerce</td>
<td>Ebay API, Yahoo Shopping</td>
</tr>
</tbody>
</table>

Additional APIs:

- Google: Google Maps, Google Search, Youtube, Google Video, Google Shopping
- Yahoo: Yahoo Maps, Yahoo Search, Yahoo Video, Yahoo Shopping
- Microsoft: Virtual Earth, Windows Live Spaces Photos, Bing, Windows Live Expo
- Amazon: Amazon e-Commerce Service
- Ebay: Ebay API
Mashup Categories (2009)

Source: programmable web http://www.programmableweb.com/

Software Systems Research Centre
Supporting Process Oriented Requirements
And Then

Positives
- Lots of API's
- Lots of data sources
- Many easily accessible compositions

Negatives
- Data centered
- More advanced compositions need manual coding
- Alphabet soup of standards
Situational Applications Versus Mashups

- Mashups can be situational applications
  - Main functionality is external
  - Advanced mashups require detailed and extensive programming
- Situational applications are broader
- Much business functionality has a dynamic component
Outline for Today

- Process Models
- Moving from Process Model to Specification
- Some Simple Approaches to Help (in this case)
- A Model Driven Perspective
- Situational Enterprise Applications
- Mashups
- Existing Tools for Situational Mashups
- Round-up
- EU FP7 SOA4All
- Resources for Building Mashup Applications
Google docs: Create a Form

Software Systems Research Centre
Supporting Process Oriented Requirements
Google docs: The Resulting Form

Favourite Shops

Fill out your favourite shops

*Required

Name of the Shop *

Location of the shop *

Submit

Powered by Google Docs

Report Abuse - Terms of Service - Additional Terms
### Favourite Shops

<table>
<thead>
<tr>
<th>Timestamp</th>
<th>Name of the Shop</th>
<th>Location of the shop</th>
</tr>
</thead>
<tbody>
<tr>
<td>17/08/2009 10:49:54</td>
<td>Lidl</td>
<td>Licherfeldstrasse 11, 9001 St. Gallen, Switzerland</td>
</tr>
<tr>
<td>17/08/2009 10:40:13</td>
<td>Coop Galusmarkt</td>
<td>Festwiesenstrasse 6, 9001 St. Gallen, Switzerland</td>
</tr>
<tr>
<td>17/08/2009 10:33:42</td>
<td>Zottibell</td>
<td>Madirasse 19, 9003 St. Gallen, Switzerland</td>
</tr>
<tr>
<td>17/08/2009 09:32:51</td>
<td>Ilona</td>
<td>Zürcherstrasse 460, 9015 St. Gallen, Switzerland</td>
</tr>
<tr>
<td>17/08/2009 09:02:12</td>
<td>Migros St. Fiden</td>
<td>Buchstrasse 25, 9000 St. Gallen, Switzerland</td>
</tr>
<tr>
<td>17/08/2009 08:21:29</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Google docs: Let's Add A Gadget

Add a Gadget

Maps

Have a better idea?

Write your own gadget to display data in cool new ways. Want to see your gadget on this list? Submit it to us using the submission form.

More of the gadgets in this directory were developed by other companies in the Google Apps for Work directory, or in the Terms of Service and Privacy FAQs, where using these gadgets is permitted.

Spreadsheet Mapper Gadget
By After Go

This gadget provides an easy way for a company to create a store locator. From a list of addresses, this gadget will generate a sample store locator page, which a marketer could copy and customize, and a downloadable XML file.
Google docs: Configure the Gadget
Google docs: Our Gadget Ready to Export
Google docs: Add the result to iGoogle
Outline for Today

- Process Models
- Moving from Process Model to Specification
- Some Simple Approaches to Help (in this case)
- A Model Driven Perspective
- Situational Enterprise Applications
- Mashups
- Existing Tools for Situational Mashups
- Round-up
- EU FP7 SOA4All
- Resources for Building Mashup Applications
Situational Application New Relevance

- More XML content
- More services
- More feeds
- Cloud provides computing in the network (The Network is the computer - SUN)
- Cheap, network accessible computing gives ability to users:
  - salesforce.com
Types of Situational Enterprise Applications

- Data-oriented applications
  widgets, gadgets, pipes and mashups

- Process-oriented applications
  SOA4All
Extending the Range of Mashups

- Support processes, not only data
- Allow non-programmer composition of advanced apps with interesting components
- Perform technical compatibility adjustments automatically
Motivated Examples
Positioned next to enterprise workflow engine

Allow for support of “unsupportable processes” that are:

- Dynamic in nature
- Not common enough
- Hard to explain to non-domain experts
- Too small for centralised implementation, benefits too small
BPM Mashup Requirements

- Central platform responsible for execution
- Easy composition language
- Easy access to services
  - Semantic matching / retrieval
  - Automatic wrappers
Minimal Business Process Mashup Architecture

Presentation/Interactivity

Control flow

Web Service (API)

Data

XML

RSS/Atom

JSON

KML

...
Outline for Today

- Process Models
- Moving from Process Model to Specification
- Some Simple Approaches to Help (in this case)
- A Model Driven Perspective
- Situational Enterprise Applications
- Mashups
- Existing Tools for Situational Mashups
- Round-up
- EU FP7 SOA4All
- Resources for Building Mashup Applications
**Goal**

- building a distributed infrastructure that brings WS and SOA to a Web scale
- providing a platform where everyone can participate.

**EU Funded Project (FP7)**

- March 2008 – February 2011
- Project Budget: 13,7 m €
- 16 Partners from academia and industry
Main Features of SOA4All From the User’s Perspective

- **empowerment** of business users
  - web-based tools
  - guidance by wizards
- **wisdom of the** business **crowds**
  - share processes and knowledge (comments, tags, ratings) through a community
- **lightweight** business processes
  - SAP ES, public web services, and human tasks
  - modeling and execution
- **semantics** as base technology
  - smart discovery and composition of services
SOA4All Overall Architecture

Software Systems Research Centre Supporting Process Oriented Requirements
Outline for Today

• Process Models
• Moving from Process Model to Specification
• Some Simple Approaches to Help (in this case)
• A Model Driven Perspective
• Situational Enterprise Applications
• Mashups
• Existing Tools for Situational Mashups
• Round-up
• EU FP7 SOA4All
• Resources for Building Mashup Applications
Mashups

Google

IBM Mashup Center

SERENA

Netvibes

NextWeb

Yahoo!

Pipes

Kapow Technologies

dapper
Resources for Creating Mashups

- Seekda (webservices.seekda.com) Web service search engine
- Programmable Web (www.programmableweb.com) mashup directories
- Social bookmarking web site for searching feeds (www.syndic8.com)
Outlined a variety of process oriented approaches to development, which all aim (by different means) to improve stakeholder involvement and increase alignment of IT with business needs.

- Simple role based process models to inform requirements.
- Methods for moving from role models to use cases, to identify system boundary, maintain mapping and enhance alignment.
- Improvement in specification to utilise richness of process and enhance use case comprehension and power.
- Model driven approaches (transform process model to software models)
- Application toolset (VIDE) to provide accessible CIM models (process models) as a first step in model driven development.
- Introduced mashups for situational enterprises.
- Process oriented mashups to provide accessible and efficient development and resources for mashup development.