Process modeling perspectives

Separation of concerns: engineer models separately
Data-centric processes

Data-centric process model combines data and behavior perspective

- Artifact-driven BPM (IBM)
- Case handling/management
- Document-driven workflows (PSU)
- Data-driven process structures (Ulm Univ)
- Process models with data flows
Why data-centric processes?

- Gaining flexibility

**classic**

**data-centric**
Why data-centric processes?

• Gaining flexibility
• Gaining precision
• Gaining understandability
  − for business users, thinking in terms of entities (artifacts) is more easy then thinking in terms of processes [IBM]
Data-centric processes

Data-centric process model combines data and behavior perspective

- Artifact-driven BPM (IBM)
- Case handling/management
- Document-driven workflows (PSU)
- Data-driven process structures (Ulm Univ)
- Process models with data flows
Outline

• Framework
• Analysis of workflow data models
• Synthesis of object-life cycles
• Transactional process views
• Splitting GSM schemas
Control flow and data flow

message passing: loose integration

shared database: tight integration
Data granularity

- ~ how many states can a data item have?
- Variable: each value is a state
- Object: discrete, explicitly specified states
Business process scope

- Intra-organizational
- Inter-organizational
Data flow dimensions

- **Data integration**
  - **Tight**: Shared database
  - **Loose**: Message passing

- **Data granularity**
  - **High**: variables
  - **Low**: object states

- **Scope**
  - **Intra-organizational**
  - **Inter-organizational**
Research topics

- Analysis of workflow data models
- Synthesis of object-life cycles
- Transactional process views
- Splitting GSM schemas

<table>
<thead>
<tr>
<th></th>
<th>Shared database</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Object</strong></td>
<td></td>
<td><strong>Object</strong></td>
</tr>
<tr>
<td>Intra-organizational</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Analysis of workflow data models</td>
<td>Synthesis of object-life cycles</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Intra-organizational</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Intra-organizational</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Transactional process views</td>
<td>Splitting GSM schemas</td>
</tr>
</tbody>
</table>

Table: Intra-organizational vs. Inter-organizational

- Shared database: Analysis of workflow data models, Transactional process views
- Message: Synthesis of object-life cycles, Splitting GSM schemas
### Research topics

- **Analysis of workflow data models**
- Synthesis of object-life cycles
- Transactional process views
- Splitting GSM schemas

<table>
<thead>
<tr>
<th>Shared database</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Object</strong></td>
<td><strong>Object</strong></td>
</tr>
<tr>
<td><strong>Intra-organizational</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Inter-organizational</strong></td>
<td></td>
</tr>
</tbody>
</table>
Analysis of workflow data models
(joint work with Diana Borrego of Un. of Seville)

• Workflow models with variables and pre/postconditions for activities
• Workflow gets stuck at activity if precondition fails
• How can we diagnose correctness of workflow models with pre/postconditions?
### Example

#### Precondition violation

<table>
<thead>
<tr>
<th>Activity</th>
<th>Precondition and Postcondition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECR</td>
<td>pre: true</td>
</tr>
<tr>
<td></td>
<td>post: true</td>
</tr>
<tr>
<td>SAP</td>
<td>pre: true</td>
</tr>
<tr>
<td></td>
<td>post: true</td>
</tr>
</tbody>
</table>

- **Precondition**

<table>
<thead>
<tr>
<th>Event</th>
<th>Precondition</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>sponsorship &gt; 0 (\lor) numPapers &gt; 60</td>
</tr>
<tr>
<td></td>
<td>post: regFee (<em>) 0.1 (\leq) dinner (\land) dinner (\leq) regFee (</em>) 0.35</td>
</tr>
<tr>
<td>L</td>
<td>sponsorship &gt; 0 (\lor) numPapers &gt; 60</td>
</tr>
<tr>
<td></td>
<td>post: regFee (<em>) 0.1 (\leq) 3 (</em>) lunch (\land) 3 (<em>) lunch (\leq) regFee (</em>) 0.35</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Event</th>
<th>Precondition and Postcondition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECR</td>
<td>pre: true</td>
</tr>
<tr>
<td></td>
<td>post: true</td>
</tr>
<tr>
<td>SAP</td>
<td>pre: true</td>
</tr>
<tr>
<td></td>
<td>post: true</td>
</tr>
</tbody>
</table>

- **Postcondition**

### Activity

- **Precondition**

  - D: sponsorship > 0 \(\lor\) numPapers > 60
  - L: sponsorship > 0 \(\lor\) numPapers > 60
  - OS: sponsorship > 0 \(\lor\) numPapers > 60
  - O: sponsorship > 0 \(\lor\) numPapers > 60
  - R: 3 \(*\) lunch + dinner + others < regFee
  - NGS: confAtt \(*\) (3 \(*\) lunch + dinner + others) <
Diagnosis

O: $\text{pre: } \text{sponsorship} \geq 0 \lor \text{numPapers} > 60$
$\text{post: } \text{others} \leq 0.25 \times \text{regFee} \land \text{others} \geq 0.05 \times \text{regFee}$

R: $\text{pre: } 3 \times \text{lunch} + \text{dinner} + \text{others} < \text{regFee}$
$\text{post: } \text{numPapers} \times 1.8 \geq \text{confAtt} \land \text{numPapers} \times 0.5 \leq \text{confAtt}$

NGS: $\text{pre: } \text{confAtt} \times (3 \times \text{lunch} + \text{dinner} + \text{others}) <$
$\text{confAtt} \times \text{regFee} + \text{sponsorship}$
$\text{post: } \text{guestSpeaker} \geq 0.2 \times \text{sponsorship} \land$
$\text{guestSpeaker} \leq \text{sponsorship} + 0.1 \times \text{regFee} \times \text{confAtt}$

IGS: $\text{pre: } \text{confAtt} \times (3 \times \text{lunch} + \text{dinner} + \text{others}) <$
$\text{confAtt} \times \text{regFee} + \text{sponsorship}$
$\text{post: } \text{guestSpeaker} \geq 0.4 \times \text{sponsorship} \land$
$\text{guestSpeaker} \leq \text{sponsorship}$

regfee=200
lunch=30
others=30
3*30+100+30<200
Diagnosis approach

- Workflow graphs + pre/postconditions
- Encode correct control flow execution (subgraph) as Integer Programming model
- Encode pre/postconditions as Constraint Satisfaction Problem
- Combined IP+CSP model processed by CSP solver
Algorithm

- For each activity a with precondition p
  - Find subgraph that leads to a but does not include a
  - Encode subgraph as CSP+IP model
  - Add not(p) to CSP+IP model
  - If solution exists, then error
- If no error found, then the workflow graph is correct
Validation

- Java-based implementation for verifying annotated XPDL models
- Using COMET CSP checker
- The execution time scales linearly in the size of the XPDL models
Results

Contributions
• Seamless integration of process and data verification
• Extensive feedback to fix data flow errors
• Fully implemented

Future work
• Extend set of workflow operators

Research topics

- Analysis of workflow data models
- *Synthesis of object-life cycles*
- Transactional process views
- Splitting GSM schemas

<table>
<thead>
<tr>
<th>Shared database</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Object</strong></td>
<td><strong>Variable</strong></td>
</tr>
<tr>
<td>Intra-</td>
<td>Analysis of workflow data models</td>
</tr>
<tr>
<td>organizational</td>
<td><em>Synthesis of object-life cycles</em></td>
</tr>
<tr>
<td>Inter-</td>
<td>Transactional process views</td>
</tr>
<tr>
<td>organizational</td>
<td>Splitting GSM schemas</td>
</tr>
</tbody>
</table>
Synthesis of object life cycles
(joint work with Pieter Van Gorp)

• UML activity diagrams implicitly specify object life cycles
• Object life cycles are specified in UML statecharts
• *How can we discover implicit object life cycles from UML activity diagrams?*
• Why?
  − Check consistency with pre-existing object life cycles
  − Generate software code

Withdraw cash → Check balance → Dispense cash → Print receipt

: Account [uncredited] → Check balance

: Account [checked] → Dispense cash

: Account [credited] → Print receipt

TU/e
Approach

Two sequential steps
• Filtering (to remove irrelevant parts)
• Synthesis (to construct statechart hierarchy)

Fully automated!
Filtering

- Reduction rules

\[ \text{Diagram showing reduction rules R1, R2, R3, R4a, R4b, R5 with symbols representing actions, nodes, bars, and diamonds.} \]
Filtering effect
Synthesis

- Transformation rules to construct state hierarchy
  - rule for creating AND state
  - rule for creating OR state
Example construction AND state
Final statechart

• Add transitions to statechart hierarchy
Implementation

GrGen
Graph Rewriting Tool
with optional Visualization

Filtering rules
Synthesis rules

UML Activity Diagram
XMI

UML Statechart
XMI

Eclipse based editors
(Eugenia/Kieler and Official UML 2 Tools)
Final remarks

Contributions

- Synthesis of expressive hierarchical state machines
- Structure preserving
- Filtering out of irrelevant parts
- Fully implemented prototype using GrGen
- Extended to deal with multiple objects

Future work

- Enhance set of translations
- Support roundtrip engineering


## Research topics

- Analysis of workflow data models
- Synthesis of object-life cycles
- *Transactional process views*
- Splitting GSM schemas

<table>
<thead>
<tr>
<th>Shared database</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Object</strong></td>
<td><strong>Variable</strong></td>
</tr>
<tr>
<td><strong>Intra-organizational</strong></td>
<td>Analysis of workflow data models</td>
</tr>
<tr>
<td><strong>Inter-organizational</strong></td>
<td><em>Transactional process views</em></td>
</tr>
</tbody>
</table>
Transactional Process Views
(Joint work with Jochem Vonk & Paul Grefen)

- Process view reveals relevant properties of underlying private business process
- Balance between interest of owner and client

How to construct a transactional process view from an internal transactional process?
Example classical process view

Company W: Fulfillment Service Consumer

Private Process View
- Select Items
- Send Order
- Set List
- Set Payment
- Receive ItemInfo
- Receive Items

Public Process View
- SendInbound Order
- SendOutbound Order
- RecItemShipmen

Company X: Fulfillment Service Provider

Public Process View
- Receive Order
- RecInbound ItemsOrder

Private Process View
- Recl outbound Order
- Get Items
- Send Items
- International
- Country-side
Classic process views

Example internal transactional process
Approach

- Extend process view construction approach with transactional features
  - Aggregation
  - Customization
Final remarks

Contribution

• Flexible construction of transactional process views
• Specification of robust & reliable behaviour at the external level
• Approach helps to establish dynamic trustworthy B2B collaborations in an efficient way

Future topics:

• More expressive process models
• Execution of transactional process views

Research topics

- Analysis of workflow data models
- Synthesis of object-life cycles
- Transactional process views
- *Splitting GSM schemas*

<table>
<thead>
<tr>
<th>Intra-organizational</th>
<th>Shared database</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Object</td>
<td>Variable</td>
</tr>
<tr>
<td></td>
<td>Analysis of workflow data models</td>
<td>Synthesis of object-life cycles</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Inter-organizational</th>
<th>Shared database</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Object</td>
<td>Variable</td>
</tr>
<tr>
<td></td>
<td>Transactional process views</td>
<td><em>Splitting GSM schemas</em></td>
</tr>
</tbody>
</table>
Guard-Stage-Milestone (GSM) schemas

- **Guard**
  - business rule when to open stage

- **Stage**
  - cluster of work to achieve milestone

- **Milestone**
  - business objective

- Status of each stage, milestone ∈ \{ on, off \}
- Business rules specify when stage/milestone changes state
  - Might refer to status of other stages/milestones
- Rules need to be evaluated in right order to ensure that all changes have maximal effect
- Unit of change triggered by external event is called a B-step
Splitting GSM subschema
(Joint work R. Hull, Y. Sun, R. Vaculin from IBM Research)

Client artifact system

Refine Deal

Draft Term

Term Drafted

Check Credit

Credit Checked

Determine Price

Price Determined

Provider artifact system

ReDraftTermNeeded

Deal Failed

Deal Refined

ReCheckCreditNeeded

ReDraftTermNeeded

ReCheckCreditNeeded

Refine Deal

Check Credit

Credit Checked

Determine Price

Price Determined

must be equivalent
Naïve splitting causes race conditions

Task
CheckCredit completes
credit_level = 200,000

open if TermDrafted and CreditChecked

achieve if credit_level > 100,000

open on +ReDraftTermNeeded

Deal Failed
Deal Refined

status is on
status is off

Check Credit
Credit Checked
Determine Price
Price Determined

ReCheckCreditNeeded

Draft Term
Term Drafted

ReDraftTermNeeded

LawOffice

IGF

Technische Universiteit
Eindhoven
University of Technology
Naïve splitting causes race conditions

- status is on
- status is off

Task
Check Credit
completes
credit_level = 200,000

- open if TermDrafted and CreditChecked
- Impossible in original GSM schema

achieve if credit_level > 100,000

open on +ReDraftTermNeeded
Possible solutions

1. Extend local GSM engines with complex synchronization logic
2. Better: **restructure** GSM schema to avoid complex synchronization
Solution approach

- Keep local cache of status of needed remote stages/milestone
  - determined per external event type
- Reallocate business rules to avoid complex synchronization
  - Sometimes: rule executes at different location than the corresponding stage/milestone
- Runtime simulation protocol based on distributed 2-phase commit
- Result (see paper): split GSM schema faithfully simulates original GSM schema
Final remarks

• Comprehensive framework for supporting outsourcing of GSM schemas
• Covers both design-time and run-time
• Future work:
  • multi-party outsourcing
  • interacting artifact types
  • extend results to OMG’s Case Management Modeling Notation

Conclusion

Future work
- Inter-organizational perspective
- Case management
For more info …

Contact details

• Email: h.eshuis@tue.nl
• Homepage: http://is.ieis.tue.nl/staff/heshuis

Thank you!