Synchronizing Business and IT Process Views with BPMN

Hagen Völzer
Joint work with Jochen Küster and Cédric Favre
Introduction: A Business Rule in IBM WODM

- Uses connection to a repository of business objects (and their attributes)
Agility through Artifact Separation

Business Rule (Textual Requirement)

Business

yesterday

Implementation of rule (Code)

IT System

today

Rules Engine

IT System

Advantages:

– Change- and maintenance friendly
– Less development effort
– What you see is what you execute

Hagen Völzer
When is such an approach successful?

Characteristics of (such) business rules:

- Rules are meaningful to business
- Can be read/created directly by business in a way that is understood by a dedicated engine
- Rules change frequently
Can this be done for other artifacts/aspects as well?

- **Characteristics for rules:**
  - Rules are meaningful to business
  - Can be read/created directly by business in a way that is understood by a dedicated engine
  - Rules change frequently

- **Other artifacts:**
  - Forms / UI
  - *Process* / control flow (main business behavior)
  - KPI / monitoring data
  - Business events
  - Data model / information structure / business objects
  - ?

- How successful / realistic is the idea for these artifacts?
Outline

- The Business-IT gap problem for process models

- A solution approach: The *Shared Process Model* (work in progress)
  - Basic solution design
  - Recorded demo of a prototype
  - Our notion of consistency

- Discussion
The Case of Process Models

- **Disadvantages:**
  - Models, being maintained independently, quickly get out of sync (inconsistent)
  - What you see is not what is executed (audit failures, ineffective operations, misinterpreted monitoring data)
  - Not change friendly
Road Block 1
(yesterday)

- Different languages and tools on business and IT level
  - BPMN / ARIS
  - BPEL

- Transformation from Business to IT is possibly automated
  - No bidirectional transformations
  - Inconsistencies quickly occur and cause problems

- Today: Can use BPMN 2.0 as common language for business and IT
Road Block 2: Different Concerns and Levels of Detail

- Current Practice:
  - Companies use multiple models
  - Is this necessary?

- Case study [Branco et al., Uni Waterloo]
  - 70 model pairs from a single bank
  - 23 interviews (business and IT)

- Study differences in model pairs (business – IT)
  - Identified change patterns and their frequency
  - Other findings regarding consistency (later)

- In addition, we talked to other BPM practitioners and architects
Business and IT want different models/views (1/2)

IT model has compared to business model:

- Complementary implementation detail (data, services, communication)
- Formalization and renaming (task/event specialization, condition formalization) – no alteration of flow
- Contd.
Business and IT want different models/views (2/2)

IT model has compared to business model:

- Behavioral refinement and refactoring
  - Hierarchical refinement / subsumption
  - Hierarchical refactoring (for maintenance and performance)
  - Added IT tasks

- Added behavior (e.g. technical exception handling)

- => Organizations use multiple models of the same process (The difference between them is technically more than just “omitting some details”)
Multiple models get inconsistent

- **Scenario 1: IT Refinement / Implementation**
  - IT may change business-relevant aspects, resulting in inconsistency with business view, because:
    - Business view is incomplete (frequently)
    - Business view contains inconsistencies (frequently to occasionally)
    - Business view contradicts some IT requirements, e.g. order of tasks (occasionally)
    - Business view does not anymore faithfully represent the actual business process (rarely)

- **Scenario 2: Business Change**
  - Business makes changes to the business view due to new business requirements (Case study: *roughly 2 fixes per project per year*)

- **Scenario 3: Technical Change**
  - New IT requirements cause change of the Technical view
    - E.g. upcoming IT infrastructure changes require changes in executable process models
    - Some changes may affect business-relevant aspects
Inconsistencies cause problems

- Shipment delays, business disruptions, audit failures

*From the case study:*

- E.g. some functionality was deleted inadvertently from IT model because there was no correspondence in the business model
  - IT implementation was outsourced
  - Discovered after process was in production
  - Running instances had to be canceled and recreated

- E.g. similar scenario but now
  - inconsistency between audit model and running process caused audit issue and a fine
Outline

- The Business-IT gap problem for process models
- A solution approach: The *Shared Process Model* (work in progress)
  - Basic solution design
  - Recorded demo of a prototype
  - Our notion of consistency
- Discussion
Solution: The Shared Process Model

- Each view can be changed
  - *Common* changes to one view are propagated to the other view (but not *private* changes)
  - Links are maintained between model elements to propagate change propagation
The Shared Process Model – Implementation Approach

- Maintain internally two separate BPMN models and *links* between their elements
  - Links are created automatically during refinement through refinement patterns and must be maintained
  - Links are used to propagate changes from one side to the other and to check consistency

- Advantages of this approach:
  - View creation is simple, export BPMN model, which can be consumed by editors and other tools
  - Generalization to more than two views relatively straight-forward
Correspondence Links

- Links are 1:1, 1:m, m:1, but could be m:m as well
- Some elements are not linked (private elements)
Basic Change Propagation Mechanism

- Create horizontal diff between IT models (generic model comparison)
- Translate diff from business to IT using links
- Apply translated diff to business model to obtain new business model
  - Links need to be maintained

```
InsertFragment(f, 'Get Request Details', 'Log Session Data')
InsertActivity('Check consistency With Records', Merge, Decision).
MoveActivity('Get Request Details', Merge, 'Check Consistency with Records')
```
Refined Change Propagation Mechanism

- Each operation of the diff is applied atomically to create a modified business model and to update the links
Recorded Demo
Outline

- The Business-IT gap problem for process models
- A solution approach: The *Shared Process Model* (work in progress)
  - Basic solution design
  - Recorded demo of a prototype
  - Our notion of consistency
- Discussion
Business-IT Consistency

What you see (in business view) is what you execute (in IT)

- Intuition: Links always represent hierarchical refinements/subsumptions
- Private elements are `between' links (not within)
- In IT, exceptional behavior may be added
Inconsistency Examples

(a) 

(b) 

(c) 

(d)
Business-IT Consistency

- Dedicated hierarchical refinement actions preserve consistency
- Common changes preserve consistency
- For private freehand-editing, consistency can be checked upon check-in
- Consistency can be checked efficiently
  - because it is on the defined directly on the model

Consistency Implies that `business behavior’ is preserved (but not the converse)
Consistency Management

- Change Management on top of Shared Model based on consistency is desirable (future work)
Conclusion

- A single process view is often not enough
- Multiple views quickly go out of sync
- Multiple views can be kept in sync with the Shared Process Model – i.e., through vertical model synchronization
Outlook to Future Work

- Extensive practical evaluation
- Alternative implementation for simple scenarios: Extended meta-model
- Advanced scenarios:
  - Change Management on top of Shared Model
  - One business model corresponds to multiple alternative IT models
  - More than two views
Can this be done for other artifacts/aspects as well?

- **Characteristics for rules:**
  - Rules are meaningful to business
  - Can be read/created directly by business in a way that is understood by a dedicated engine
  - Rules change frequently

- **Other artifacts:**
  - Forms / UI
  - Process / control flow (main business behavior)
  - KPI / monitoring data
  - Business events
  - Data model / information structure / business objects
  - ?

- How successful / realistic is the idea for these artifacts?
References


- C. Favre, Jochen Küster, and Hagen Völzer: The Shared Process Model. BPM Demos 2012