

Modelling of Context-Adaptable Business Processes and their Implementation as Service-Oriented Architecture

Vojtěch Mates, **Marek Rychlý**, Tomáš Hruška

Department of Information Systems
Faculty of Information Technology
Brno University of Technology
(Czech Republic)

17th Annual International Conference on
Enterprise and the Competitive Environment
6 – 7 March, 2014



- 1 Introduction
 - Flexible Business Processes
 - Context-Adaptability in Business Processes
 - The Approach to Context-Adaptable Business Processes
- 2 Context-Adaptable Business Processes
 - Modelling the Context-Adaptable Business Processes
 - Product Dependency Tree and its Building
 - Implementation as Service-Oriented Architecture
- 3 Summary and Future Work



Flexible Business Processes

- business processes can be realised in different ways
(same products but different realisations driven by resources and conditions)
- difficult to select an optimal realisation of a business process
- the optimal realisation is not stable, neither in time nor in context
- context adaptability needed in order to build flexible processes

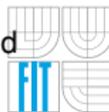
A research on data describing changes in project scheduling:

- almost all projects (over 80%) rescheduled due to
 - missing the deadlines of sub-processes of the projects or
 - temporary unavailability of resources required by realisations
- the solution was to change some parts of the projects



Context-Adaptability in Business Processes

- various approaches to adaptable business processes
 - Rule-based – predefined rules, how to react in particular situations (limited by rules, difficult flexibility and evolution)
 - Case-based – cases describing the process variants (high variability in the cases, difficult generalisation)
 - Declarative – logical formulas limiting the process variants (difficult to understand, especially in negative forms)
 - Goal-based – hierarchy of goals, often used in logic (focused of goals, not realisation, difficult to implement)
- a combined approach needed with the following requirements:
 - rule-based with rules automatically derived from known cases, (adaptable to contextual changes optimally, based on statistical analysis)
 - declarative by products (goals) and limits of the production, (suitable for both managers, who are focused on process effectiveness, and business process engineers, who are focused on process efficiency)



Context-Adaptable Business Processes

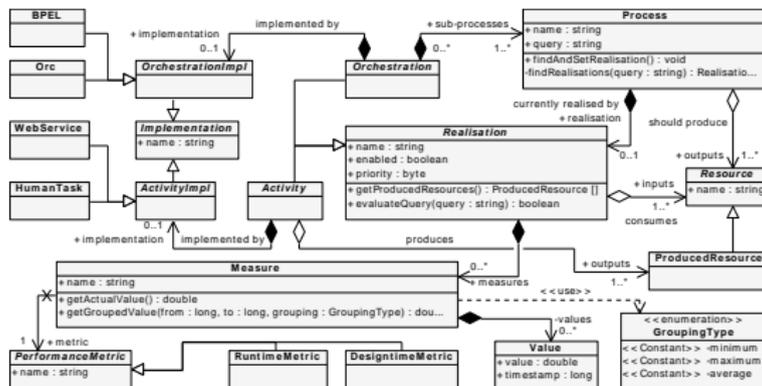
the approach proposed in the paper

- An approach to design, realisation, and implementation of context adaptable business processes.
- More specifically, the approach describes:
 - a **meta-model** for modelling the context adaptable processes,
 - a **product dependency tree** able to describe all possible complete realisations of a particular business process,
 - an **algorithm for building the product dependency tree** based on process abstractions and realisations, and on required products (goals) and user-defined production limitations and preferences,
 - a **mapping of the product dependency tree to implementation** of business processes as service-oriented architectures adaptable to runtime contextual changes by observing predefined KPIs.



Meta-model of Context-Adaptable Business Processes

- a conceptual model of context-adaptable business processes
- it describes a **Process** and **Resources** which it should produce
- a process's **Realisation** as an **Orchestration** or an **Activity**
- a realisation's consumed **Resources** and its relevant **Measures**
- an orchestration's **Sub-processes** and activity's **ProducedResources**
- an **Implementation** of an orchestration or an activity
(by **BPEL** and **Orc**, or by **WebService** or **HumanTask**, respectively)



Product Dependency Tree (PDT)

- a directed acyclic graph $G = (N, E)$ where each node $n \in N$ represents a process realisation at a particular level of details (according to a depth of the node in the tree)
- the root node = a business process without any assigned realisation
- sibling nodes represent different **process realisations** of all processes unrealised in their parent node, i.e., **process abstractions**
- leaf nodes represents complete decomposition of the root's process to hierarchy of sub-processes decomposed to the level of atomic activities
- child nodes are assigned to parent nodes by the **product dependency** (a process realisation in each child node has to produce the same products as should produce a process abstraction in its parent node)
- each branch represents particular way how to achieve goals of the root's business process and lists all resources needed



Building Product Dependency Tree (PDT)

- PDT represents all ways to realise a particular business process
- some ways are forbidden or preferred, they need not to be built (depends on required resources, on efficiency of realisations, etc.)
- PDT can be built automatically, a user just needs to provide
 - 1 all possible realisations of all possible (sub-)processes, each including a list of products and resources
 - 2 a main business process (= a root of PDT)
 - 3 restrictions, limitations, and preferences

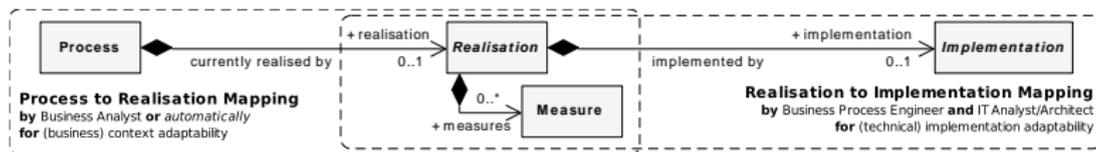
(the first is defined in a database of processes, the next two are defined by a user in an SQL-like query according to desired products and production efficiency)

```
/* a process variable name, the source for the query */  
SELECT p FROM processDatabase  
/* final product definition (a process abstraction) */  
WHERE p.finalProducts = "the_real-time_network_simulation"  
/* constraints of variables based on historical performance of p */  
AND (p.responseTime < 1 sec) AND (p.cost <= 2 EUR)  
/* order the results according to their costs and return the first */  
ORDER BY p.cost LIMIT 1;
```



From Process Realisations to Implementations in SOA

- one of branches of PDT has to be selected for implementation (the branch which is currently the most optimal implementation)
- to select optimal realisation, performance of activities (atomic parts of the realisation) have to be measured and evaluated
- in the case of SOA, activities are Web services or Human tasks
- several criteria to measure performance and QoS of the services (availability and reliability of the service, price, throughput, response time, latency, performance, security, accessibility, regulatory, robustness/flexibility, accuracy, servability, integrity, and reputation)
- switching branch of PDT at runtime may be expensive, switching implementation of activities is often sufficient/should be preferred (changeover costs of switching from current process realisation to another)



Summary and Future Work

- The meta-model for modelling the context adaptable processes.
- The product dependency tree to describe all possible realisations of a particular business process.
- The algorithm for building the PDT which utilises a process database and a user-defined query.
- The mapping of the PDT to an implementation in SOA.

Future work

- combined service performance properties and process input resources (different performance according to its different input configurations)
- evaluation of the approach and implementation of software tools



Thank you for your attention!

Marek Rychlý

<rychly@fit.vutbr.cz>

