Integrating Context-Awareness in Service-Oriented Healthcare Applications

Dorina C. Petriu

Carleton University
Department of Systems and Computer Engineering
Ottawa, Canada, K1S 5B6
http://www.sce.carleton.ca/faculty/petriu.html

Research Team

Name of student / PDF and email address	Program MSc/PhD/ PDF/ u-grad	Task # (Specify if PRP)	hSITE Start date	Graduation date (actual or anticipated)	Funded by: hSITE and/or other
Alhaj, Mohammad malhaj@sce.carleton.ca	Ph.D.	1.2.2 and 2.1.1	May 2009	August 2012	hSITE
Mani, Nariman nmani@sce.carleton.ca	Ph.D.	2.1.1	Jan 2010	August 2013	other
Khan, Muhammad Kaleem mkalim@sce.carleton.ca	M.App.Sc.	1.2.2	Sept 2010	Dec. 2011	hSITE + other
Vrbaski, Mira mvrbaski@connect.carleton.ca	M.App.Sc.	2.1.1	Sept 2009	Dec. 2011	part-time student
Gunter Mussbacher gunter@sce.carleton.ca	postdoc	2.1.1	Jan 2011	Dec 2011	other + hSITE

Tasks and Objectives

- **Theme 2, Project 2.1, Task 2.1.1:**
  - i. ***Integrating context awareness in Service-Oriented Architecture (SOA).*** **The objective is to implement context-awareness at the application level (not only at the network level). The applications are executed by composing reusable services at runtime.**
  - ii. ***Investigating Performance Effects of SOA design patterns*****. Addressing the problem of service architecture quality by applying SOA design patterns from literature. Each design pattern aims to improve a given software characteristics (be it functional or non-functional) and has performance side-effects, which are evaluated with the help of performance models.**
  - iii. ***Deriving automatically performance models from SOA software models*****. continuing the work presented last year by Ph.D student M. Alhaj. The model transformation algorithms has been designed and now are being implemented.**
- **Theme 1, Project 1.2, Task 1.2.2:**
  - iv. ***Applying Model-driven SOA.*** **The objective is to apply model-driven techniques and tools for developing service-based systems that implement health-care workflows identified in Theme 1. For now we have been experimenting with tools for model-driven SOA.**

i. Integrating context-awareness in service-oriented healthcare applications

Requirements for IT in healthcare

- **According to the paper:**

  Erin Yu, Ryan Kealey, Mark Chignell, Joanna Ng, and Jimmy Lo, "Smarter Healthcare: An Emergency Physician View of the Problem", in M. Chignell et al. (Eds.): The Smart Internet, LNCS 6400, pp. 9–26, 2010.

  the following are some IT shortcomings in emergency medicine:

  a. **A lack of integration from the user's perspective**
  b. **A lack of individualization & context-awareness**
  c. **Lack of server-initiated connections**
  d. **Lack of service-level collaboration**
  e. **Limited user control**

- **Our proposed solution to solve some of these shortcomings:**
  - **Adopt Service-Oriented Architecture (SOA) → addressing (a), (d)**
  - **Integrate context-awareness in SOA → addressing (b), (c), (e),**

Context-aware SOA

- **SOA (Service-Oriented Architecture) paradigm:**
  - promotes the idea of composing applications from loosely coupled reusable services to create flexible, dynamic business processes and agile applications that span organizations and computing platforms.
- **Context-aware SOA:**
  - integrating context-awareness in SOA by means of special services for:
    - acquiring and monitoring the context of different entities
    - abstracting and understanding the context
    - providing context information to other services when needed
    - triggering actions based on the context
  - context-aware services make use of different level of contexts and adapt the way they behave according to the current context based on context rules
  - context-aware services are composed at runtime with the purpose of executing context-aware applications described by business workflows

Context management services

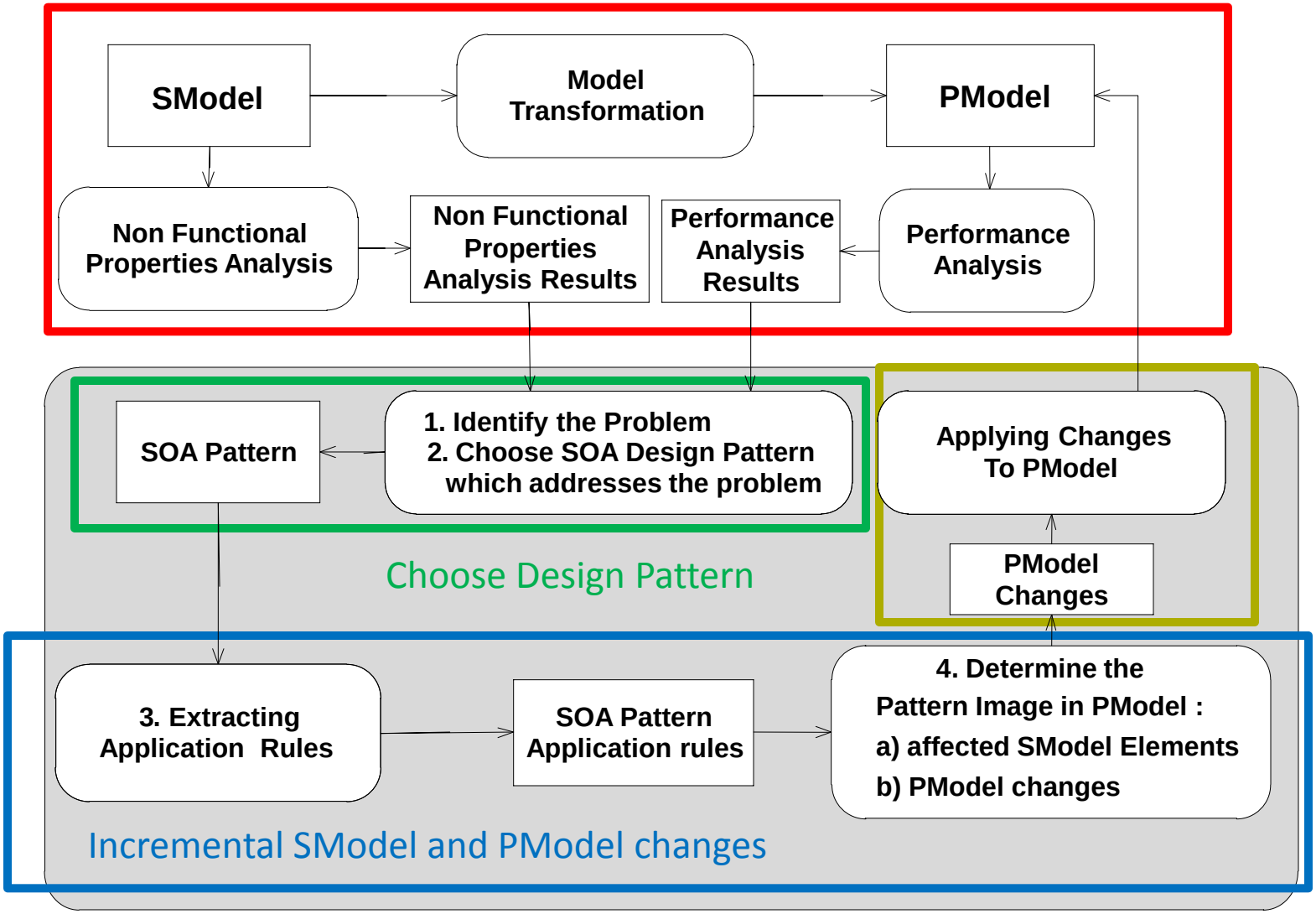
- The proposed context management framework includes the following services:
  - context provider service: collects raw context information from various sources and translates it to low level context information
  - context aggregation service: receives requests for context information and finds which context provider service can offer it
  - context notification service: uses a publish-subscribe mechanism to provide notification to clients when the context changes
  - context reasoning service: provides results based on context rules which allow to tailor the action to be taken to the actual context value
  - context discovery service: mapping between context provider service and context information.
- The proposed context management system is implemented using a number of open-source components and tools, as presented in Mira Vrbaski's poster.

ii. Investigating performance effects of SOA design patterns

Objective

- Service Oriented Architecture (SOA) design patterns provide generic solutions for many architectural, design and implementation problems
  - any pattern may have an impact on performance, either positive or negative.
- Objective: study the performance impact of a SOA design pattern applied to a system in early development phases
- The planned approach exploits the context of model driven engineering (MDE): SModel →PModel
  - PUMA model transformation chain is used to generate the initial PModel of the system
  - A SOA design pattern is applied to SModel and the change is propagated incrementally to PModel.

Overview of the Proposed Approach

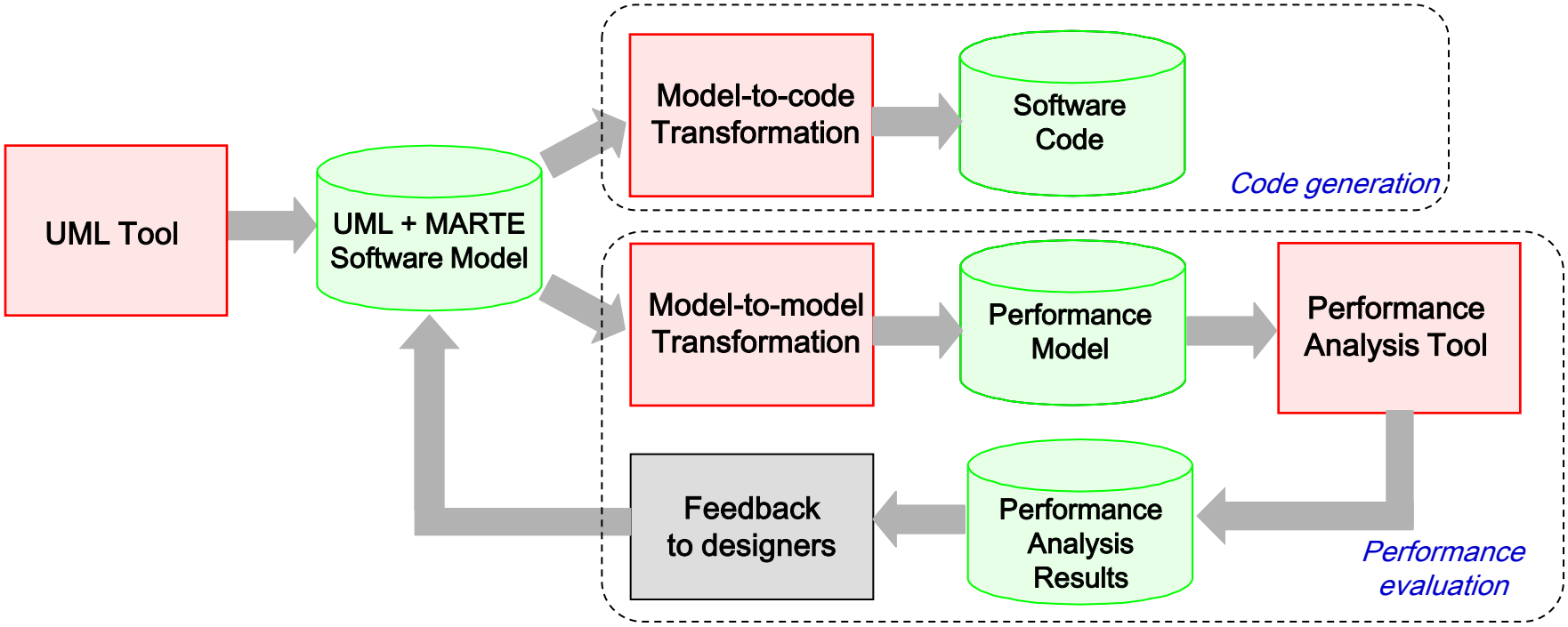


Research status

- Concerned with the quality of a service-oriented system, which can be improved by applying SOA design patterns.
- Propose an approach to propagate changes due to the application of SOA design patterns from the SModel to the corresponding PModel
  - incremental model transformation to speed up the change propagation
- Current status
  - preliminary experiments demonstrated feasibility
  - these are by no means definitive and a general approach is still to be developed.
- Future work
  - fully develop a general approach for incremental change propagation
  - apply it to many SOA patterns from literature
  - automate incremental change propagation from SModel to PModel for different patterns by using traceability links
  - screen automatically different solutions for improvements.

iii. Deriving automatically performance models from SOA software models

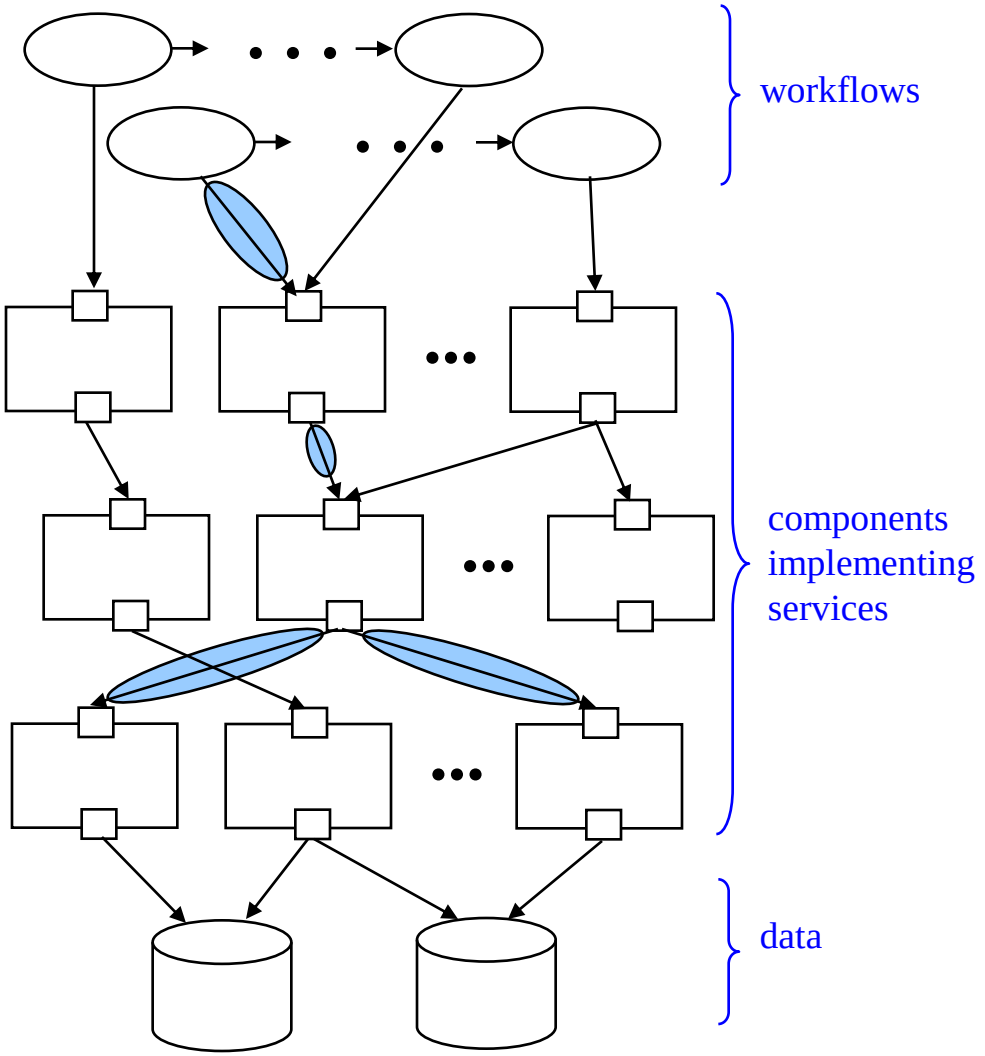
Model-driven software engineering



- **Software performance/dependability evaluation in the context of Model-Driven Engineering:**
  - **starting point: UML software model used also for code generation**
  - **add performance annotations (using specialized profiles such as MARTE)**
  - **generate a performance/dependability analysis model**
    - **queueing networks, Petri nets, stochastic process algebra, Markov chain, fault tree, etc.**
  - **solve analysis model to obtain quantitative results**
  - **analyze results and give feedback to designers**

Performance Analysis of SOA

- PUMA4SOA model transformation
  - Source: UML+MARTE model
  - Target: performance model (LQN)
- The source model contains:
  - workflow model
  - service architecture model (dependencies, components)
  - service behaviour model
  - deployment
  - middleware overheads
- Model transformation steps:
  - Aspect-oriented approach for adding middleware overheads
  - Transformation 1: from source model to Core Scenario Model (CSM)
  - Transformation 2: from CSM to performance model (LQN)



Conclusions

- Our research is in the software engineering area, at the confluence of the following sub-areas:
  - Service-Oriented Architecture application to healthcare
  - Integrating context awareness within SOA applications and services
  - Enhancing SOA quality through SOA design patterns
  - Verification of SOA performance and dependability based on quantitative models generated form the software models
  - Model-driven development of service-oriented systems.
- We are getting ready to develop case-studies of service-oriented systems which support healthcare workflows that are the outcome of Theme 1.
  - will use model-driven development techniques and tools that change the development focus from code to models, raising the level of abstraction.