Context-Aware Service-Oriented Systems

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. 2012	hSITE + other
Vrbaski, Mira mvrbaski@connect.carleton.ca	M.App.Sc.	2.1.1	Sept 2009	Sept 2012	part-time student
Gunter Mussbacher gunter@sce.carleton.ca	postdoc	2.1.1	Jan 2011	Dec 2012	other + hSITE

Objectives

Theme 2, Project 2.1, Task 2.1.1:

1. ***Integrating context awareness in Service-Oriented Architecture (SOA)***
   - **Last year: designed a framework composed of open-source components for integrating context-awareness within SOA**
   - **New: *Context-Aware Reasoning using Goal-Orientation (CARGO)* - extend the context-aware reasoning approach based on rules with goal-oriented models evaluated at runtime, to provides more flexibility and configurability**
     (see poster by Mira Vrbaski and Gunter Mussbacher)
2. ***Investigating Performance Effects of SOA design patterns*. Addressing the problem of service architecture quality by applying SOA design patterns.**
   - **Each design pattern aims to improve a given software characteristics (functional or non-functional) and has performance side-effects, which are evaluated with the help of performance models.**
3. ***Automatic derivation of performance models from SOA software models*.**
   - **This year, the model transformation has been enhanced by considering separately the platform effects modeled as aspects.**

1. Context-Aware Reasoning using Goal-Orientation (CARGO)

Context

- **Context definition: any information that can be used to characterize the *situation of an entity*.**
  - **entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and the application themselves.**
- **Context includes the following *environmental aspects*:**
  - **computing environment: available processors, devices accessible for user input and display, network capacity, connectivity, and costs of computing;**
  - **user environment: location, collection of nearby people, social situation;**
  - **physical environment: lighting and noise level.**
- **Context awareness:**
  - **the ability of a system to adapt to an ever-changing context**
  - **proactively anticipate the user's needs without placing the burden on the user**

Context-aware SOA

- **SOA (Service-Oriented Architecture):**
  - **architectural paradigm where applications are composed from loosely coupled reusable services to create flexible 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 levels of contexts and adapt the way they behave according to context reasoning based on pre-defined rules**
  - **context-aware services are composed at runtime with the purpose of executing context-aware applications described by business workflows**

Context-aware Goal Modeling

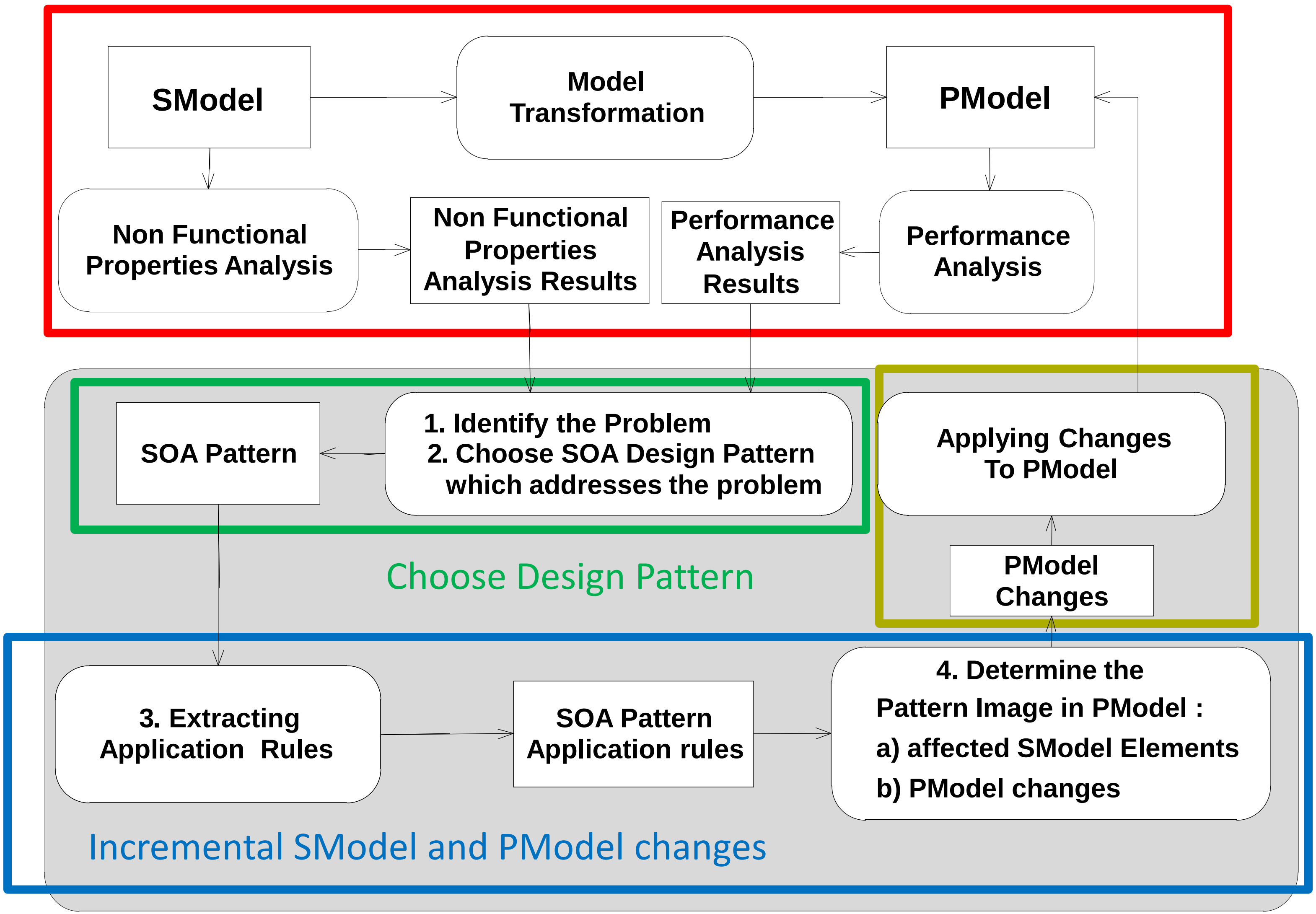
- **Goal modeling is an early requirements technique focuses on:**
  - **modeling stakeholders and their high-level goals;**
  - **modeling solutions and their impact on achieving the goals;**
  - **key performance indicators, i.e., real-world measures that characterize the proposed solutions.**
- **Goal models can be evaluated:**
  - **assessment of a solution results in satisfaction values for stakeholders;**
  - **trade-off analysis compares the proposed solutions taking the satisfaction values of all stakeholders into account.**
- **In context-aware systems:**
  - **a Goal Engine can complement a logic-based Rule Engine by allowing a more holistic assessment of the context while taking the goals of many stakeholders into account;**
  - **key performance indicators capture context-related information, making it available for reasoning at the goal level.**

2. 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
  - general approach for incremental change propagation was developed
  - traceability links have been defined
  - "role-based modeling" is used to formally define the change brought by a pattern
- Future work
  - automate incremental change propagation from SModel to PModel for different patterns by implementing the proposed approach
  - apply it to many SOA patterns from literature
  - screen automatically different solutions for improvements.

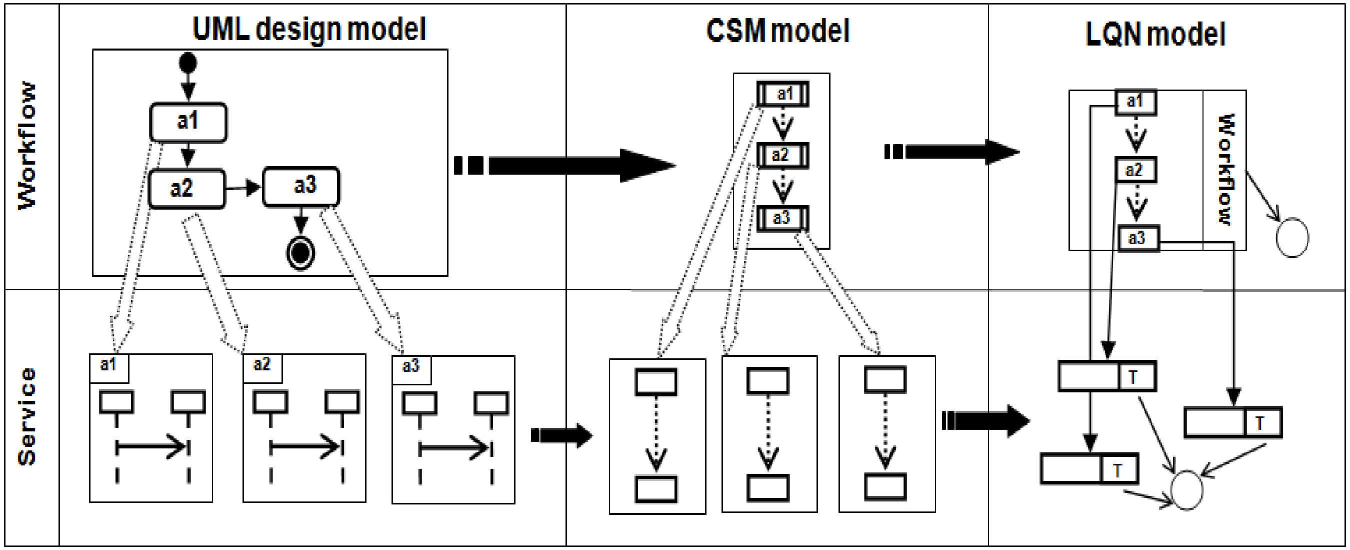
3. Automatic derivation of performance models from SOA software models

Objectives

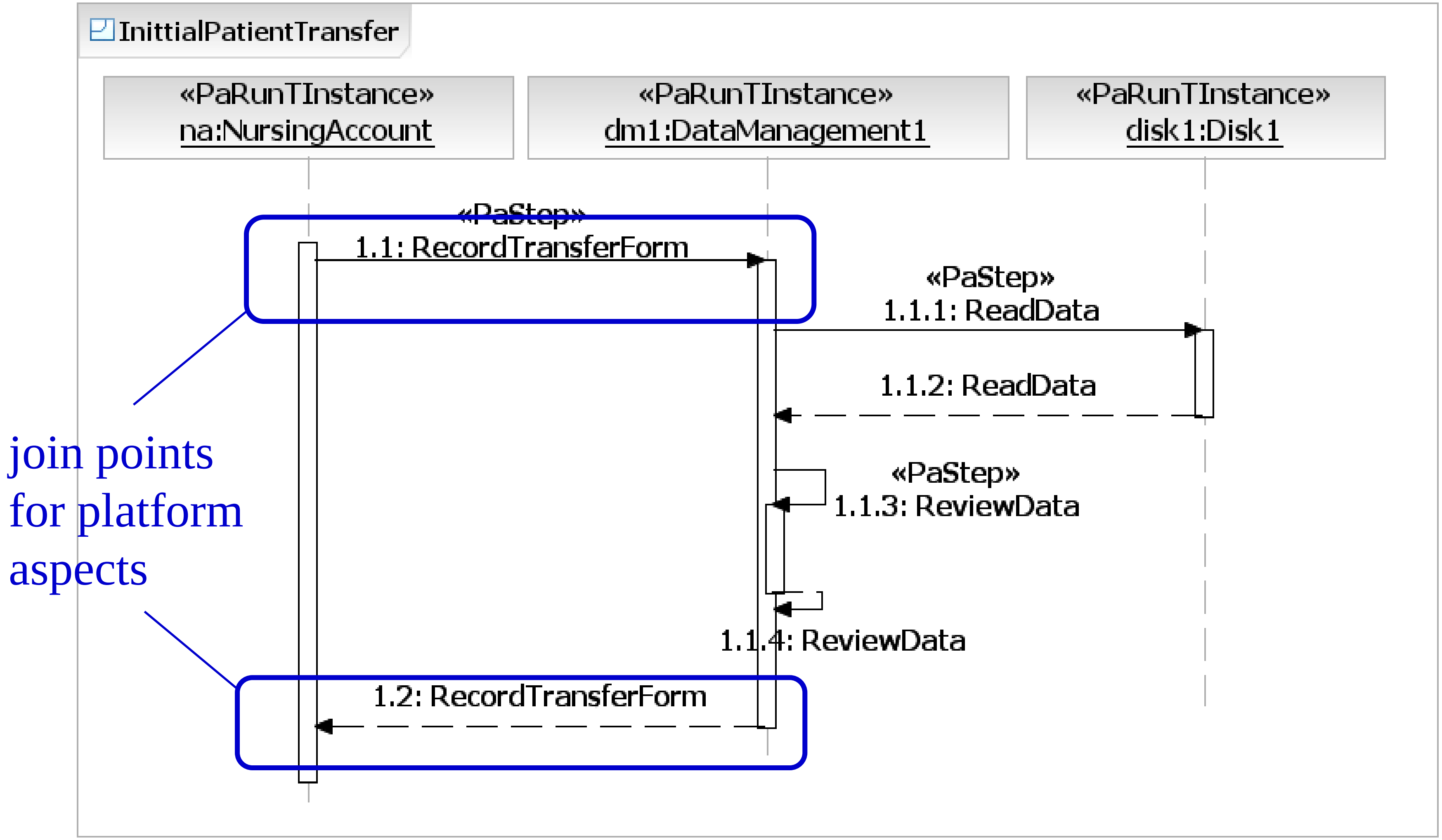
- **Automatic approach for deriving performance models from UML software models to evaluate the run time performance of SOA systems in the early development phases.**
  - **performance models: queueing network, Layered Queueing Networks (LQN) Petri nets, Stochastic Process Algebra**
  - **the software models are extended with performance annotations**
- **Why: early performance evaluation helps in choosing the appropriate architecture and design alternatives to meet the performance requirements.**
- **Separation of concerns when modeling platform overheads**
  - **the starting point is a Platform-Independent Model (PIM) of a SOA system (business workflow and services)**
  - **Platform operations are represented as “aspect models”**
  - **a Platform-Specific Model (PSM) is obtained by weaving platform services into the Platform-Independent Model.**

Transformation chain

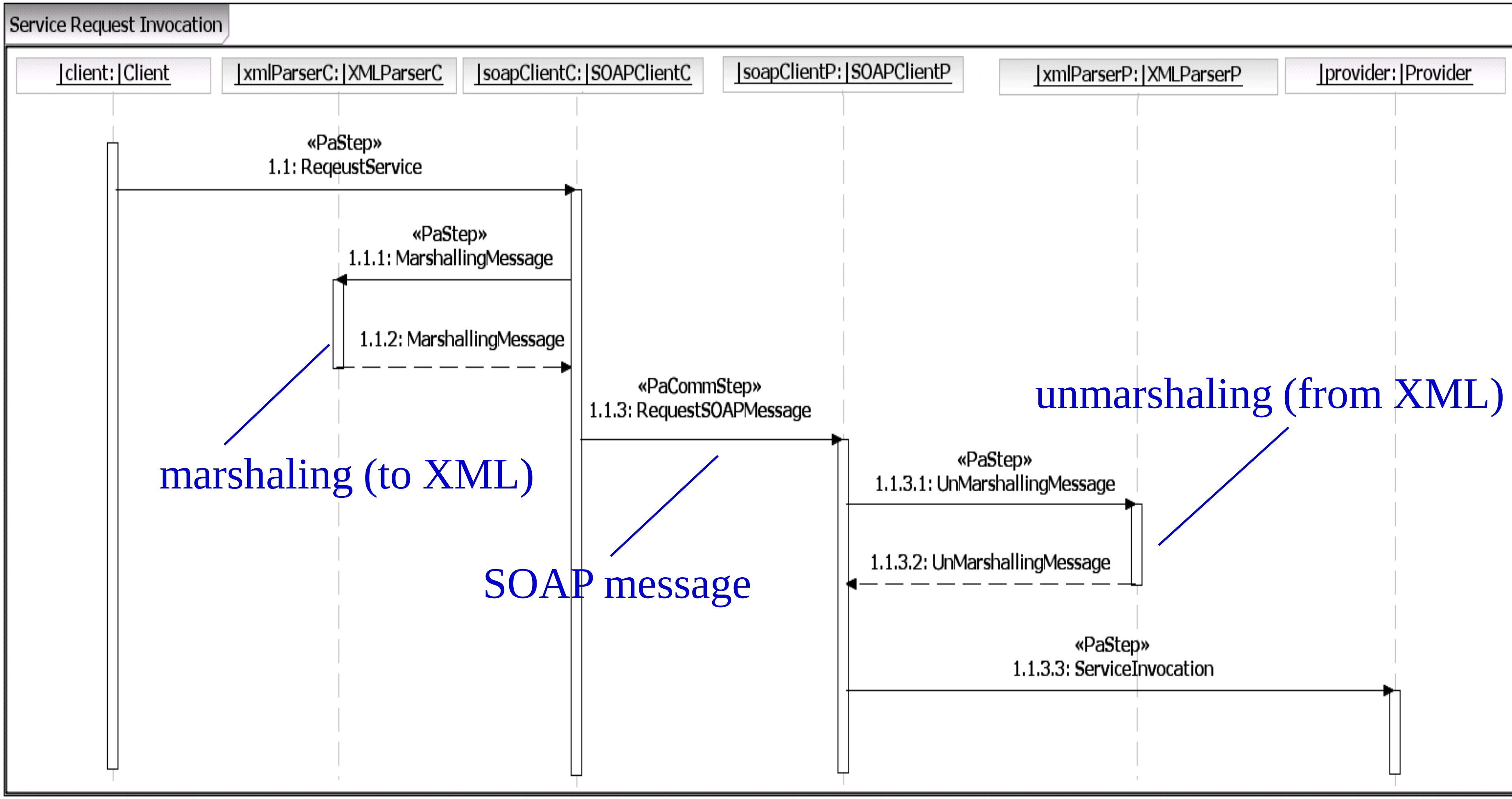
- PUMA4SOA model transformation
  - Source model: UML+MARTE model (structure, behaviour, deployment)
  - Target model: performance model (LQN)
  - Intermediate model: Core scenario Model (CSM)
- Model transformation steps:
  - Aspect-oriented approach for adding middleware overheads
  - Transformation 1: from source model to Core Scenario Model (CSM)
  - Transformation 2: from CSM to performance model (LQN)



Source PIM: Service Behaviour Model

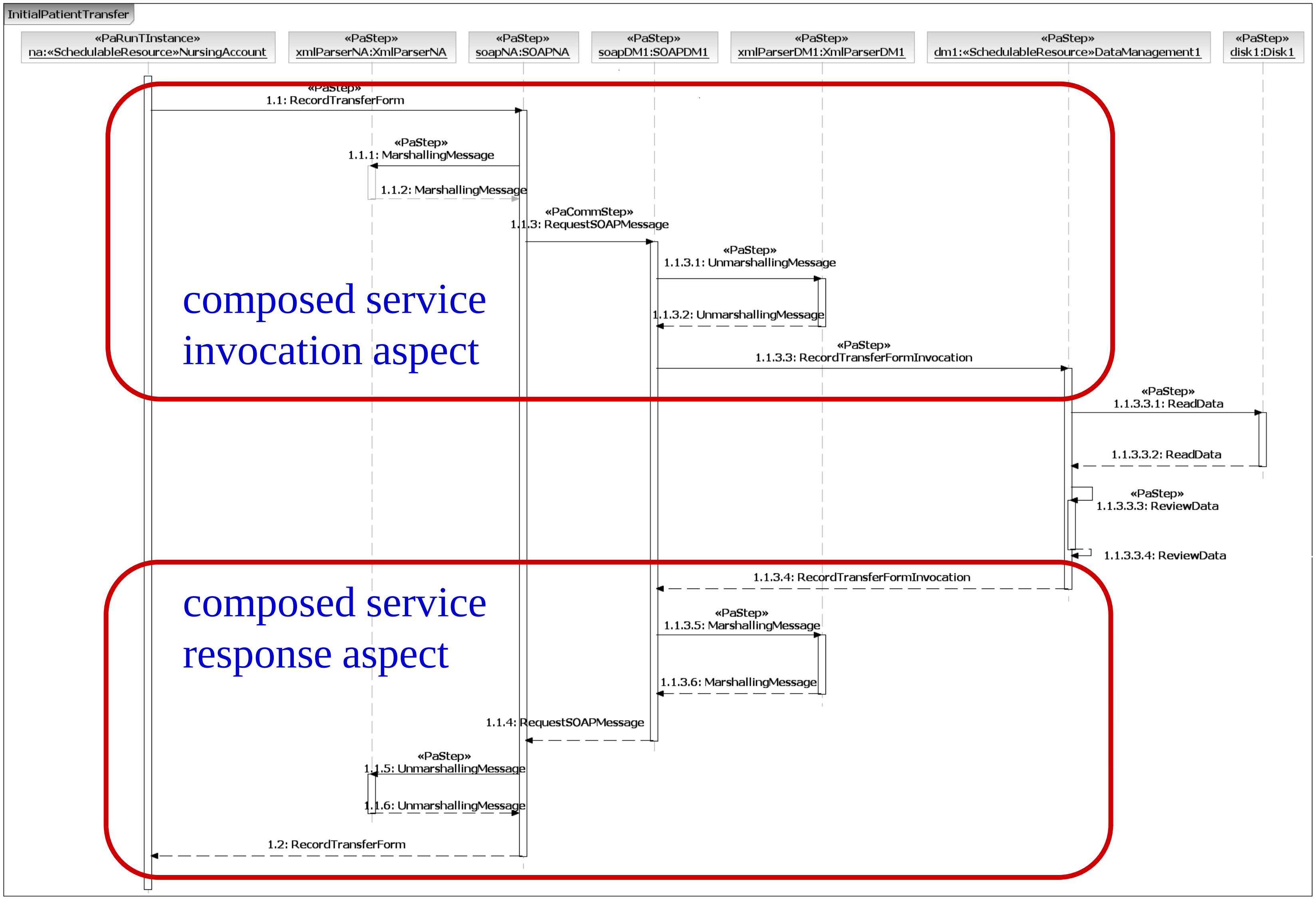


Generic aspect model: Service Request Invocation (behaviour view)

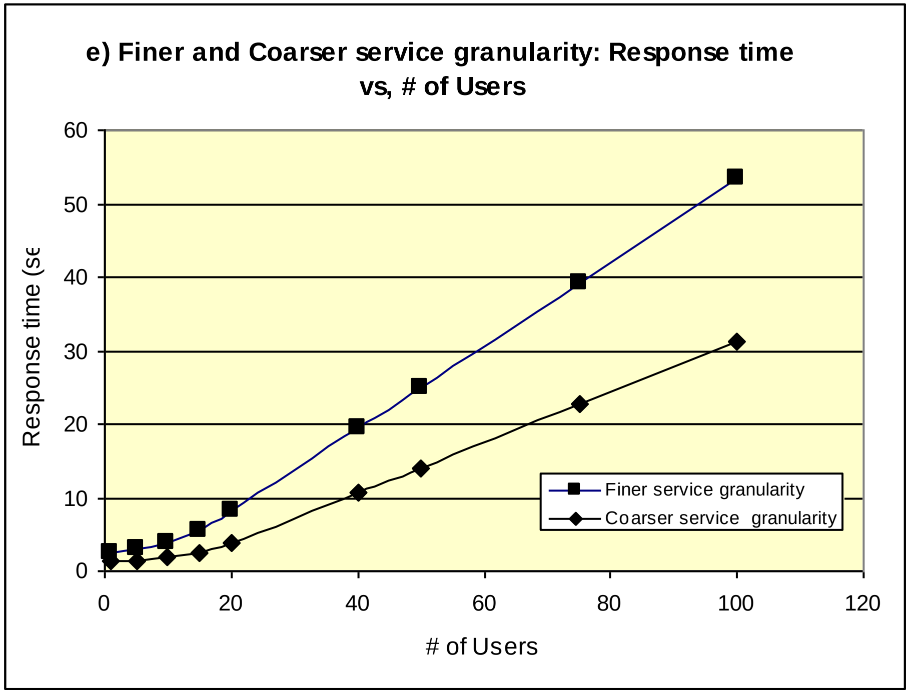


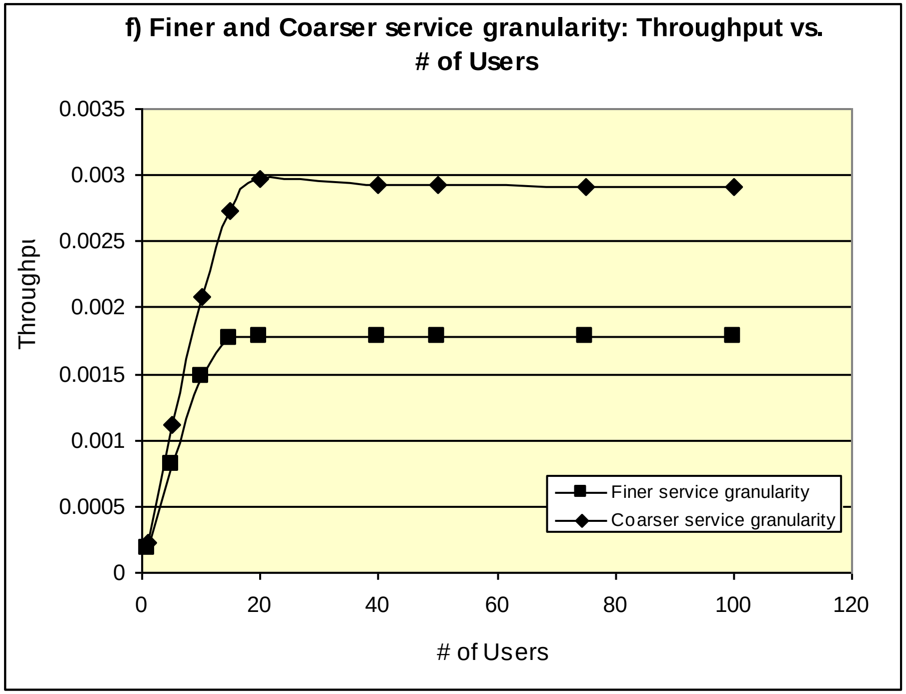
Note: there is a similar model for the service response operation

PSM: scenario after composition



Example of performance results: coarse Vs. fine service granularity





- The compared configurations are similar in number of resources (processors, disks and threads) except that the second performs fewer service invocations through the web service middleware.
- The difference in response time and throughput is due only to the difference in platform overheads

Conclusions

- **We conduct research is in the software engineering area, at the confluence of the following sub-areas:**
  - **Service-Oriented Architecture application to healthcare**
  - **Context aware SOA enhanced through goal models**
  - **Enhancing SOA quality through SOA design patterns**
  - **Verification of SOA performance and dependability based on quantitative models generated form the software models**
- **Collaboration for Years 4 and 5 to integrate multi-sensor fusion algorithms developed at the University of Ottawa with the context aware SOA framework.**
  - **Build multi-sensor fusion services using lower-level context aware services which, in turn, can be invoked from higher level services or business processes.**