hSITE Annual Research Review

Friday, June 4, 2010, Montreal, QC

Dorina C. Petriu

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

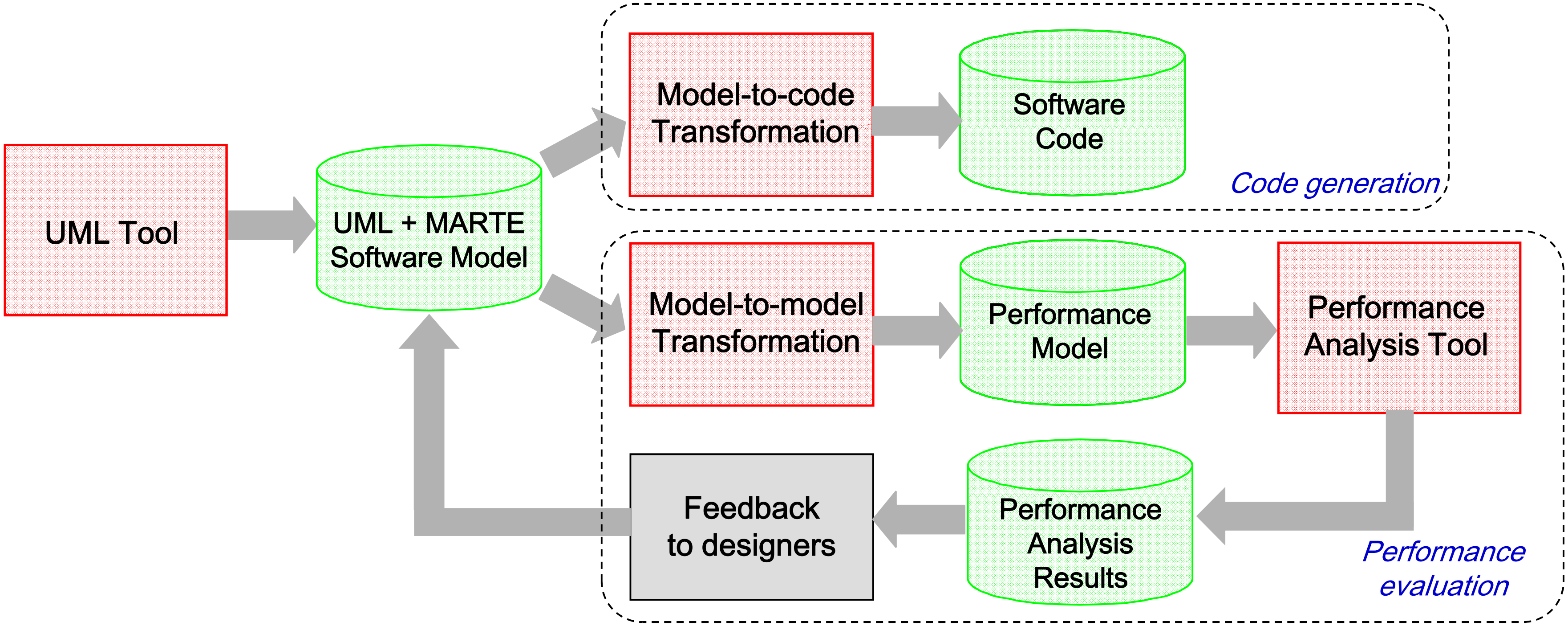
Tasks and Milestones

- Met the objectives for the following milestones:
  - [Task 1.2.2, M1.7] Formal specification of three-layered system architecture
  - [Task 2.1.1, M2.2 a] Definition of a UML profile for dependability/availability annotations that extends the generic Quantitative Analysis model from MARTE.
- Currently working on the following milestones:
  - [Task 2.1.1, M2.1] Development of methodology to assess SOA quality according to functionality partitioning quality
  - [Task 2.1.1, M2.2 b] Development of model transformation techniques to build performance models from UML models of SOA systems with performance annotations

Research Team

Name of student/PDF and e-mail	Program	Task Number	hSITE Start date	Expected Graduation Date	Funding
Mohammad Alhaj malhaj@sce.carleton.ca	Ph.D.	1.2.2, M1.7 2.1.1, M.2.2.b	May 2009	August 2012	2009: other 2010: hSITE
Mira Vrbaski mvrbaski@sce.carleton.ca	M.App.Sc.	2.1.1, M.2.1	Sept. 2009	August 2011	No funding (part-time)
Nariman Mani nmani@sce.carleton.ca	Ph.D.	2.1.1, M.2.1	Jan. 2010	May 2013	2010: other
M. Kaleem Khan mkhan@connect.carleton.ca	M.App.Sc.	2.1.1, M.2.2.c	Sept. 2010	May 2012	2010: hSITE

Approach to Software Performance / Dependability Analysis


UML Tool
UML + MARTE Software Model
Model-to-code Transformation
Software Code
Code generation
Model-to-model Transformation
Performance Model
Performance Analysis Tool
Feedback to designers
Performance Analysis Results
Performance evaluation

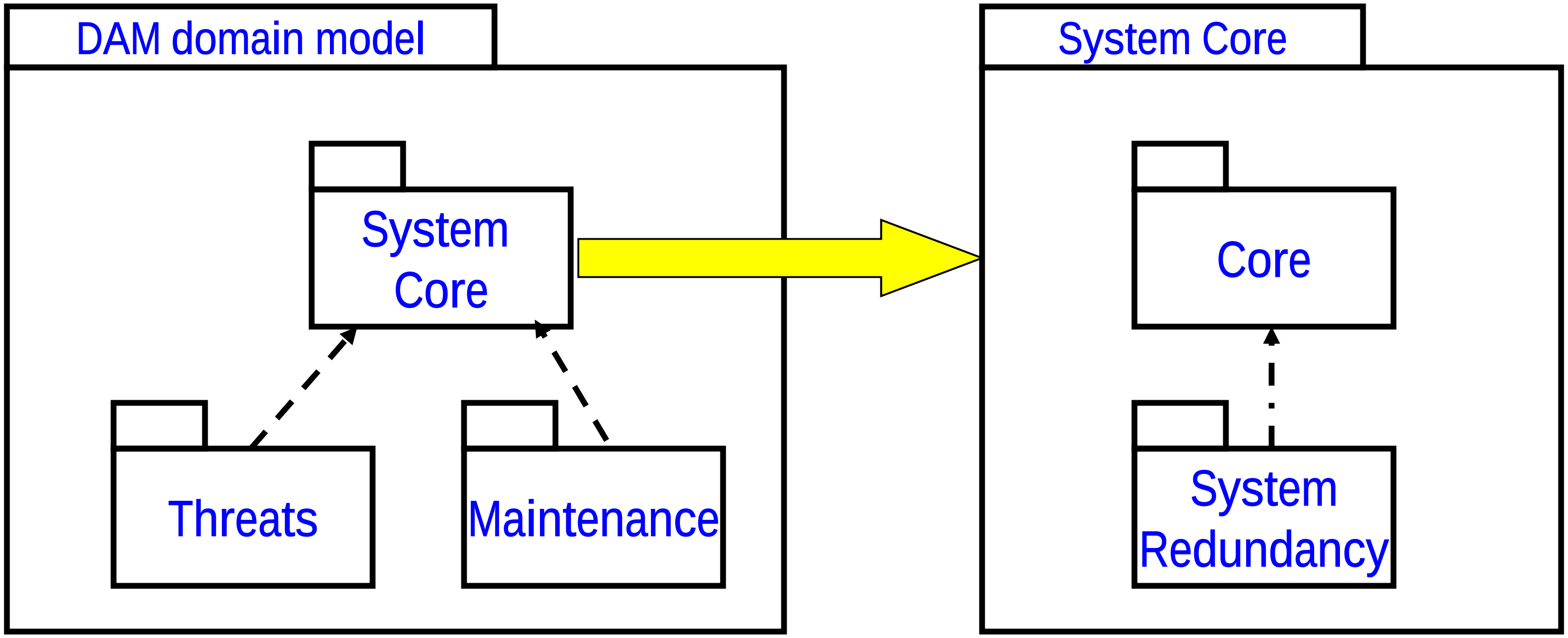
- Software performance/dependability evaluation in the context of Model-Driven Engineering:
  - starting point: UML software model used also for code generation
  - add performance annotations (using specialized profiles such as MARTE)
  - generate a performance/dependability analysis model
    - queueing 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

Definition of a UML profile for Dependability

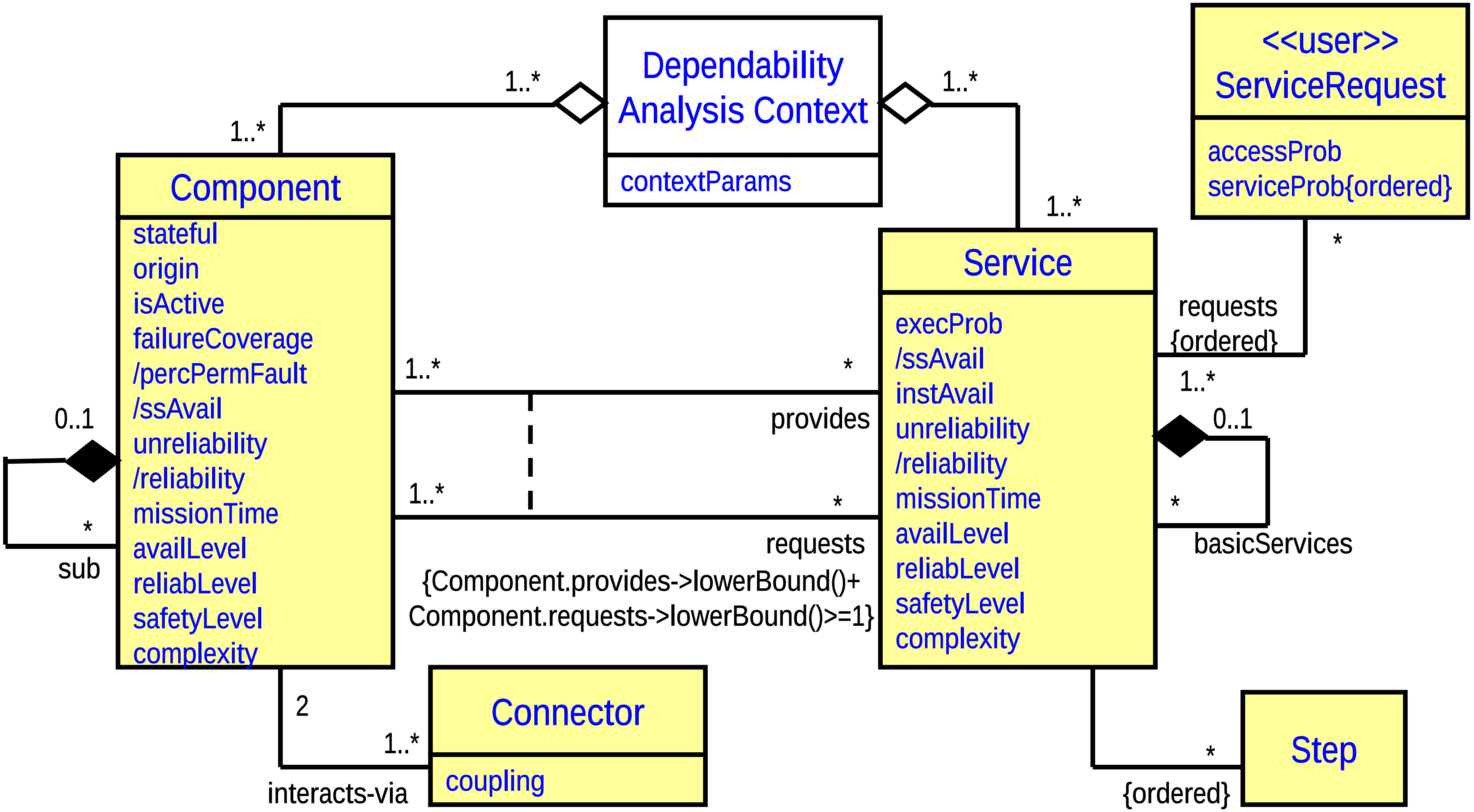
- Dependability: the ability to avoid failures more frequent or more severe than acceptable. Dependability attributes:
  a) availability: the readiness for correct service;
  b) reliability: the continuity of correct service;
  c) safety: the absence of catastrophic consequences on the users and environment;
  d) maintainability: the ability to undergo modifications and repairs.
- Dependability analysis techniques:
  - Failure Mode and Effect Analysis (qualitative evaluation)
  - stochastic Petri nets (quantitative evaluation)
  - fault trees (qualitative and quantitative)
- Research Goals
  - add dependability annotations to UML software models -> define dependability profile as an extension of the MARTE standard
  - automate the generation of dependability models from UML software models annotated with dependability information

Dependability Analysis Model

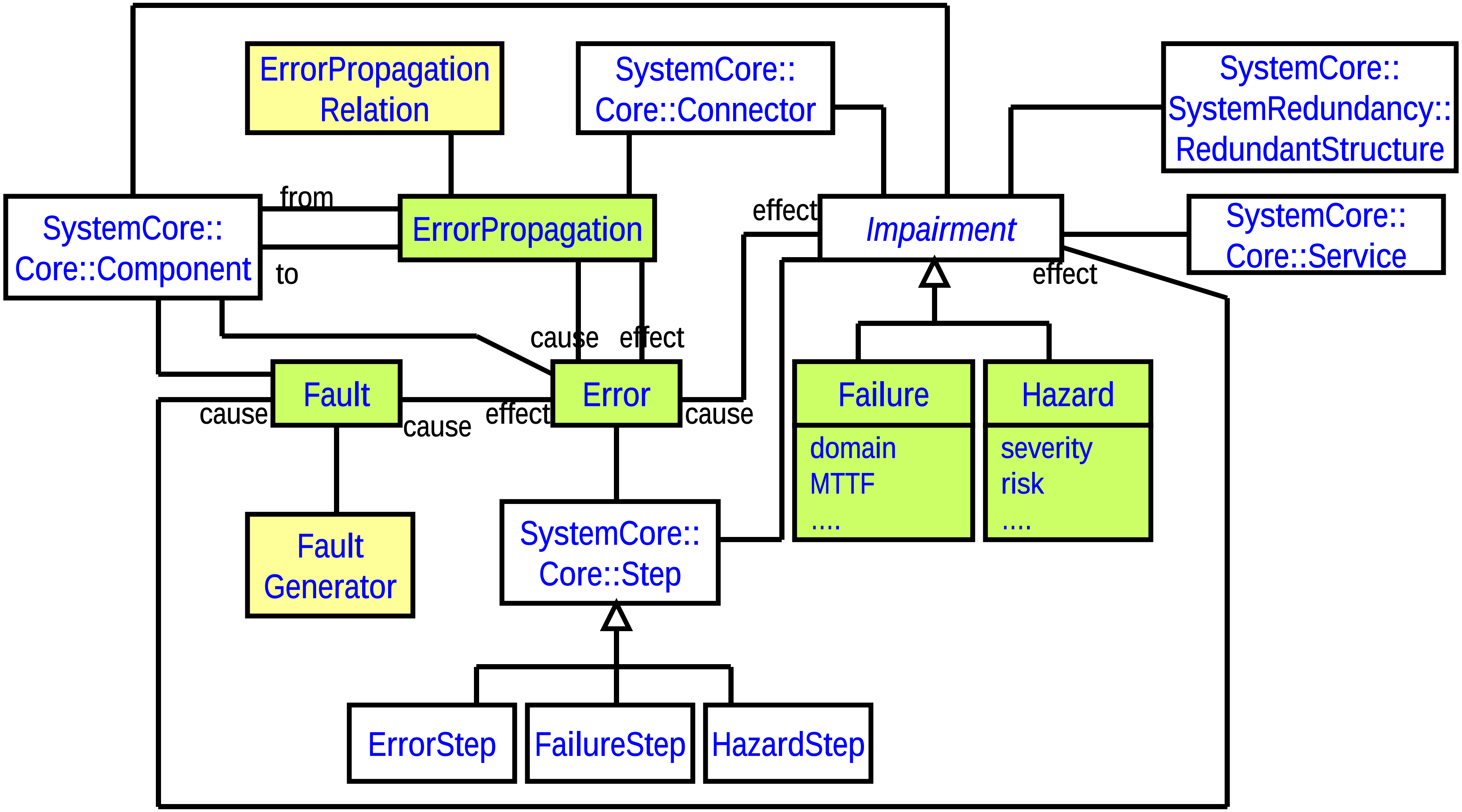
- **Domain model: represents the main concepts as classes grouped into packages**



Core Model


Dependability Analysis Context
contextParams
<<user>> ServiceRequest
accessProb
serviceProb{ordered}
Component
stateful
origin
isActive
failureCoverage
/percPermFault
/ssAvail
unreliability
/reliability
missionTime
availLevel
reliabLevel
safetyLevel
complexity
Service
execProb
/ssAvail
instAvail
unreliability
/reliability
missionTime
availLevel
reliabLevel
safetyLevel
complexity
Connector
coupling
Step
1..*
1..*
1..*
1..*
*
requests {ordered}
1..*
0..1
*
basicServices
1..*
*
provides
1..*
*
requests
{Component.provides->lowerBound()+ Component.requests->lowerBound()>=1}
0..1
*
sub
2
1..*
interacts-via
*
{ordered}

Threat Model



ErrorPropagation Relation

SystemCore:: Core::Connector

SystemCore:: SystemRedundancy:: RedundantStructure

SystemCore:: Core::Component

from

to

ErrorPropagation

effect

Impairment

SystemCore:: Core::Service

effect

cause

effect

Fault

cause

cause

effect

Error

cause

Failure

domain
MTTF
....

Hazard

severity
risk
....

Fault Generator

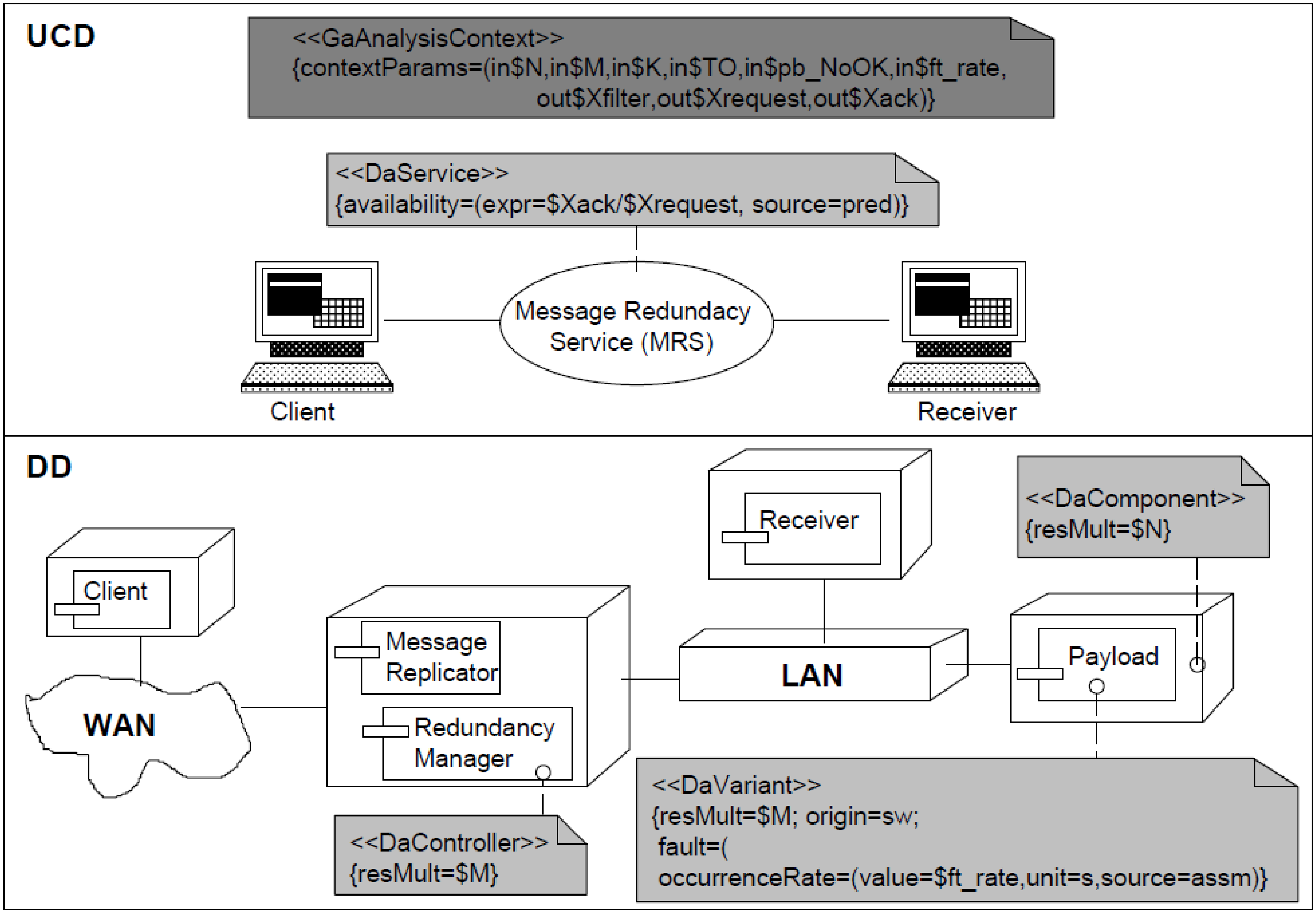
SystemCore:: Core::Step

ErrorStep

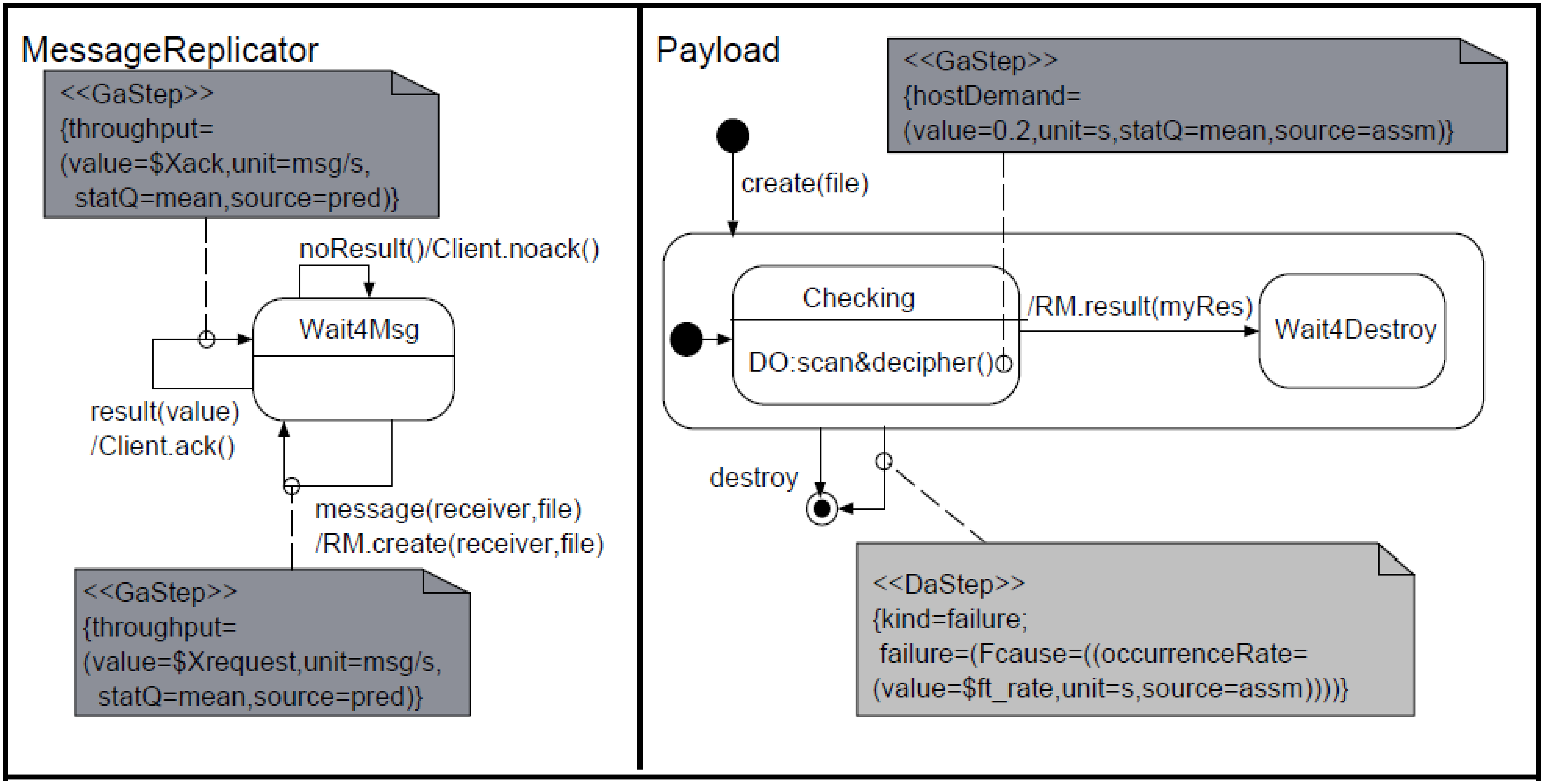
FailureStep

HazardStep

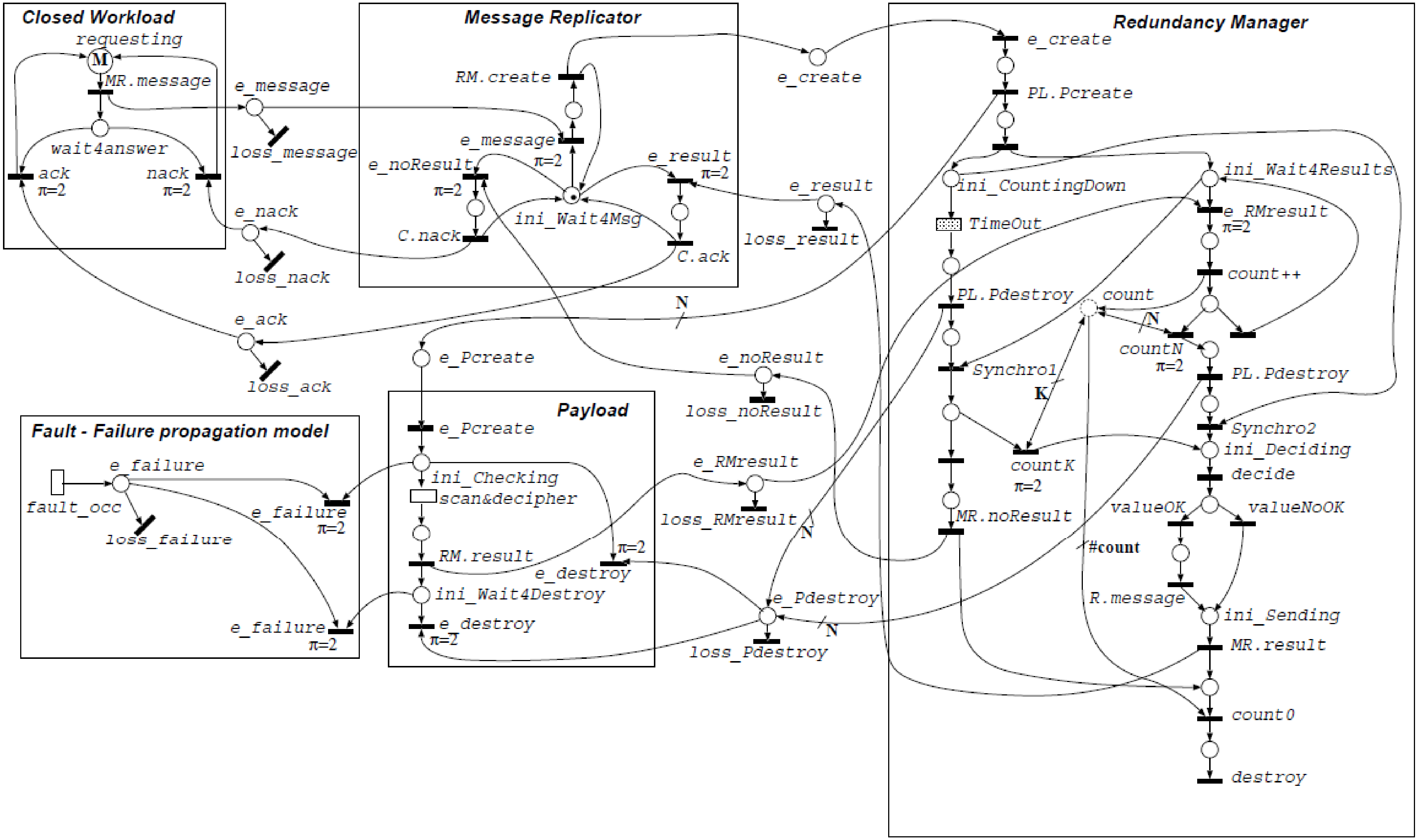
Message Redundancy Service: annotated UML model



Annotated state machines



Generated Stochastic Petri Nets Model



Context-aware SOA

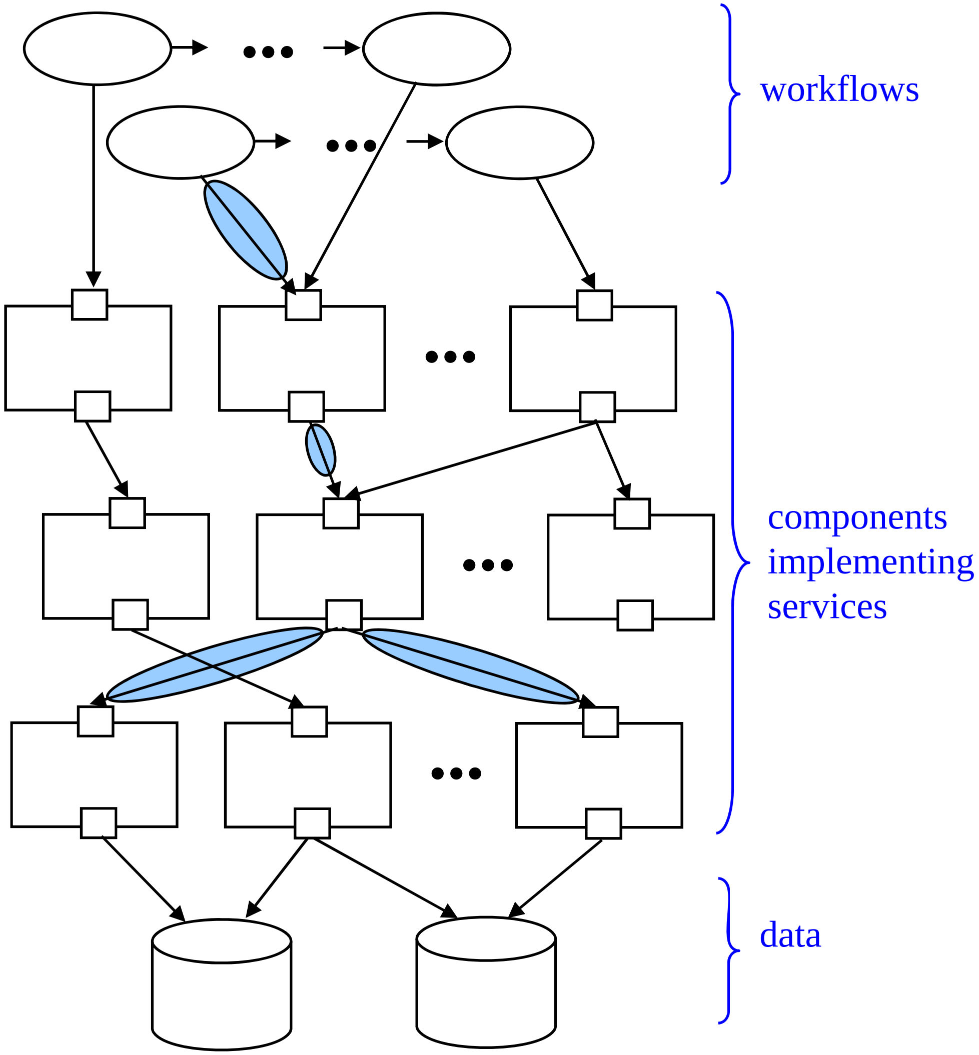
- SOA (Service-Oriented Architecture):
  - a software development paradigm aiming to develop and deploy software applications as a set of reusable composable services.
- Context-aware SOA:
  - context-aware services make use of different level of contexts and adapt the way they behave according to the current context
  - context-aware services are composed at runtime with the purpose of executing context-aware applications described by business workflows
  - 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
- Convergence of four trends:
  - Service orientation
  - Context awareness
  - Software product lines (managing variability)
  - Model-driven development

Choosing a modeling language for SOA

- Requirements for the SOA modeling language:
  - able to model different aspects of SOA systems such as:
    - workflows representing the top-level of a SOA system
    - underlying system architecture: components, services and their relationships
    - service contracts
    - detailed models of internal structure and behaviour of components and services
  - models should be complete to serve as basis for code generation
  - language should be extensible to allow adding extra information for the analysis of non-functional properties, such as performance and dependability
  - language should be preferably standard, widely used and supported by existing tools.
- Chosen language - UML extended with profiles:
  - BPMN profile (for business process models)
  - SoaML (models service relationships and contracts)
  - MARTE (modeling and analysis of real-time and embedded systems)

Performance Analysis of SOA

- PUMA4SOA Poster presented by Mohammad Alhaj
- Model transformation from a UML+MARTE model to a 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)



workflows

components implementing services

data

Publications

- **hSITE publications**

[1] S. Bernardi, J. Merseguer, D.C. Petriu, “A Dependability Profile within MARTE”, *Software and System Modeling* (SoSyM) journal, DOI:10.1007/s10270-009-0128-1, accepted 2009.

[2] D.C. Petriu, “Software model-based performance analysis”, book chapter in *Model-Driven Development for Distributed and Real-Time Embedded Systems*, (eds. J.P. Babau, J. Champeau, S. Gerard), Hermes, in press, accepted 2009.

[3] M.Alhaj, D.C.Petriu, “Approach for generating performance models from UML models of SOA systems”, submitted to Cascon 2010.

- **Other publications**

[3] C.M. Woodside, D.C. Petriu, D.B. Petriu, J. Xu, T. Israr, G. Georg, R. France, J.M. Bieman, S.H. Houmb, J.Jürjens, "Performance Analysis of Security Aspects by Weaving Scenarios Extracted from UML Models", *Journal of Systems and Software* Special Issue WOSP'2007, Vol.82, pp.56–74, 2009.

[4] S.H. Houmb, G. Georg, D.C. Petriu, B. Bordbar, I. Ray, K. Anastasakis, and R.B. France, "Balancing Security and Performance Properties During System Architectural Design", book chapter in *Software Engineering for Secure Systems: Industrial and Research Perspectives*, H.Mouratidis (Ed). IGI Global, in press, accepted 2009.

Honours and other news

- **Honours:**
  - **elected as a Fellow of the Canadian Academy of Engineering – 2010**
- **Program Committee Chair:**
  - **The 13th ACM International Conference on Model Driven Engineering Languages and Systems (MoDELS 2010)**
    - **the premier international conference on model-driven software development**
    - **high-quality conference, typical acceptance rate under 20%.**
- **Keynote Speaker:**
  - **QUASOSS' 2010: Quality of Service-Oriented Software Systems**
- **Program Committees:**
  - **10 conferences in 2010**
- **Contributor to international standards:**
  - **UML Profile for Modeling and Analysis of Real-Time and Embedded Systems (MARTE) – part related to performance**
    - **RFP issued by OMG in 2005**
    - **Version 1.0 adopted as a OMG standard in December 2009.**