PSML-S: A Process Specification And Modeling Language For Service-Oriented Computing

W.T. Tsai, Ray A. Paul*, Bingnan Xiao, Zhibin Cao, Yinong Chen
Department of Computer Science and Engineering, Arizona State University
PO Box 8809, Tempe, AZ 85287-8809, U.S.A.
*Department of Defense, U.S.A.
wtsai@asu.edu

Agenda

- Background
- Considerations
- PSML-S Introduction
- Case study
- Summary
- Questions

SOC Background

- Service-oriented computing including service-oriented architecture (SOA) and Web Services (WS) are receiving significant attention recently.
- Most major corporations and government agencies (including DoD and NASA) are pushing this technology.
- The idea that a software program as a service that can be discovered, matched, composed, executed, verified, and monitored in real time and at runtime provide a new paradigm of computing.

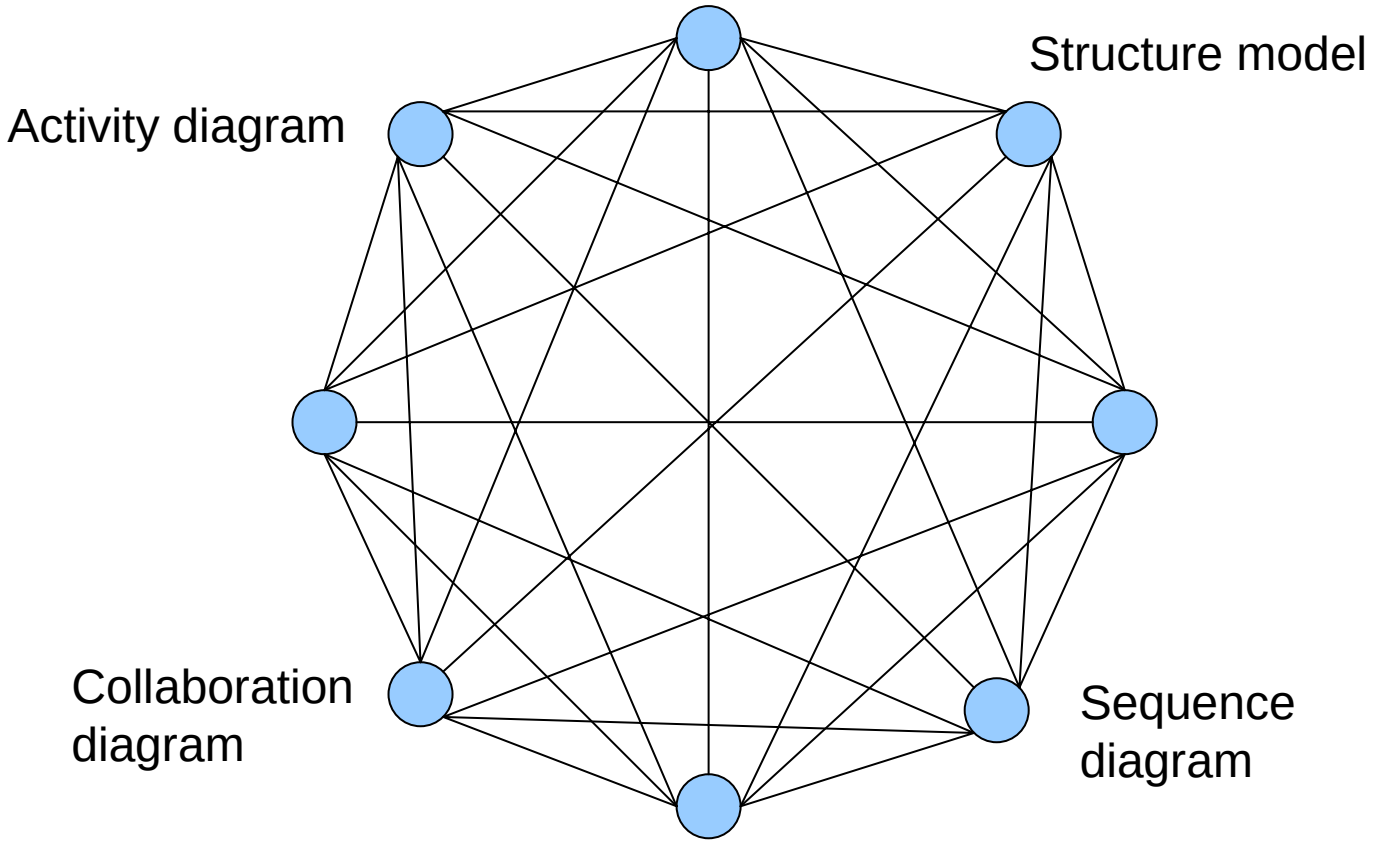
Impacts of SOC

- SOC has the following impacts on the software engineering:
  - Knowledge of existing services is as important as ability to build a new one;
  - System interaction via loosely coupled service interaction;
  - The critical role of SOC architecture;
  - Quality assurance issues;

Challenges of UML

- Although UML is an excellent methodology for OO programming, it has weak support for dynamic architecture with run-time composition, re-composition, and reconfiguration.
  - The different diagrams in UML are essentially inter-connected models for a software system.
    - They are inter-connected but not fully integrated in the sense that some transformations between models are difficult to automate.
    - Each individual model requires lots of effort to create and maintain.
      - If one model is changed, the other models must be updated, and often manually.
      - The efforts to maintain the consistency among the models are huge because these models are not integrated but interconnected.

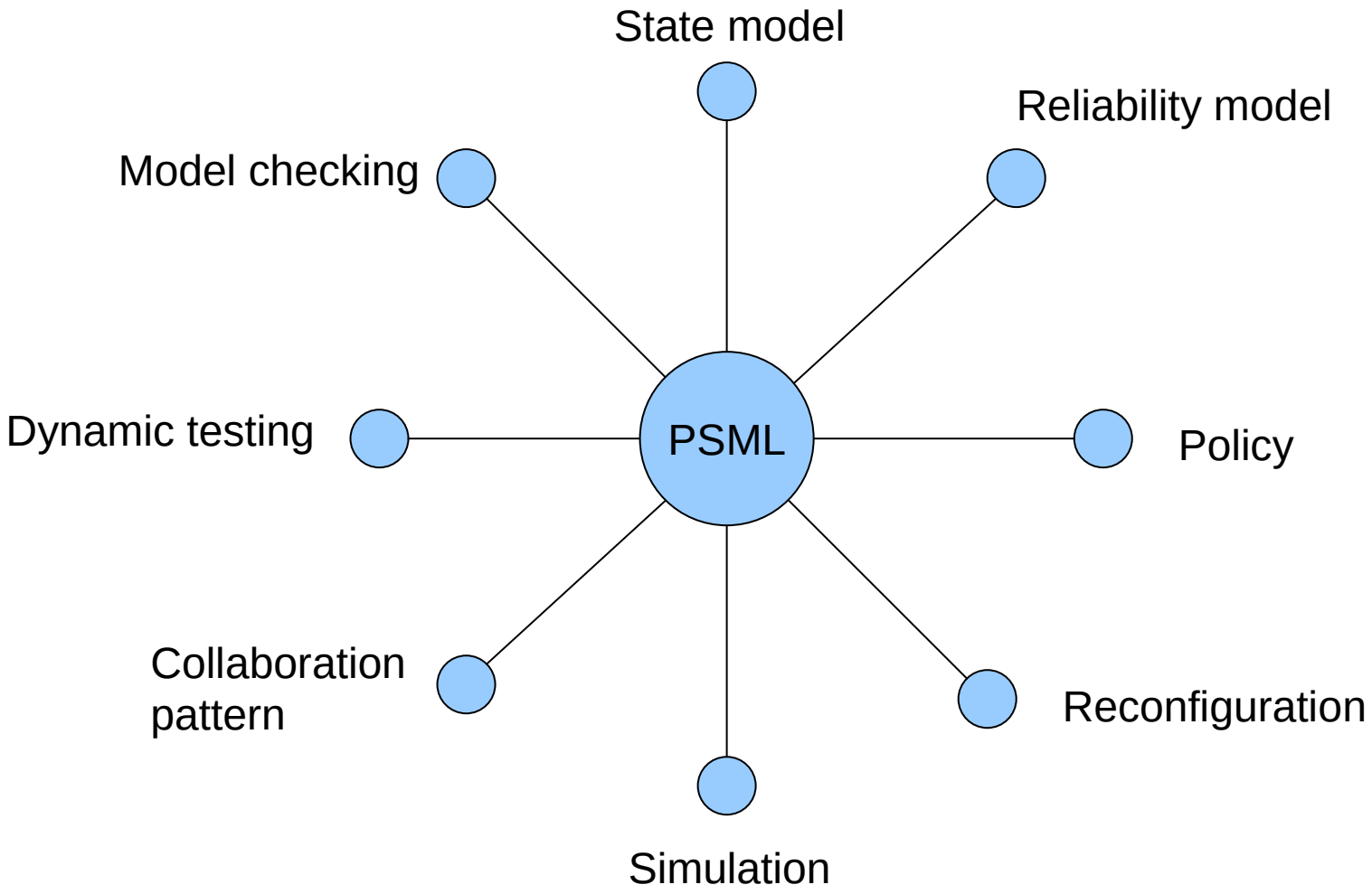
Multi-Model Multi-Analyses



Why Is PSML Important?

- Instead of multiple inter-connected models for different analyses, PSML provides a single-model which can support multiple analyses.
- The integrated PSML model and the automation can
  - Generate other models:
    - State model
    - Reliability model
    - Policy model (constraint model)
  - Perform multiple analyses:
    - Model checking
    - Simulation
    - Completeness & consistency analysis
    - Event analysis
  - Support dynamic properties:
    - Dynamic testing
    - Dynamic reconfiguration
    - Dynamic collaboration with patterns

Single-Model Multi-Analyses



PSML-S Design Considerations

- PSML-S
  - It is a language for specification and modeling of services in system engineering, software engineering, and service engineering.
  - It facilitates the specification, analysis, composition, recomposition, simulation, execution, and testing of the system based on the system specification.

PSML-S Motivation

- Good models are essential for communication among project teams and to assure architectural soundness.
- With the system model specified, we can evaluate the system architecture and simulate system behaviors in the early phase of software engineering life cycle.
- The advance in compilation and code generation technologies makes it possible to directly translate the specification into executable, which greatly simplify the programming tasks.

PSML-S for Rapid Development and Runtime Change

- PSML-S is designed to support rapid development of services in the SOC.
  - Once the PSML-S model is constructed, engineers can perform:
    - Static analysis
    - Dynamic analysis
    - System verification
    - System evaluation
  - Automated code generation
  - Automated change application
    - When changes are needed, the designers need only to change the specification and the changes may be applied to the system with the automation

Goals of PSML-S

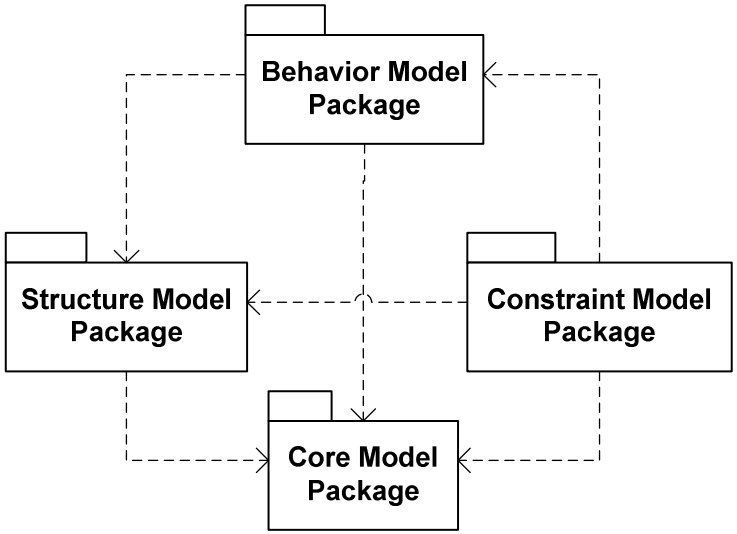
- The goals of PSML-S is to specify a model which is:
  - Service-oriented for dynamic composition and re-composition
  - Self-contained and coherent
  - Hierarchical
  - Dynamic verification, validation, simulation, runtime monitoring, and evaluation
  - Data-centric
  - Event-driven and time-triggered
  - Policy-driven

PSML-S Metamodel and models

- The PSML-S metamodel is defined following the four-layer modeling architecture from UML
- The four layers of models defined in PSML-S are:
  - At meta-metamodel layer: the MOF (Meta Object Facility) meta-metamodel
  - At metamodel layer: the PSML-S metamodel
  - At model layer: PSML-S model
  - At user object layer: a real system

PSML-S Model Packages

- Core model package
- Structure model package
- Behavior model package
- Constraint model package

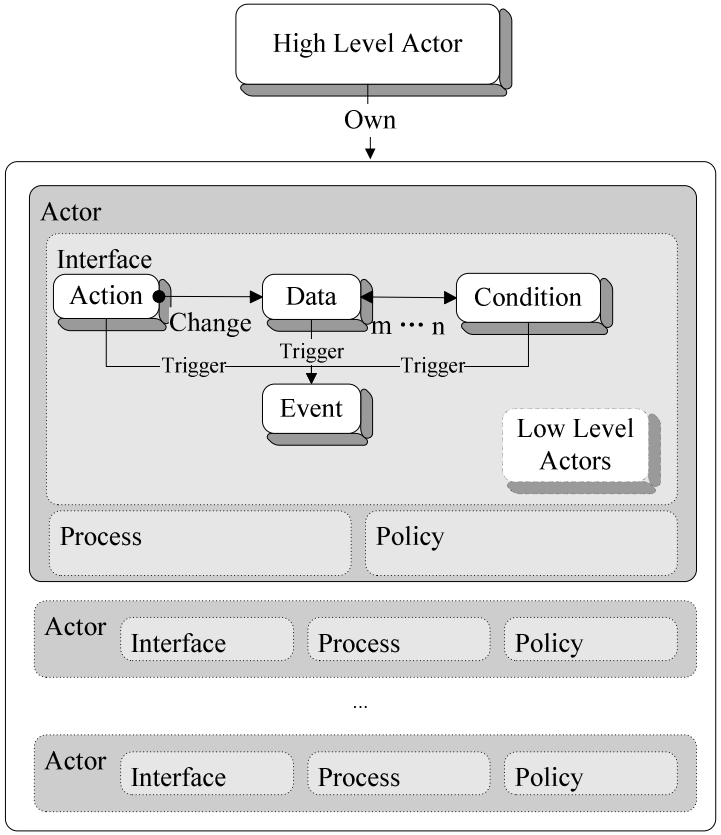


Core Model Package

- Defines data types and operators.
  - Data types
    - Defines a set of data types which are used to categorize PSML-S model elements
  - Operators
    - Defines the relations and operations among the model elements

Structure Model Package

- Defines model elements for specifying the static structure of a PSML-S model.
  - Actor
  - Action
  - Condition
  - Data
  - Event
  - Attributes
- The figure shows the hierarchical structure of structure model elements.

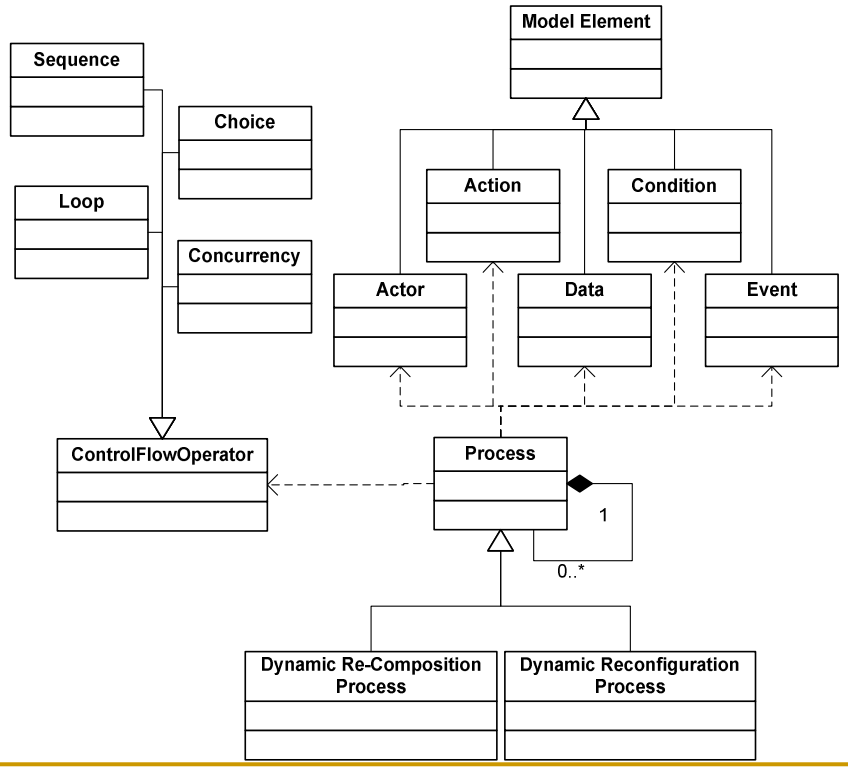


Relations of Structure Model Elements

Element	Actor	Action	Condition	Data	Event
Actor	Own Compose Composed of	Own Perform	Own	Own	Own Initiate Receive
Action	Belong to Performed by	Sequential Concurrent	Change	Change Carry	Trigger Handle
Condition	Belong to Describe	Pre-status Post-status	Compose Composed of	Specify	Trigger
Data	Belong to	-	Compose	Compose Composed of	Trigger
Event	Belong to Initiated by Received by	Triggered by Handled by	Change	Carry	Compose Composed of
Attributes	Associate to	Associate to	Associate to	Associate to	Associate to

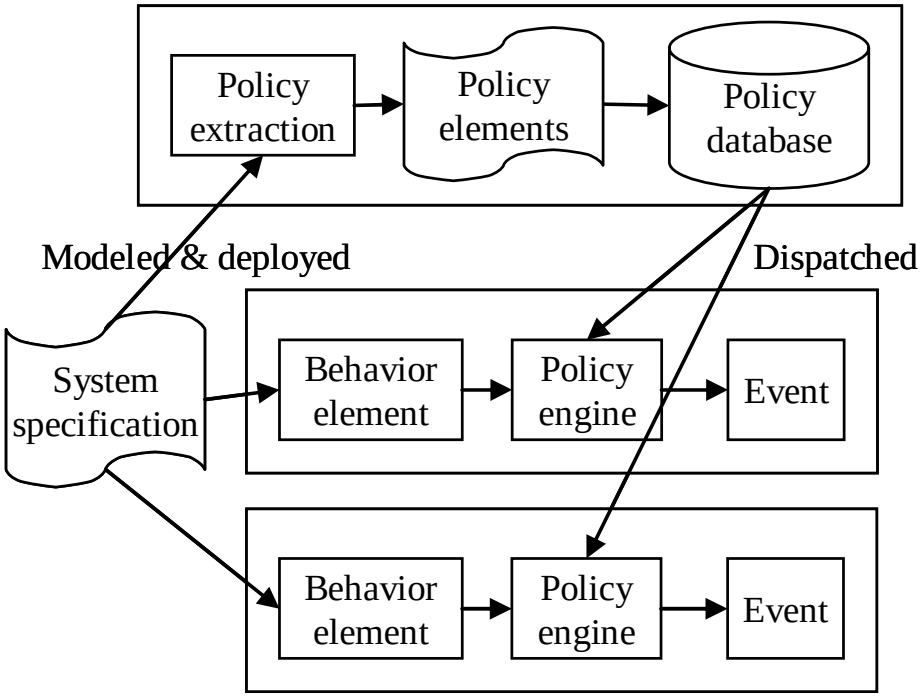
Behavior Package

- Defines model elements for specifying the dynamic behavior of a PSML-S model



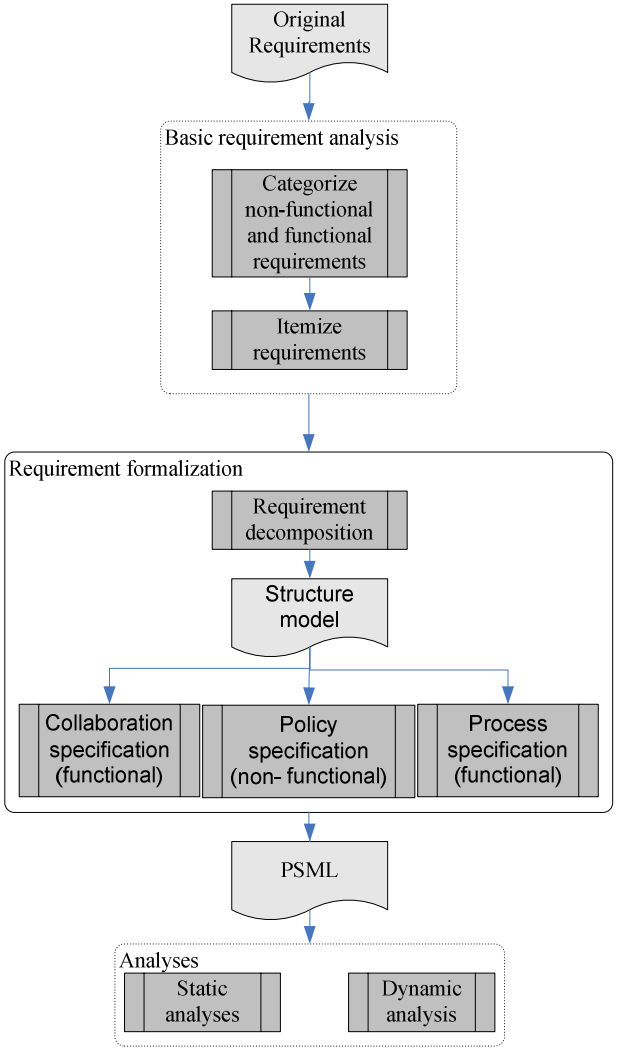
Constraint Model Package

- Defines model elements for specifying the constraints in both structure and behavior packages



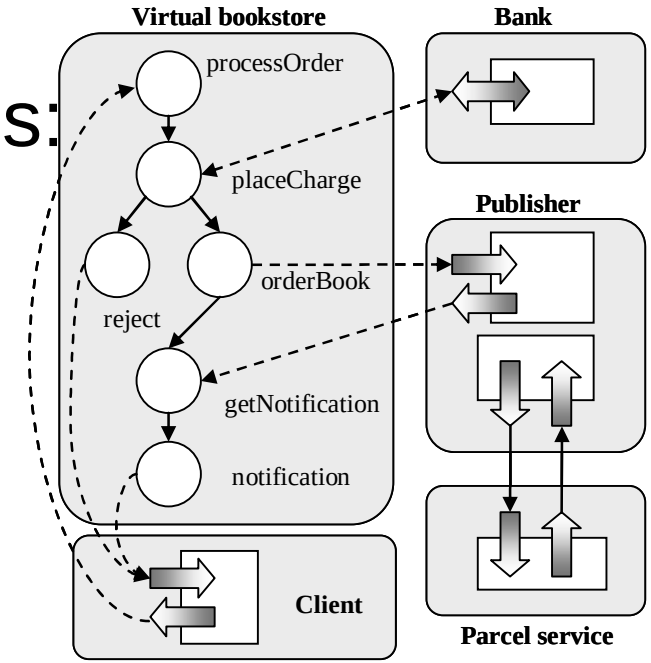
PSML-S Modeling Processes

- The figure shows the over all process of the modeling
  - It is a spiral process.
  - Once the specification is ready, multiple analyses can be automated.
  - Once the model is ready, the system can be generated with the automation.



Case Study

- We use PSML-S to specify the structure model and the behavior model of a virtual bookstore.
- The structure model includes:
  - 5 Actors
  - 1 Condition
  - 9 Data
  - 8 Actions
  - 4 Events



Case Study (Cont.)

- Based on the structure model, we specified the following processes:
  - *BeginToProcessOrder*: Triggered by event *CustomerOrderReceived*.
  - *BusinessProcessRejection*: Triggered by event *ChargeFail*.
  - *BusinessProcessNormal*: Triggered by event *SuccessfullyCharging*.
  - *BusinessProcessReceipt*: Triggered by event *SendingConfirmation*.
- After we specified the process specification, we performed following analyses to evaluate and refine the model:
  - Completeness & Consistency Checking
  - Simulation
  - Model Checking
  - Event Analysis

Summary

- As a part of an integrated Web service development environment, PSML-S provides the modeling and specification capacity.
- The model specified in PSML-S can be verified by completeness & consistency analysis and model checking tool, and then, automatically translated into executable for simulation and testing.
- This paper presented the metamodel of PSML-S and the model elements in different model packages.

Questions