

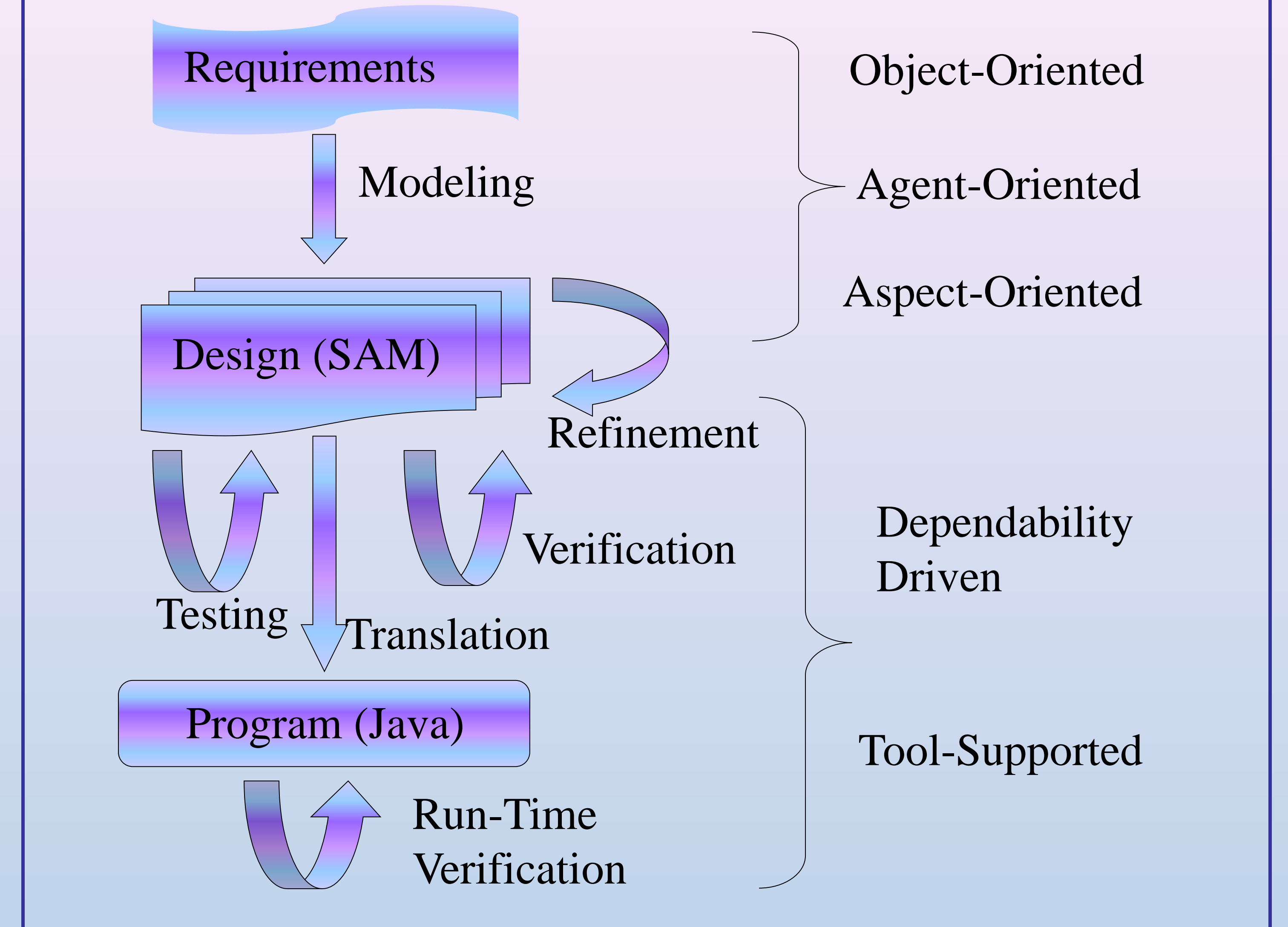
SAM Framework for Software System Modeling, Analysis and Realization

Dr. Xudong He Director, Center for Advanced Distributed System Engineering

Introduction

In the past decade, software architecture has emerged as a major research area in software engineering. Many architecture description languages and analysis techniques have been proposed. Software Architecture Model (SAM), a graphical software architecture description model has been developed at FIU since 1998. SAM is a general software architecture development framework based on two complementary formalisms - Petri nets and temporal logic. Petri nets are used to visualize the structure and model the behavior of s/w architectures while temporal logic is used to specify the required properties of s/w architectures. These two formal methods are nicely integrated through the SAM software architecture framework. Furthermore, SAM provides the flexibility to choose different compatible Petri net and temporal logic models according to the nature of system under study. Most importantly, SAM supports formal analysis of software architecture properties in a variety of well-established techniques – simulation, model checking, and testing.

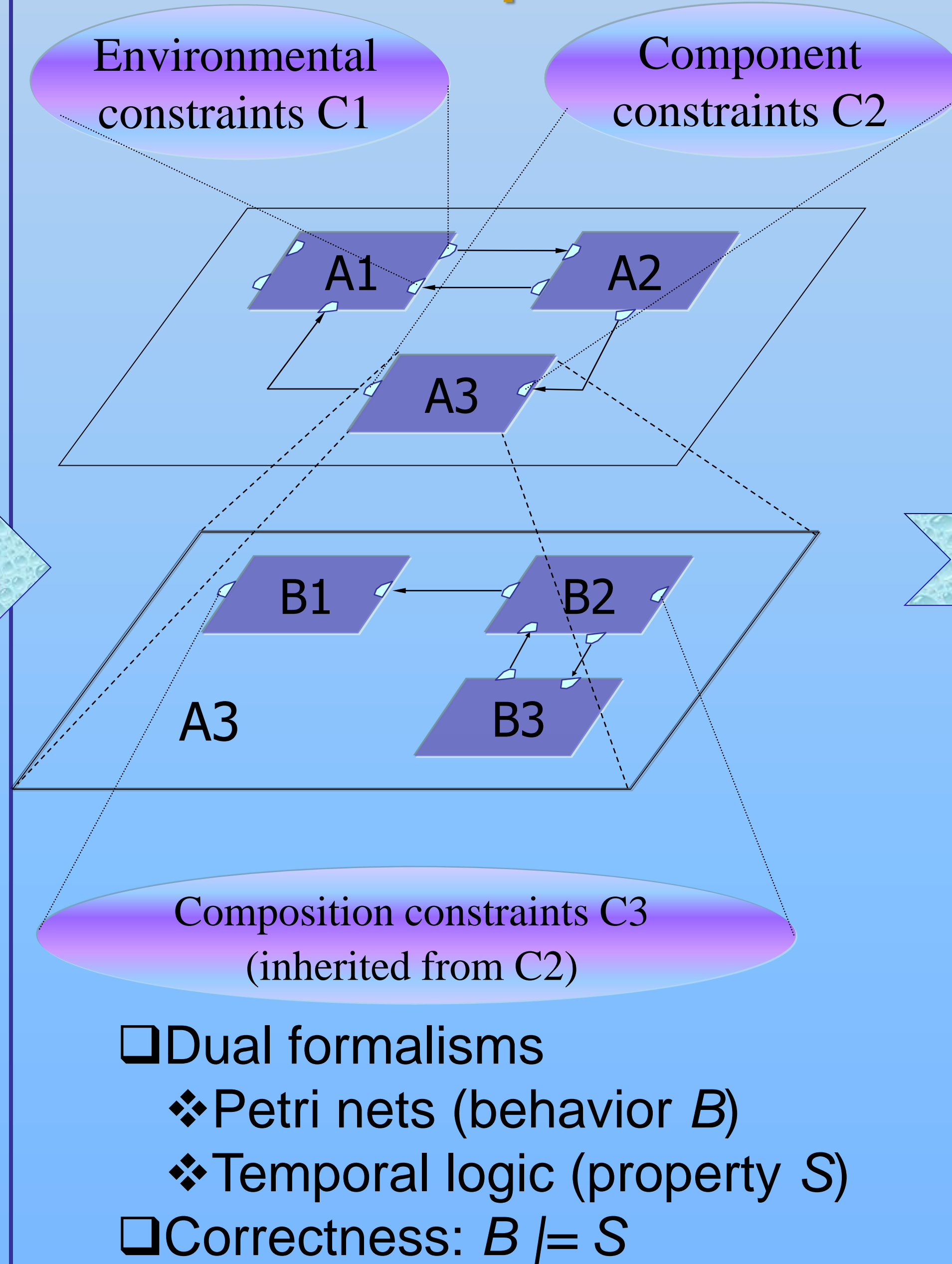
SAM Framework



Modeling

- ❑ Object Oriented Modeling
 - ❖ Classes / objects
 - attributes – places
 - operations – transitions with proper constraints
 - classes – net structures
 - objects – tokens
 - ❖ Class relationships
 - net structures
- ❑ Agent Oriented Modeling
 - ❖ Agent Interaction - upper level net
 - ❖ Individual agent behavior - lower level net
- ❑ Aspect Oriented Modeling
 - ❖ Base Model – Petri net
 - ❖ Aspects – Petri nets
 - Pointcut – places/transitions
 - Advice – net structure
 - ❖ Weaving – integrating base Petri net with aspect Petri nets

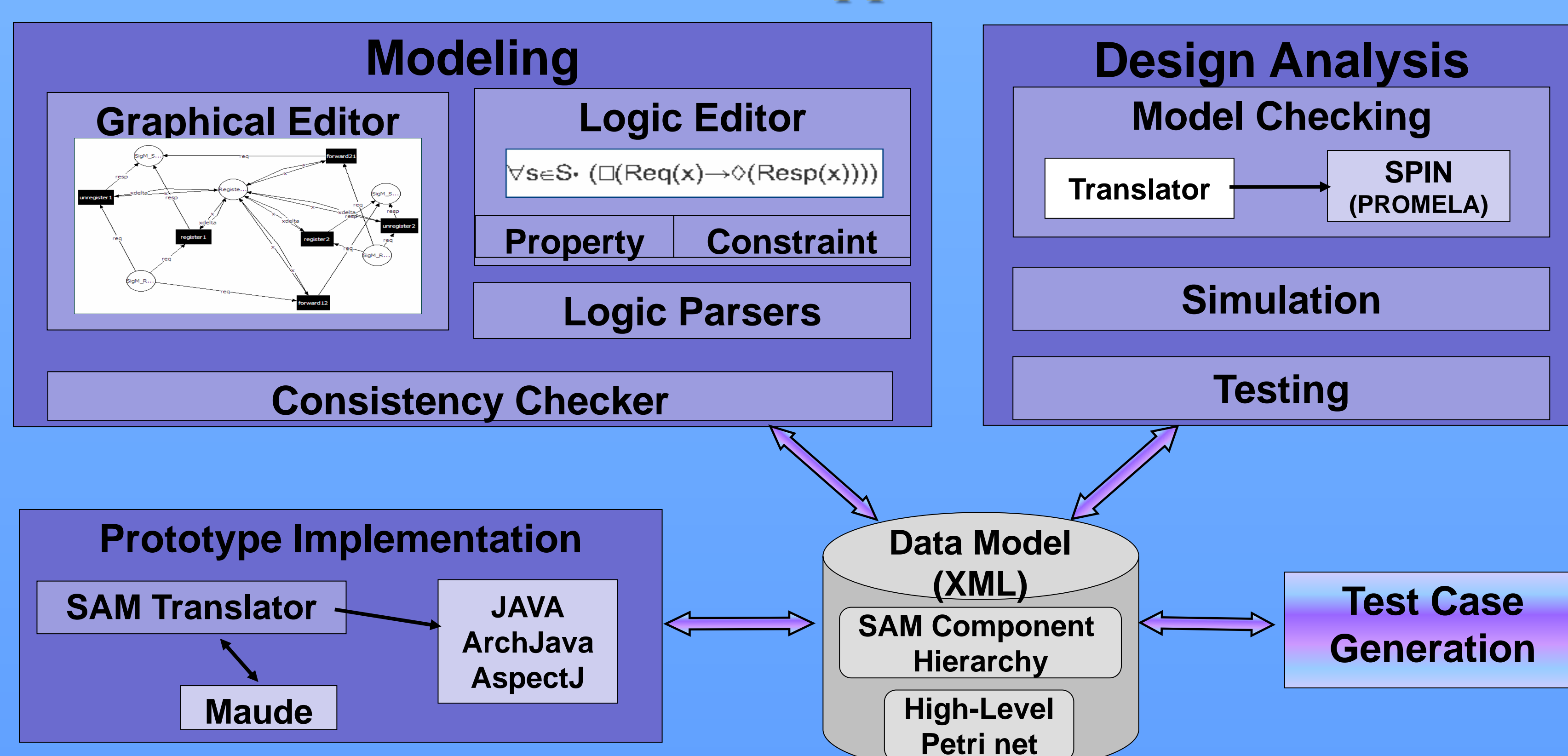
Structures of SAM Descriptions



Analysis

- ❑ Model Checking
 - ❖ Selecting a model checker
 - ❖ Translating a SAM model into the input language of a model checker
 - ❖ Ensuring the translation correctness
 - ❖ Running the model checker to verify safety and liveness properties
 - ❖ Model checkers used – SMV, SPIN, MAUDE
- ❑ Testing
 - ❖ Defining and selecting model coverage criteria
 - ❖ Instrumenting the Petri net simulator
 - ❖ Measuring the coverage
 - ❖ Evaluating the testing results

Tools Support



Realization

- ❑ Translating a SAM model into Java source code
 - ❖ Behavior – Java
 - ❖ Structure – ArchJava
 - ❖ Property – AspectJ
- ❑ Ensuring Translation Correctness
 - ❖ Completeness – The resulting Java program defines all elements of the given SAM model.
 - ❖ Consistency – Semantic Consistent