



TEXAS TECH UNIVERSITY™

Scalability in Analysis of Software Architecture

Advisor: Dr. Michael Shin

Bosah Chukwuogo
Thesis Defense

June 14th 2007



PRESENTATION AGENDA



- Introduction
- Related Work
- Problem Statement
- Research Approach
- Validation
- Future Work and Conclusion



- UML does not provide dynamic analysis capabilities for software models such as software architecture
 - *Transformation from UML to Color Petri Net (CPN) provides a way of dynamic analysis of UML models*

- UML to CPN transformation
 - *prone to state explosion problems for software architecture of large-scale systems*

- This thesis
 - *proposes a solution for analyzing software architecture for large-scale systems*

PRESENTATION AGENDA



- Introduction
- Related Work
- Problem Statement
- Research Approach
- Validation
- Future Work and Conclusion



RELATED WORK

- Several transformation approaches have been proposed
 - *Baresi's Approach*
 - *Elkoutbi's Approach*
 - *Saldhana's Approach*
 - *Pettit's Approach*
 - *Shin's Approach*

- They do not consider scalability

PRESENTATION AGENDA



- Introduction
- Related Work
- Problem Statement
- Research Approach
- Validation
- Future Work and Conclusion



- The CPN model that is transformed from a UML-based software architecture model for a large-scale system can generate the large state space, which may result in a state explosion problem in the analysis of the software architecture. It is necessary to develop a transformation approach that reduces the CPN state space while preserving the modeled behavior of the software.

PRESENTATION AGENDA



- Introduction
- Related Work
- Problem Statement
- **Research Approach**
- Validation
- Future Work and Conclusion



SOFTWARE ARCHITECTURE UML MODEL

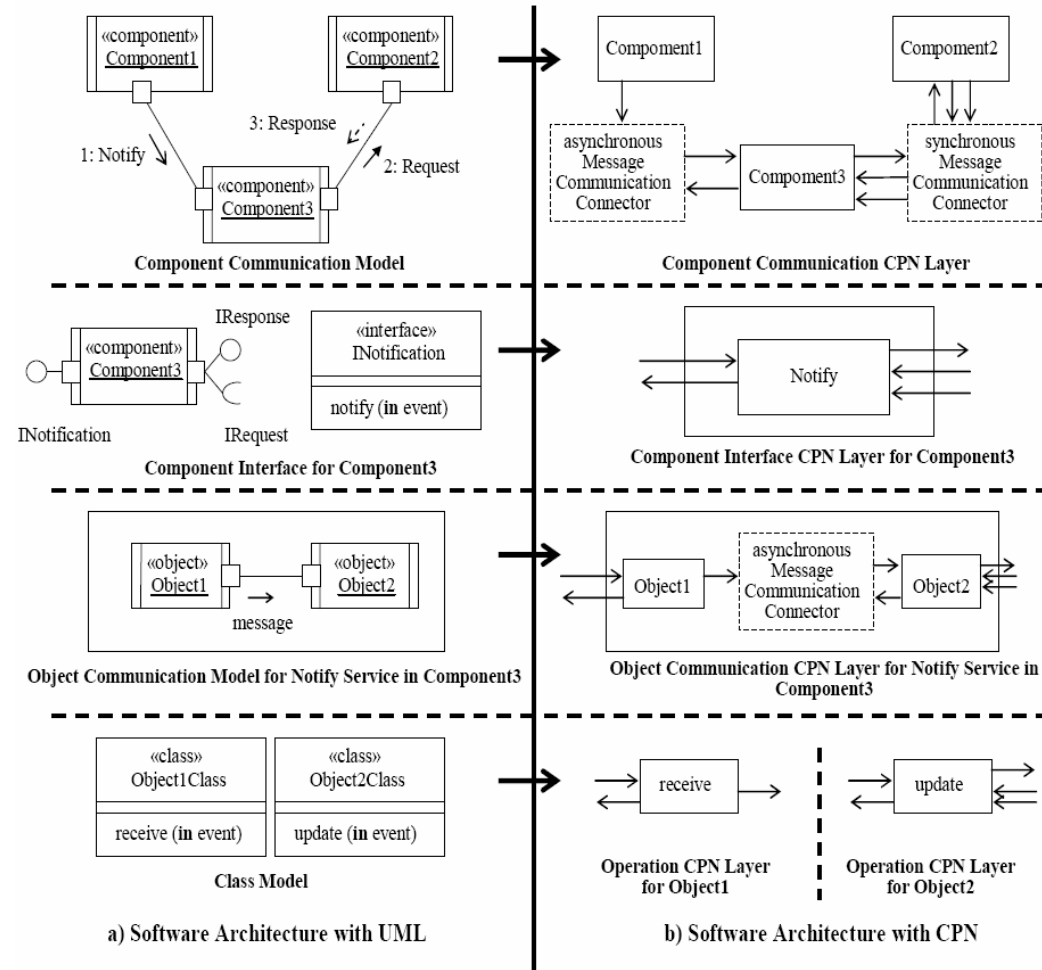
- Software architecture UML model consists of:
 - *Component communication model*
 - *Component interface structure model*
 - *Object communication model*
 - *Class model*

- Message communication styles in software architecture with UML
 - *Synchronous with reply*
 - *Synchronous without reply*
 - *Asynchronous*

OVERVIEW OF UML TO CPN TRANSFORMATION APPROACH



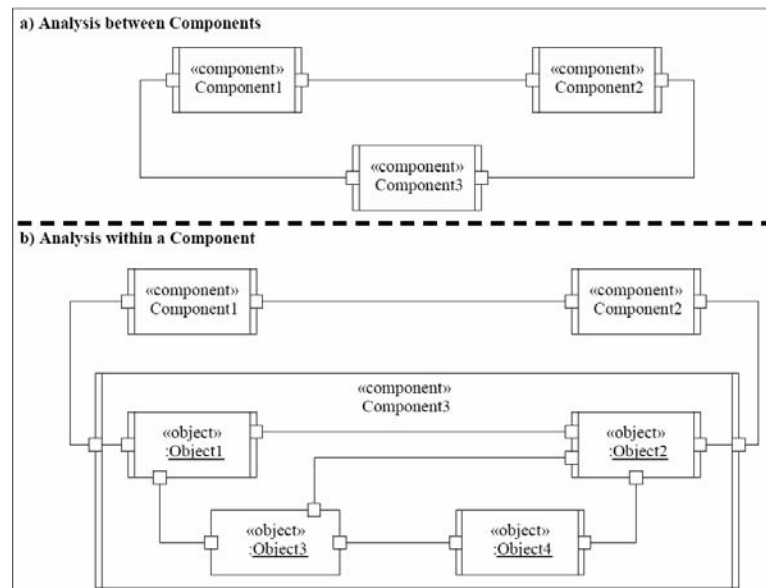
- 4 CPN hierarchy layers
 - *Component communication*
 - *Component interface*
 - *Object communication*
 - *Operational*
- Defined message communication connectors in CPN model
 - *Corresponding to message communication styles in UML model*
- CPN tokens are created for each message





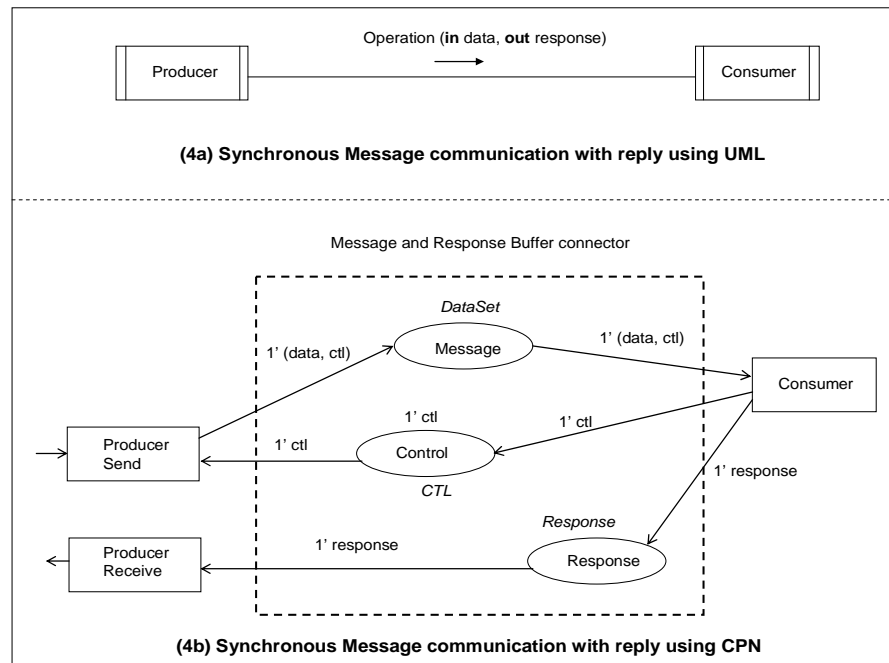
ANALYSIS APPROACH FOR LARGE-SCALED SYSTEMS

- Analysis is performed at different levels of component abstraction
- All abstract components are analyzed first, and then each concrete component is analyzed in turn
 - *Instead all concrete components are analyzed together*





- Connectors defined in CPN for message communication in UML model
 - *Synchronous message communication with reply*
 - *Synchronous message communication without reply*
 - *Asynchronous message communication*

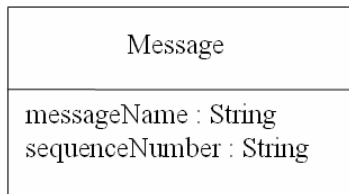


UML TO CPN TRANSFORMATION APPROACH

- CPN message token transformation



- CPN product data-type is used to model a UML message
- The UML message class provides the basic parameters for the CPN message token
- Parameterized UML messages are also transformed



a) Message Class Diagram

```
colset msgSeqID = string;  
colset msgName = string;  
colset msg = product msgName * msgSeqID;
```

b) CPN colorset for Message Class

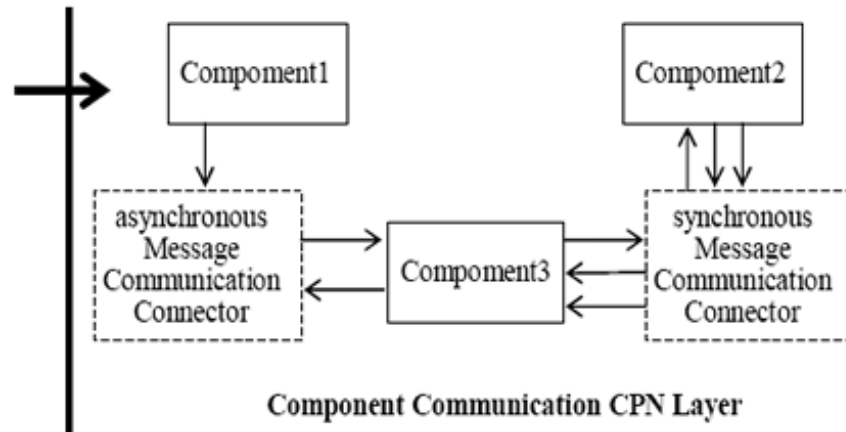
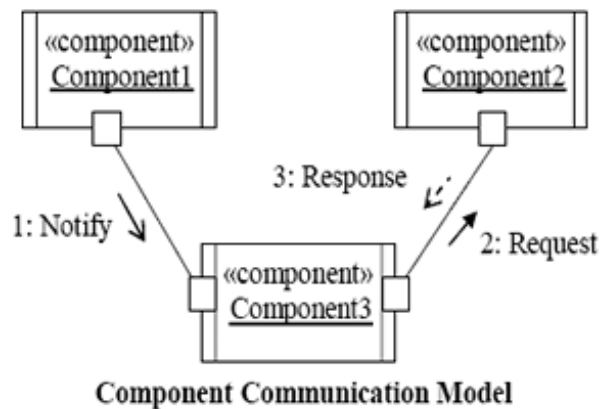
```
colset msgSeqID = string;  
colset msgName = string;  
colset msg = product msgName * msgSeqID;  
  
colset cardID = string;  
colset cardPIN = string;  
colset reqPINArgs = product cardID * cardPIN;  
  
colset reqPINValidationMsg = product msg * reqPINArgs;
```



UML TO CPN TRANSFORMATION APPROACH

- Step 1: Component communication layer transformation

- UML components are transformed to CPN transitions
- Message communication is transformed using the appropriate CPN message connector

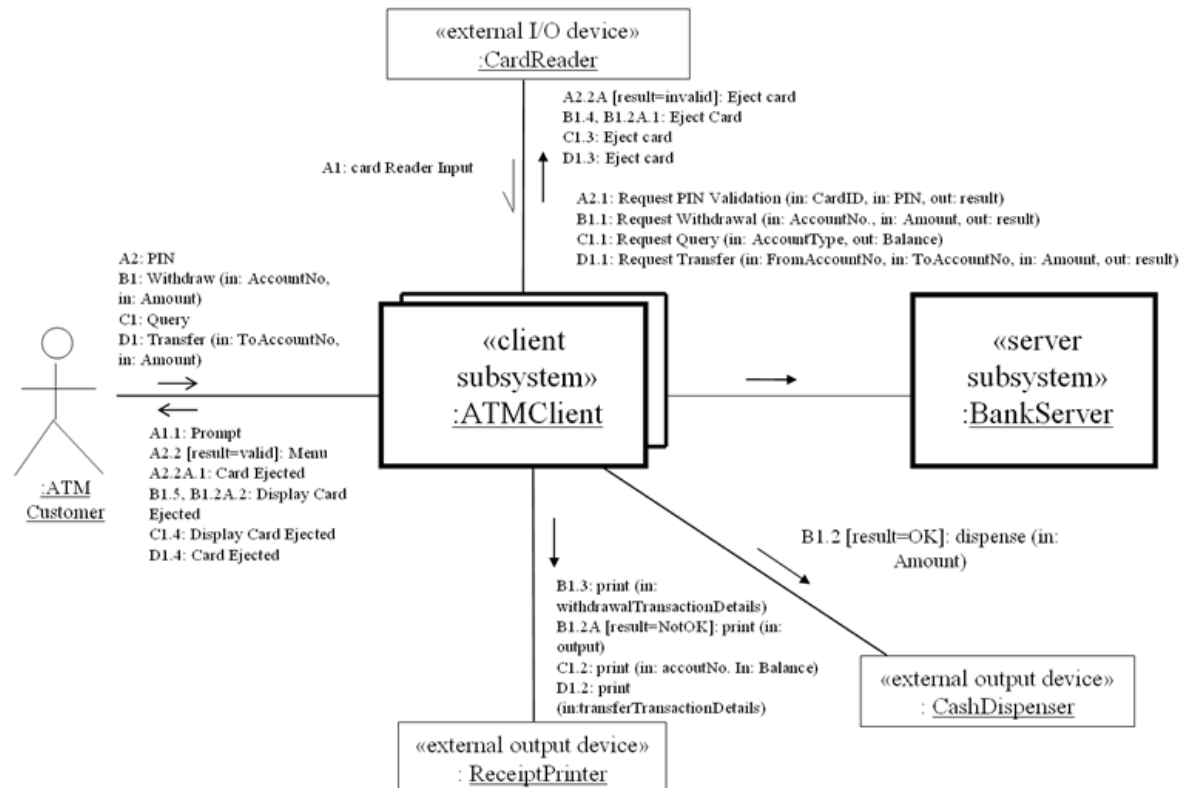


UML TO CPN TRANSFORMATION APPROACH

- Step 1: Component communication layer transformation



- Component communication UML layer for ATM system.
- 4 services are provided by the server to the client.

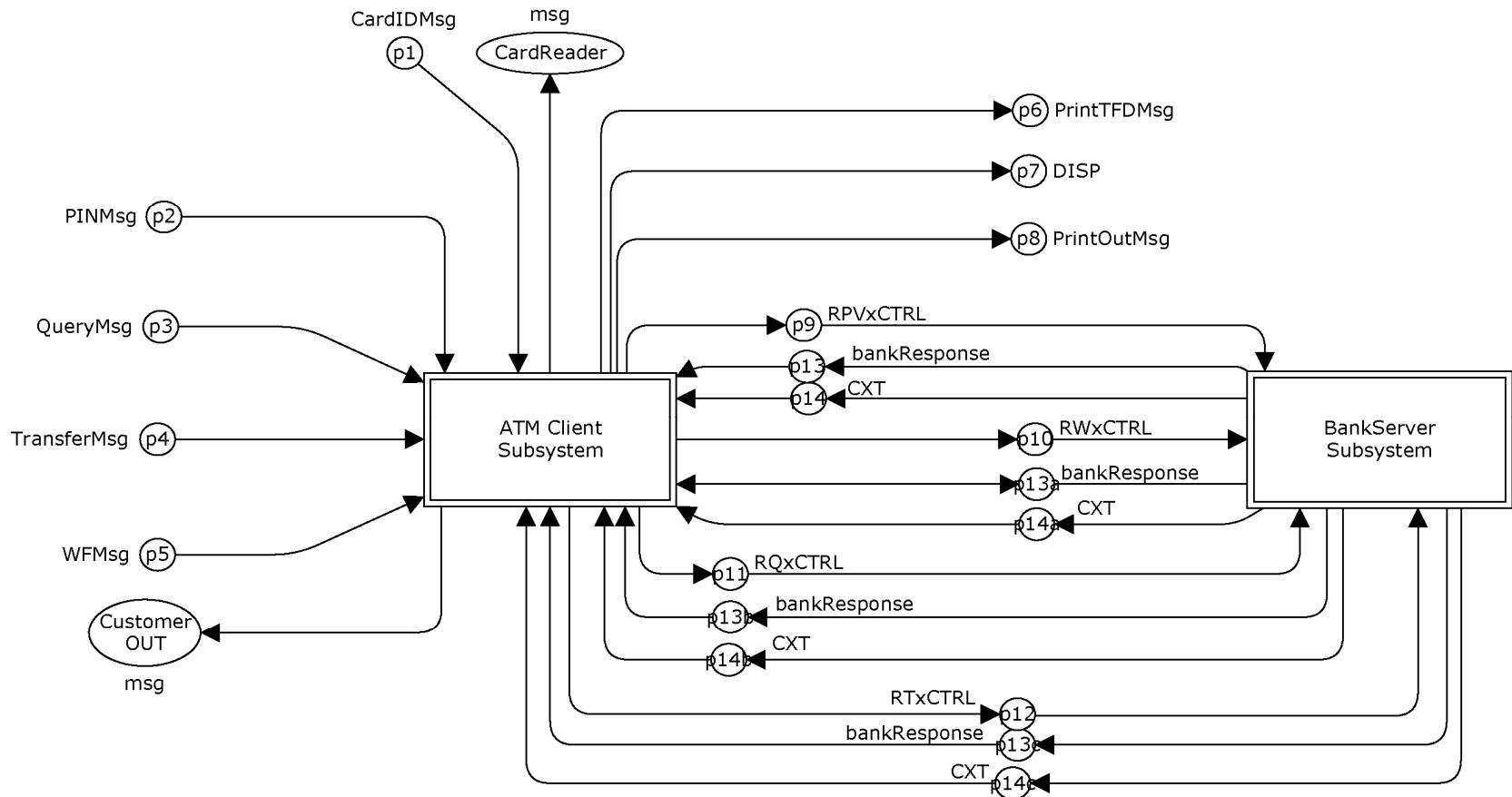


UML TO CPN TRANSFORMATION APPROACH

- Step 1: Component communication layer transformation



■ Component communication CPN layer for ATM system

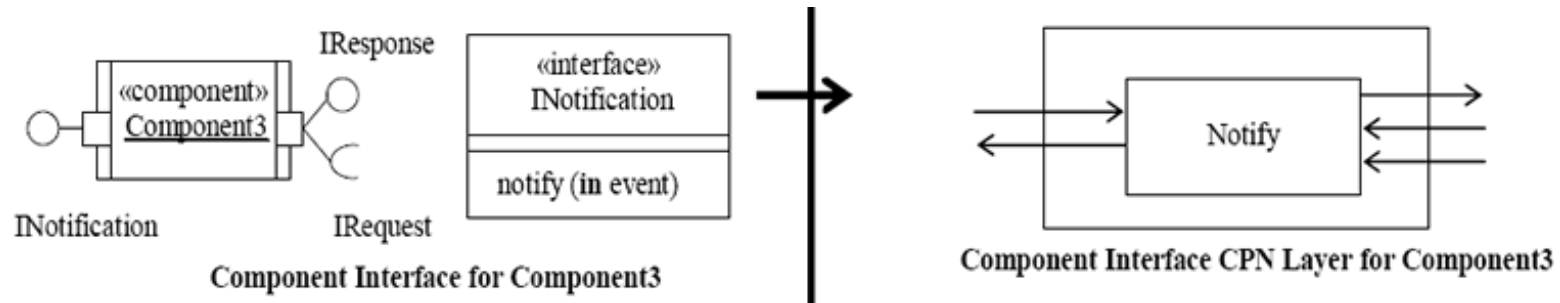


UML TO CPN TRANSFORMATION APPROACH

- Step 2: Component interface layer transformation



- Each service defined in an interface is transformed to a CPN transition

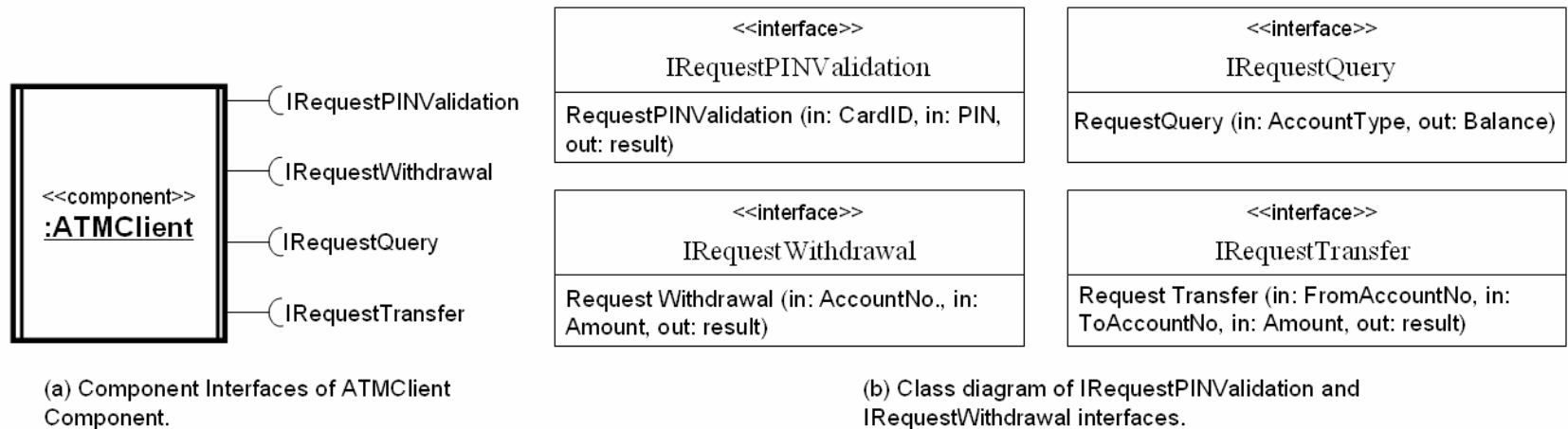


UML TO CPN TRANSFORMATION APPROACH

- Step 2: Component interface layer transformation



- Interfaces implemented by ATM Client component.

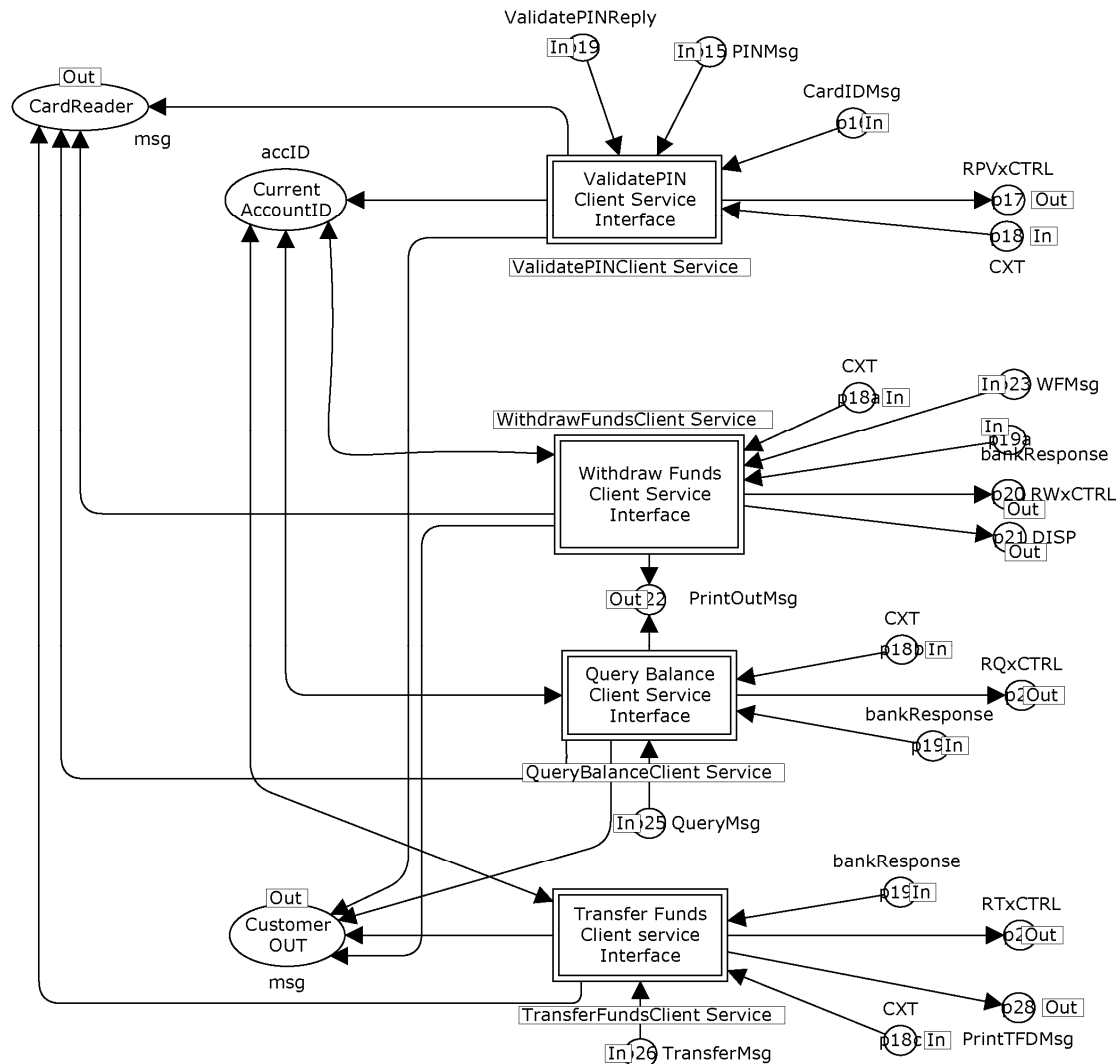


UML TO CPN TRANSFORMATION APPROACH

- Step 2: Component interface layer transformation



■ Component interface CPN layer for ATM client component

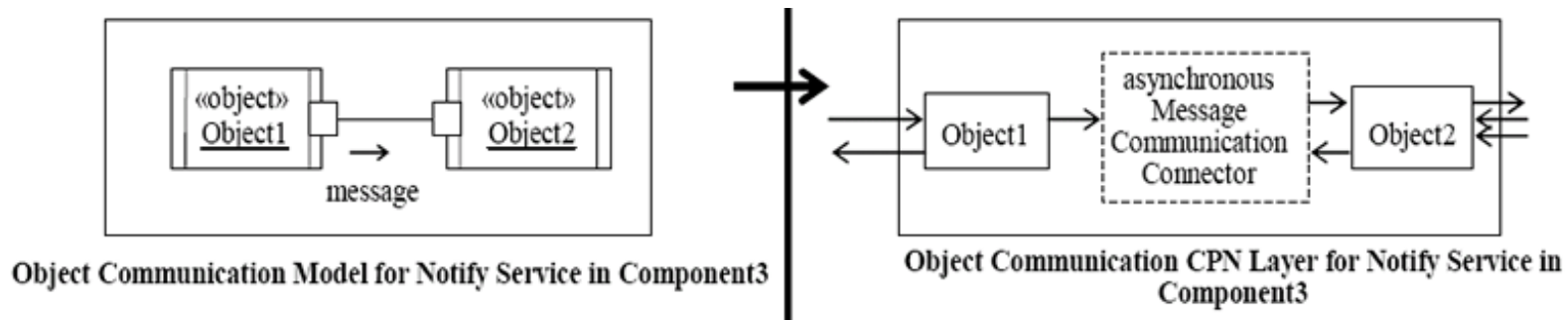


UML TO CPN TRANSFORMATION APPROACH

- Step 3: Object communication layer transformation



- Each service in component interface is realized by objects
 - *Objects supporting each service are transformed to CPN transitions*



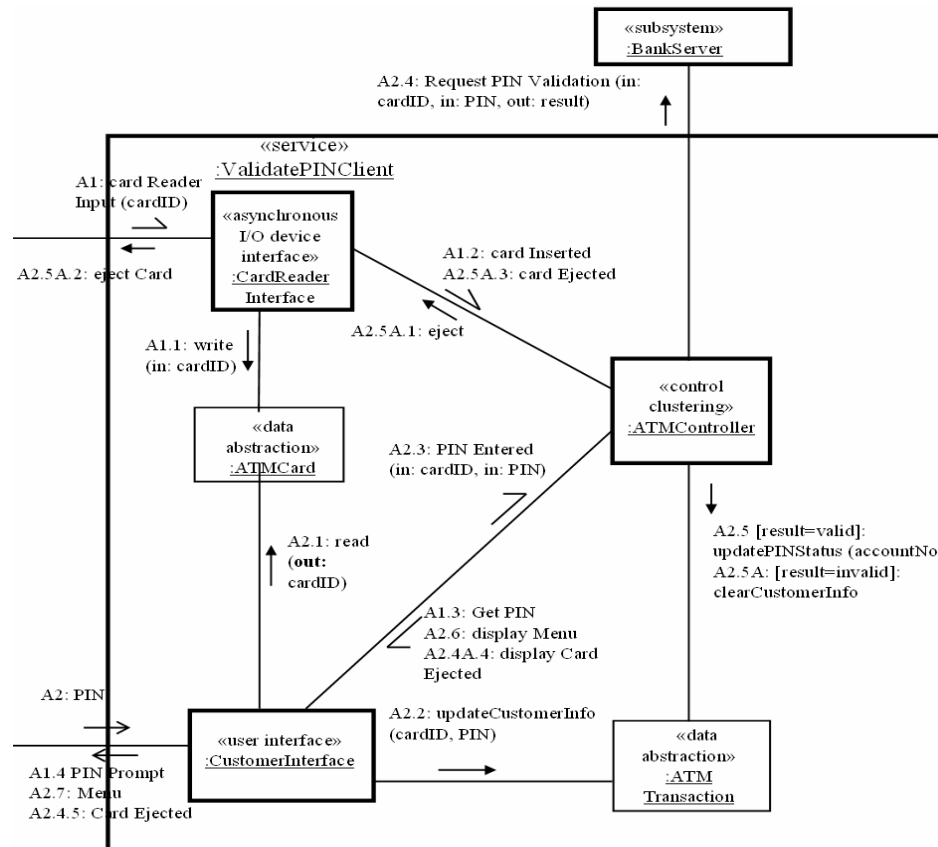
- Message communication between objects are transformed to the corresponding CPN message connector
 - *Similar to message communication between components*
- 2 models are created:
 - *Concrete model*
 - *Abstract model*

UML TO CPN TRANSFORMATION APPROACH

- Step 3: Object communication layer transformation



- Concrete object communication UML model
 - *With all objects supporting a service in interface*
 - *Example - PIN validation service provided by the ATM Client component*

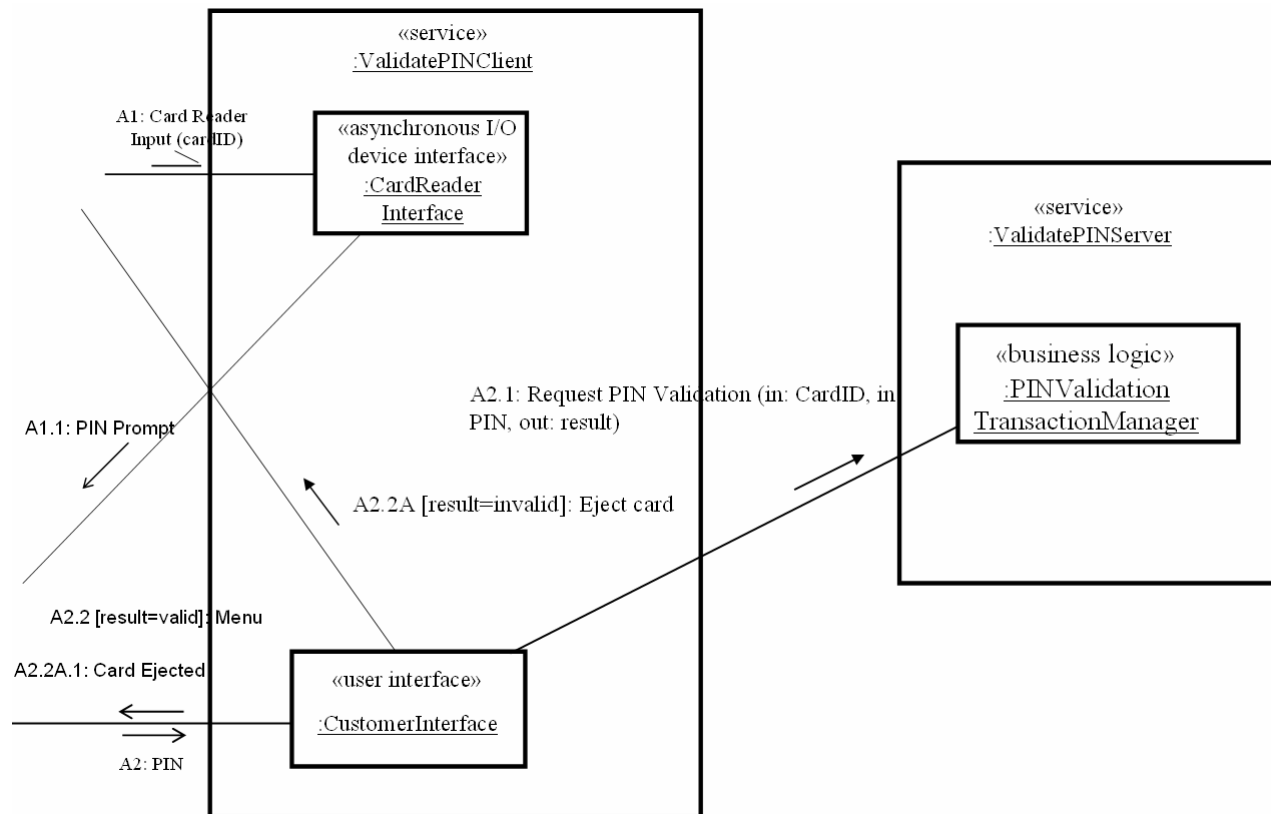


UML TO CPN TRANSFORMATION APPROACH

- Step 3: Object communication layer transformation



- Abstract object communication UML model
 - *With minimal objects supporting each service in interface*
 - *PIN validation service provided by the ATM Client component*

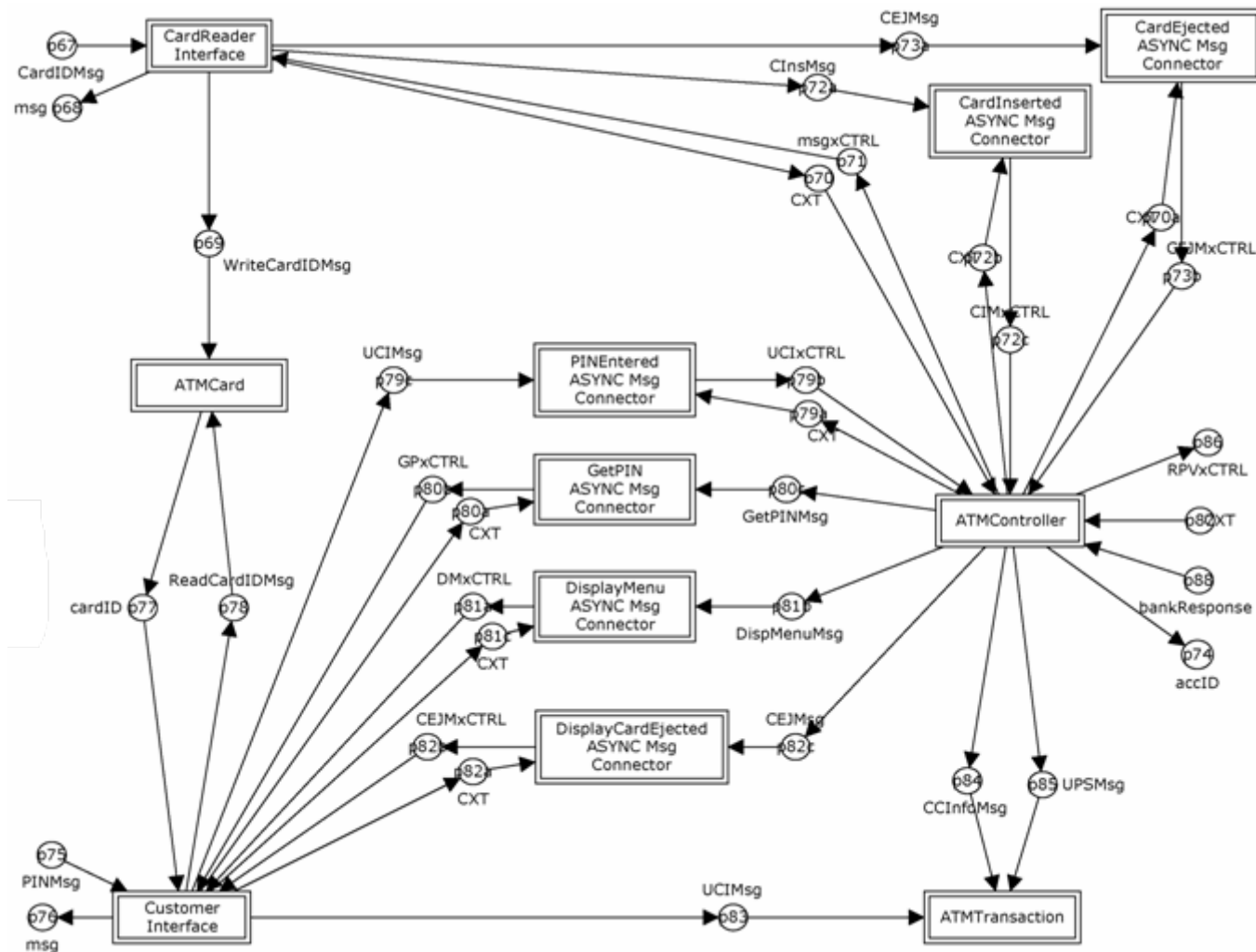


UML TO CPN TRANSFORMATION APPROACH

- Step 3: Object communication layer transformation



- CPN model of concrete object communication UML model for PIN validation service in ATM Client component

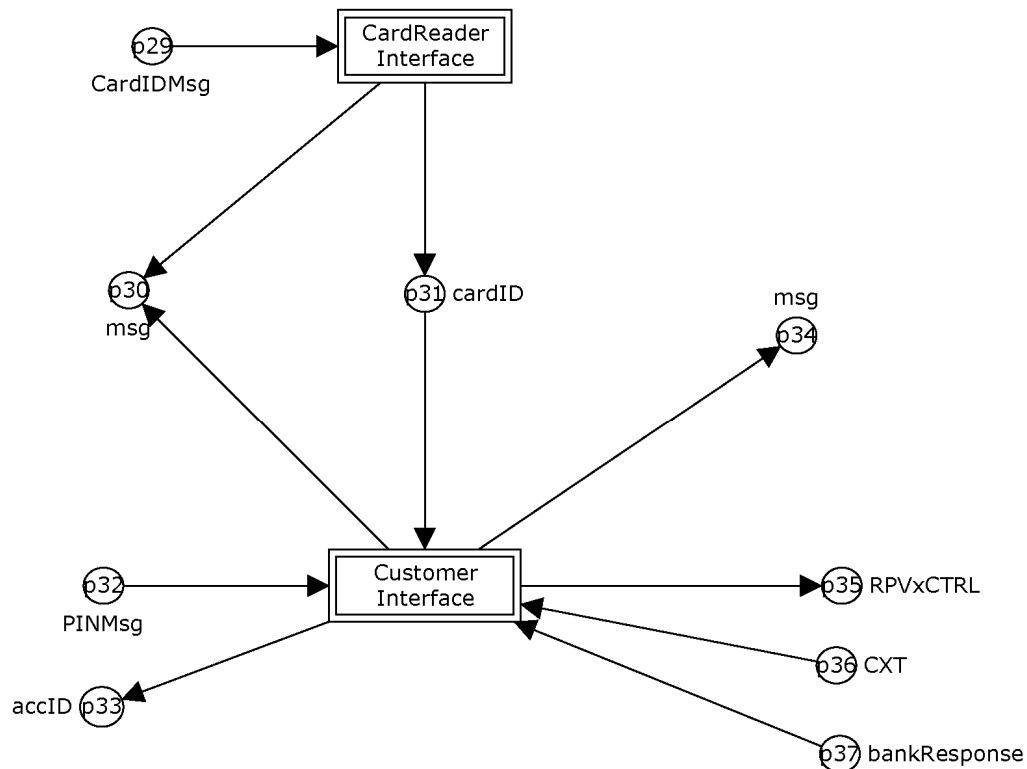


UML TO CPN TRANSFORMATION APPROACH

- Step 3: Object communication layer transformation



- CPN model of abstract object communication UML model for PIN validation service in ATM Client component

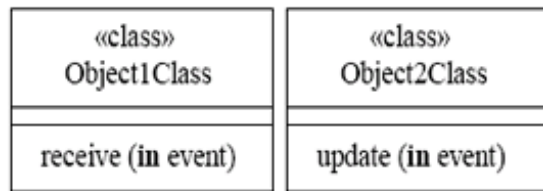


UML TO CPN TRANSFORMATION APPROACH

- Step 4: Operational layer transformation

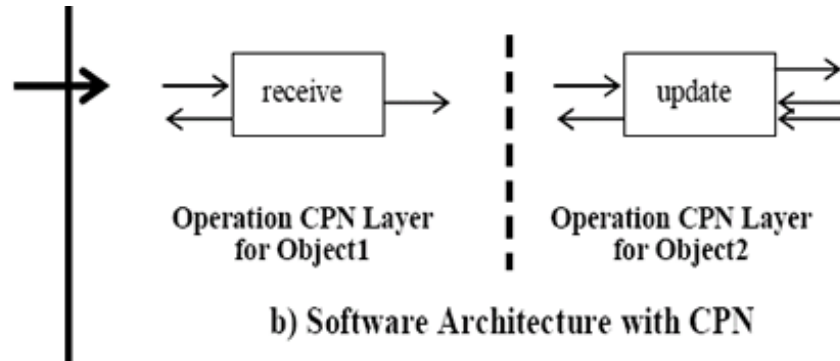


- Each object (class) provides operations
- Operations of each object in the object communication model are mapped to transitions in the operational CPN layer



Class Model

a) Software Architecture with UML



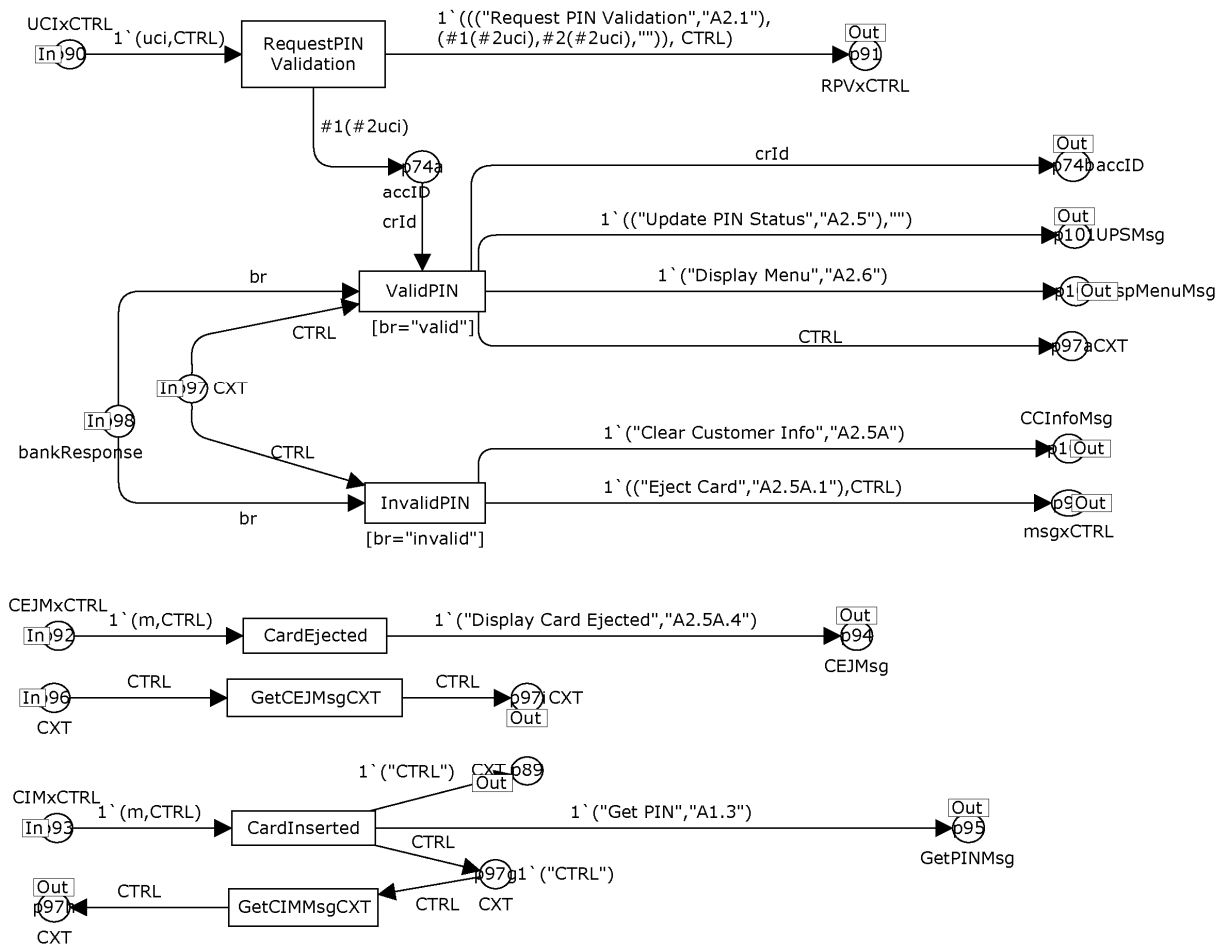
b) Software Architecture with CPN



UML TO CPN TRANSFORMATION APPROACH

- Step 4: Operational layer transformation

- Operational CPN layer for ATM Controller for the PIN validation service in ATM Client component



PRESENTATION AGENDA



- Introduction
- Related Work
- Problem Statement
- Research Approach
- Validation
- Future Work and Conclusion



VALIDATION

- Two case studies for validation
 - *ATM banking system, Elevator system*
- Results show a reduced CPN state space
- Example

Statistics

State Space Nodes: 150155 Arcs: 289266 Secs: 1485 Status: Full
Scg Graph Nodes: 150155 Arcs: 289266 Secs: 35
Home Properties

Home Markings Initial Marking is not a home marking
Liveness Properties

Dead Markings [150155]
Fairness Properties

No infinite occurrence sequences.

(a) State space report for concrete ATM client and ATM server components

Statistics

State Space Nodes: 23198 Arcs: 137565 Secs: 1200 Status: Full
Scg Graph Nodes: 23198 Arcs: 137565 Secs: 20
Home Properties

Home Markings Initial Marking is not a home marking
Liveness Properties

Dead Markings [23198]
Fairness Properties

No infinite occurrence sequences.

(b) State space report for concrete ATM client and abstract ATM server components

PRESENTATION AGENDA



- Introduction
- Related Work
- Problem Statement
- Research Approach
- Validation
- Future Work and Conclusion



- More validation using different case studies
- Tool support
- A more systematic approach to creating the abstract and concrete models
- Abstraction is effective in addressing scalability



TEXAS TECH UNIVERSITY™