Pattern Based Adapters
Formal Design and Mining

Christian Gierds
Service Adapters

With patterns!
Agenda

1. Design with Enterprise Integration Patterns (EIP)
   - Application
   - Details on Translation to CPN
   - Why CPN?

2. Mining
Enterprise Integration Patterns

• by Gregor Hohpe & Bobby Woolf
• collection of integration patterns
• from industry
Application of EIP

Hohpe & Woolf:

- 1\textsuperscript{st} step: use \textit{Enterprise Integration Patterns} (EIP) for modeling
- 2\textsuperscript{nd} step: implement (each EIP)
Judging EIP

Advantages:

• model-driven design

Disadvantages:

• informal description of concepts

(Likely) Problem:

• correct implementation
Idea: Translate EIP to CPN

Pattern Based Messaging Systems - Formal Design and Mining

Simulation
Model Checking
Completion
Execution
Translation in Detail (1/3)

```
var x : Type1;
var z : Type2;
fun enrich x z = ...
```

```
pipe
  x enrich x z
chan
req x
chan x x z
```

```
Content Enricher
```

```
Loan Request
Get Credit Score
Get Banks
Credit Bureau
Rule Base
Recipient List
Aggregator
Bank 2
Bank 1
Bank 3
Loan Broker
Best Quote
```

```
Type1
Type1
Type2
```
Recipent List

```
colset InTyp = Type;
colset SL = list String;
colset valWithList = product Type * SL;
var x : Type;
var l : SL;
fun inList l string = ...
```
Translation in Detail (3/3)

```
var x : Type;
var y : AggregatedType;
fun isFirst x = ...
fun canUpdate x y = ...
fun isComplete y = ...
fun update y x = ...
fun f x = ...
```
Connecting Patterns

```haskell
-- Connecting Patterns

import Captured

colset InTyp = Type;
colset SL = list String;
colset valWithList = product Type * SL;
var x : Type;
var l : SL;
fun inList l string = ...
```

Now unify types and change declarations!
Why Translation to CPN? (1/2)

Simulation in CPN Tools
• finding flaws
• performance analysis

Model Checking
• ASK-CTL
• abstraction
• consider P/T net

Execution
• refine model
• Tsinghua Workflow Management System
• code generation

Pattern Based Messaging Systems - Formal Design and Mining
Why Translation to CPN? (2/2)

Control Flow Completion:

• generate control flow dependencies
Summary: EIP to CPN

• for each pattern, at least one CPN realization
• connecting patterns via pipes and channels
• manual step: defining types, functions, conditions
• Technical report at BPM Center with all patterns
• TODO: Modeling tool with automatic translation
Agenda

1. Design with Enterprise Integration Patterns (EIP)

2. Mining of Adapters
   - “Our” Problem
   - Assumptions & First Solution
Setting

Log CB

Log Cust

Credit Bureau

Loan Broker

Bank 1

Bank 2

Bank 3

Pattern Based Messaging Systems - Formal Design and Mining

Process model(s) not available!

builds on

stores

Patterns

Log B1

Log B2

Log B3
Goal

Formal model, s.t.
• patterns are building blocks
• patterns well-defined
• good fitness

Patterns

Mined Loan Broker

fits all logs

builds on

Log Cust
Log CB
Log B1
Log B2
Log B3
Challenges

• infinite search space
  — idea: restrict number of pattern uses

• correlation of logs
  — idea: correlation by replay (only 2 logs)

![Diagram showing Log₁ and Log₂ with examples of concurrent instances.]

_e.g.: order of concurrent instances in log?_
Our Idea in a Nutshell

Replay each trace of Log₁ with each trace of Log₂

Correlated pair of traces := completed run

Count # of correlated pairs

Assumption: patterns are deterministic

Pattern Based Messaging Systems - Formal Design and Mining
Our Idea in a Nutshell

Pattern Based Messaging Systems - Formal Design and Mining

Patterns

Log_1
!a(5)?b(10)
?b(8)!a(4)
...

Candidate_1

Log_2
!y["AB"]?x["Z"]
?x["V"]!y["X"]
...

Candidate_n

Resulting Model: Candidate with most correlated traces

0.3

0.9

0.7
How to Construct Candidates

List of candidates, s.t.

• type-valid connection of patterns
• candidate borders match services’ interfaces
• exhaustively consider all combinations
Replay in CPN Tools

• for each pair of traces:

Log₁

!a(5)?b(10)

?b(8)!a(4)

…

Log₂

!y("AB")?x("I")

?x("V")!y("X")

…

Trace as transition sequence  Candidate  Trace as transition sequence
Take Home Points

1. Design?
   • translation EIP to CPN
   • made modeling of messaging systems more formal
   • enabling the usage of CPN related tools
   • Editor will follow

With patterns!

2. Mining?
   • use CPN patterns as building blocks
   • restrict on numbers
   • naive approach: check all possible candidates
   • implementation ongoing

THANKS AND KEEP MINING