TC-3
e-Biz Standardization

KAIST 한국과학기술원

Pattern-Based Analysis of BPSS

February 25, 2004

Ja-Hee Kim and Tae-Eog Lee

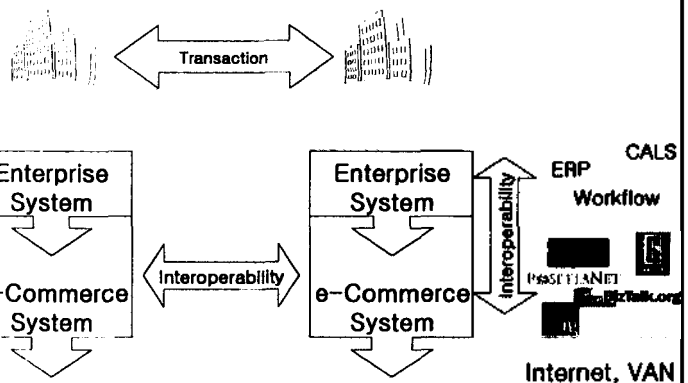
1

Agenda

- Motivation
- Analysis
  - Basic Control Patterns
  - Advanced Branching and Synchronization Patterns
  - Structural Patterns
  - Patterns involving Multiple Instances
  - State Based Patterns
  - Cancellation Patterns
- Conclusion

KAIST 한국과학기술원

e-Commerce

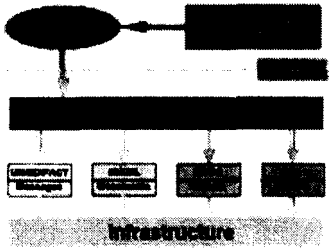


KAIST 한국과학기술원

BCF

- Business Collaboration Framework
- Object

  implementation neutral approach to the exchange of global business requirements.



KAIST 한국과학기술원

View of Business Process Modeling Language

- Workflow: XPDL (WfMC)
- ebXML: BPSS
- WebServices: WSCI, XLANG, BPEL4WS, etc.
- etc.

- Comparison of Expression Powers of Business Process Modeling Languages
  - Pattern based analysis (Aalst 2000)

KAIST 한국과학기술원

20 Patterns

- Aalst et al.
- 20 Patterns
- Derived from Conventional Workflow Systems

pattern	standard					
	XPDL	UML	BPEL	XLANG	WSFL	BPML
Sequence	+	+	+	+	+	+
Parallel Split	+	+	+	+	+	+
Synchronization	+	+	+	+	+	+
Exclusive Choice	+	+	+	+	+	+
Simple Merge	+	+	+	+	+	+
Multi Choice	+	+	+	-	-	-
Synchronizing Merge	+	-	+	-	+	-
Multi Merge	-	-	-	-	-	+/-
Discriminator	-	-	-	-	-	-
Arbitrary Cycles	+	-	-	+	-	+
Implicit Termination	+	-	-	+	+	+
MI without Synchronization	+	-	+	+	-	+
MI with a Priori Design Time Knowledge	+	+	+	+	-	+
MI with a Priori Runtime Knowledge	-	+	-	+	-	-
MI without a Priori Runtime Knowledge	-	-	-	+	-	-
Deferred Choice	-	-	-	+	-	+
Interleaved Parallel Routing	-	-	-/-	+	-	-
Milestone	+	+	+	+	-	-
Cancel Activity	-	+	+	+	+	+
Cancel Case	-	+	+	+	+	+

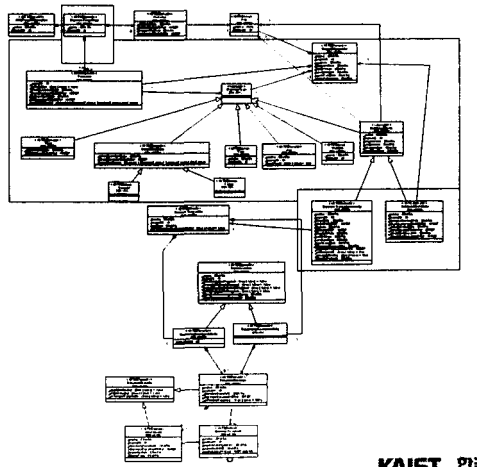
KAIST 한국과학기술원

BPSS

- Business Process Specification Schema
- Based on
  - XML
  - Activity diagram
- Defined by
  - UML
  - DTD (v. 1.01)
  - XML schema (v. 1.10)

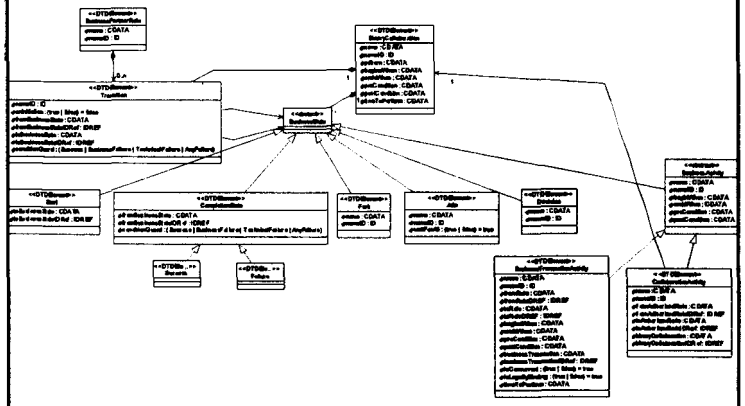
KAIST 한국과학기술원

UML Model for BPSS



KAIST 한국과학기술원

Choreography



KAIST 한국과학기술원

Basic Control Patterns

- Sequence
  - Execute activities in sequence
  - Usage
    - Start, Success, Failure, BusinessTransactionActivity, CollaborationActivity
    - Connected by Transition

KAIST 한국과학기술원

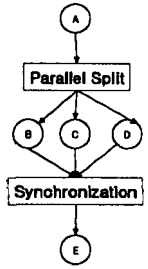
Basic Control Patterns

- Parallel Split
  - Execute activities in parallel

```
<Fork
        name = "Parallel Split"
        nameID = "12345B678"
        type = "OR"/>
```

- Synchronization
  - Synchronize two parallel threads of execution

```
<Join
        name = "Synchronization"
        nameID = "12345B876"
        waitForAll = "true"/>
```



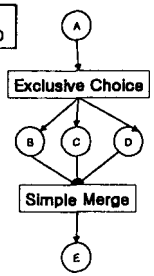
KAIST 한국과학기술원

Basic Control Patterns

- Exclusive Choice
  - Choose one execution path from many alternatives

```
<Transition
        fromBusinessState = "Exclusive Choice"
        fromBusinessStateIDEF = "12345B874"
        ...
        Condition Guard = "Success">
        <ConditionExpression .../></Transition>
<Decision
        name = "Exclusive Choice"
        nameID = "12345B874"/>
```

BPSS Ver 1.10



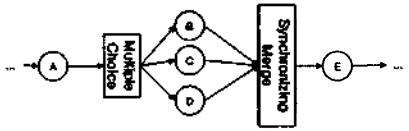
- Simple Merge
  - Merge two alternative execution paths

```
<Join
        name = "Simple Merge"
        nameID = "12345B876"
        waitForAll = "false"/>
```


KAIST 한국과학기술원

Advanced Branching and Synchronization Patterns

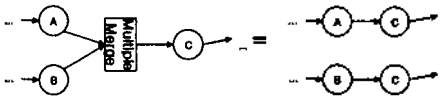
- Multiple Choice
  - Choose several execution paths from many alternatives
  - Not support
- Synchronizing Merge
  - Merge many execution paths. Synchronize if many paths are taken. Simple merge if only one execution path is taken
  - Not support



KAIST 한국과학기술원

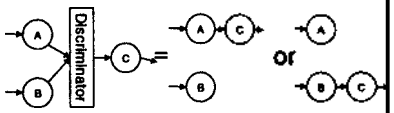
Advanced Branching and Synchronization Patterns

- Multiple Merge
  - Merge many execution paths without synchronizing
  - Not support



- Discriminator
  - Merge many execution paths without synchronizing. Execute the subsequent activity only once
  - Example

```
<Fork     ..
          type = "OR" />
<Join
          name = "Discriminator"
          nameID = "12345B874"
          waitForAll = "false"/>
```

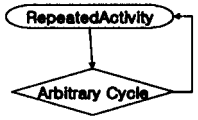


KAIST 한국과학기술원

Structural Patterns

- Arbitrary Cycles
  - Execute workflow graph without any structural restriction on loops

```
<BusinessTransactionActivity
          name="RepeatedActivity"
          nameID="122A39A3"
          .../>
<Decision
          name="Arbitrary Cycle"
          nameID="2323A1B7"/>
<Transition
          fromBusinessState ="RepeatedActivity"
          fromBusinessIDREF ="122A39A3"
          toBusinessState ="Arbitrary Cycle"
          toBusinessIDREF ="2323A1B7"...
<Transition
          fromBusinessState ="Arbitrary Cycle"
          fromBusinessIDREF ="2323A1B7"...
          toBusinessState ="RepeatedActivity"
          toBusinessIDREF ="122A39A3"
```



- Implicit Termination
  - Terminate if there is nothing to be done

KAIST 한국과학기술원

Patterns Involving Multiple Instances

- MI without synchronization
  - Generate many instances of one activity without synchronizing them afterwards

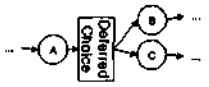
```
<BusinessTransactionActivity
          name="MI without Synchronization"
          ...
          isConcurrent="true"
          .../>
```

- Not Support
  - MI with a priori known design time knowledge
  - MI with a priori known runtime knowledge
  - MI with no a priori runtime knowledge

KAIST 한국과학기술원

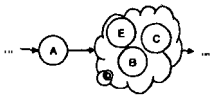
State-based patterns

- Deferred Choice
  - Execute one of the two alternatives threads. The choice which thread is to be executed should be implicit. (ver 1.10)

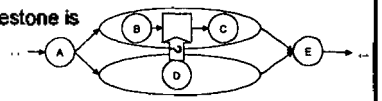



```
<Fork
        name = "Deferred Choice"
        nameID = "12A45B678"
        type = "XOR"/>
```

- Interleaved Parallel Routing
  - Execute two activities in random order, but not in parallel.
  - Not Support



- Milestone
  - Enable an activity until a milestone is reached
  - Not Support



Cancellation Patterns

- Business Transaction is an atomic unit of work in a trading arrangement.
- We can make Business Transaction for Cancellation.
- Patterns
  - Cancel Activity
    - Cancel (disable) an enabled activity
  - Cancel Case
    - Cancel (disable) the process

	BPSS v1.01	BPSSv1.10	BPEL4WS	XLANG
Sequence	+	+	+	+
Parallel Split	+	+	+	+
Synchronization	+	+	+	+
Exclusive Choice	+	+	+	+
Simple Merge	+	+	+	+
Multi Choice	-	-	+	-
Synchronizing Merge	-	-	-	-
Multi Merge	-	-	-	-
Discriminator	-	+	+	-
Arbitrary Cycles	+	+	+/-	-
Implicit Termination	+	+	+	-
MI without Synchronization	+	+	+	+
MI with a Priori Design Time Knowledge	-	-	+	+
MI with a Priori Runtime Knowledge	-	-	-	-
MI without a Priori Runtime Knowledge	-	-	-	-
Deferred Choice	-	+	+	+
Interleaved Parallel Routing	-	-	+/-	-
Milestone	-	-	-	-
Cancel Activity	+	+	+	+
Cancel Case	+	+	+	+

Conclusion

- In the View of Expression Power
  - BPSS ver 1.10 is Improved from BPSS ver 1.01
  - Comparison with business process languages for WebServices
    - BPEL4WS > BPSS > XLANG
- Application
  - Improving BPSS
    - Should BPSS supply all workload patterns? NO!
  - Developing the conversion rule
    - BPSS ←→BPEL4WS, XLANG
    - using XLST