

BPML 101

Implementing the BPML Specification

Jeanne Baker
Director of BPI Solutions, Sterling Commerce
Director, BPMI.org

Ismaël Ghalimi
CEO, Intalio, Inc.
Chairman, BPMI.org

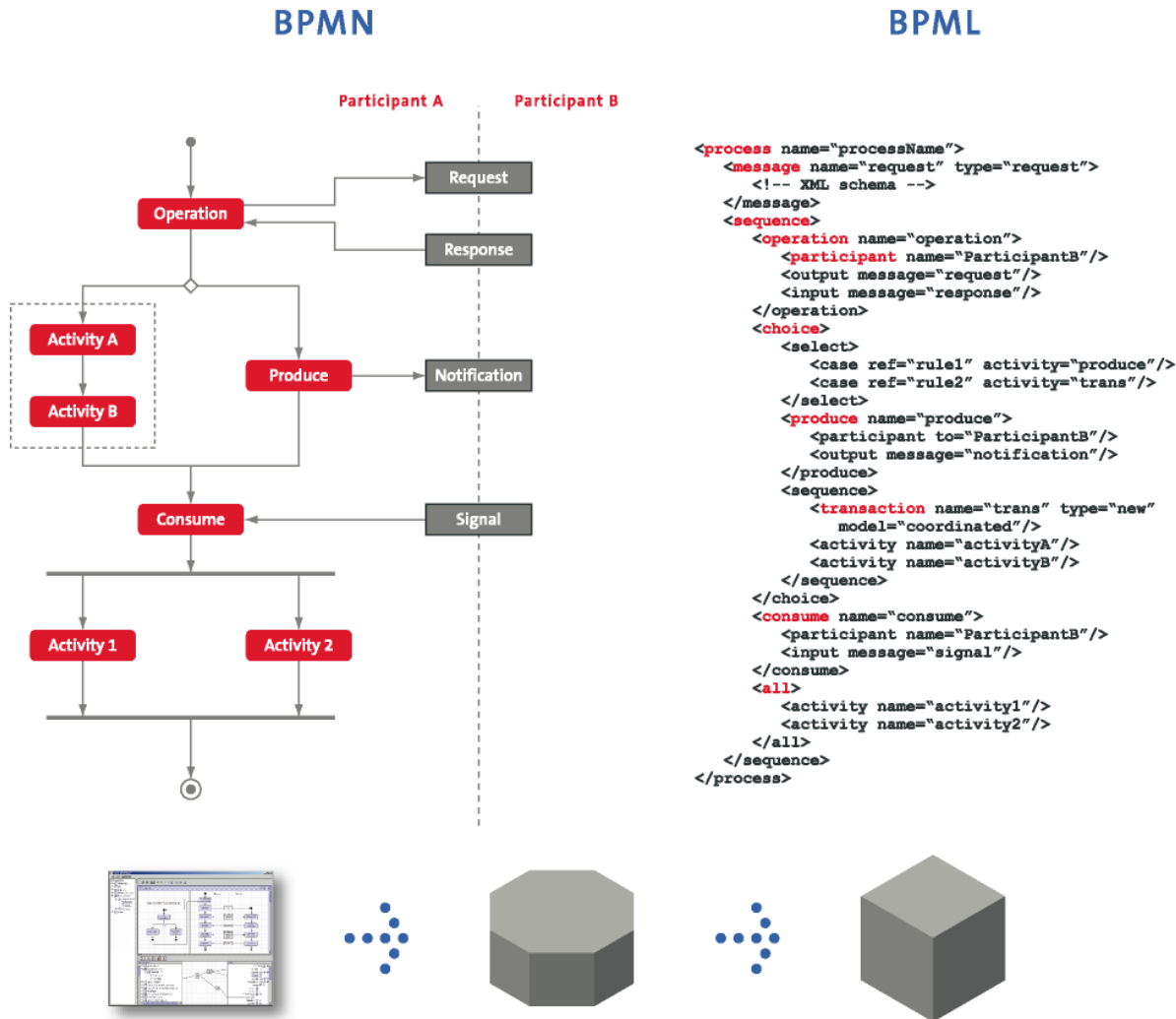
-
-
-
-
-

What is BPML?

- ❖ Business Process Modeling Language
- ❖ XML-based declarative language
- ❖ Created by the Business Process Management Initiative (BPMI)

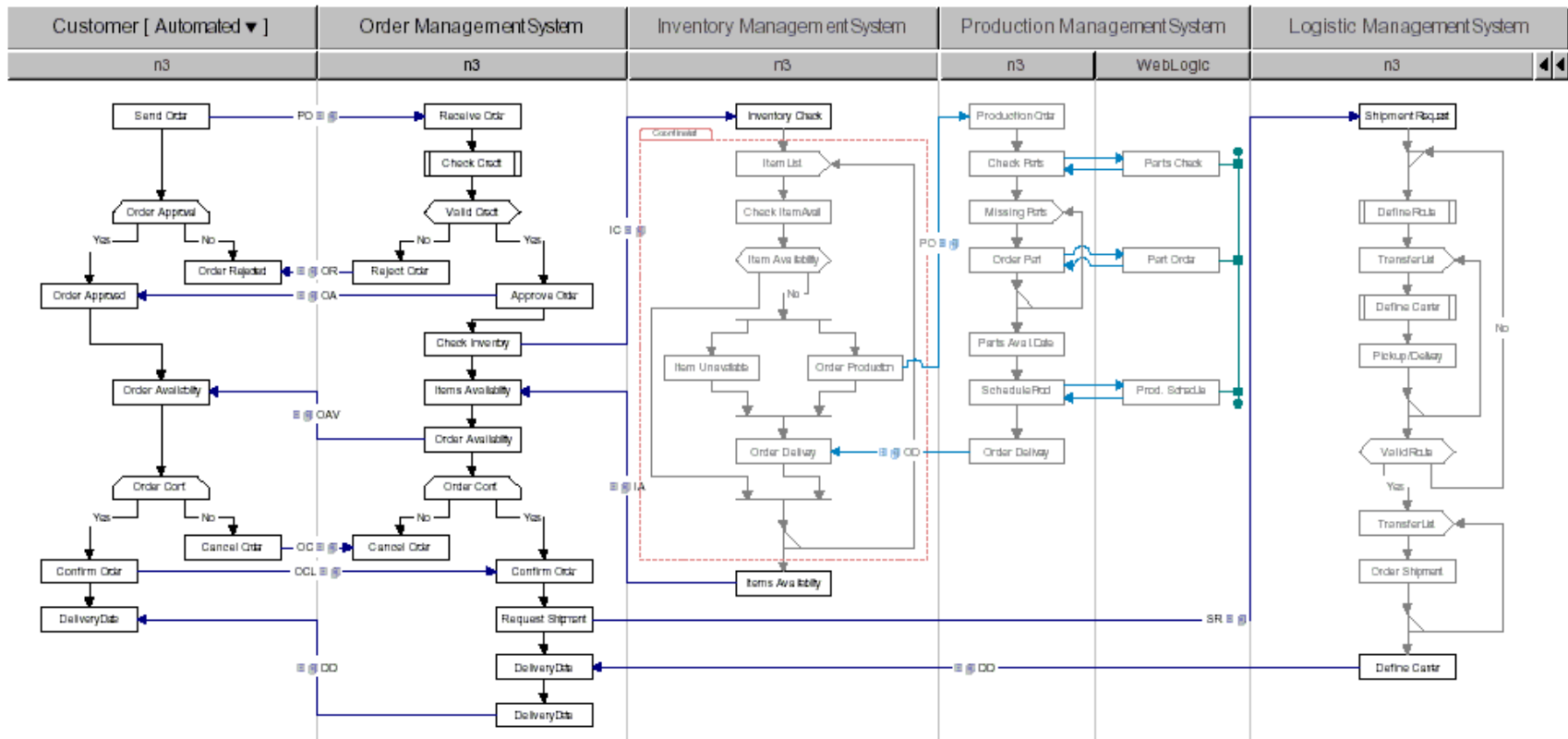
BPML Process Modeling

Empowering Business Analysts to Design Executable Business Processes



Modeling Complex Processes

Modeling Processes with 10s of Participants and 1000s of Activities



BPML has an XML syntax

Facilitating collaboration and shortening the learning curve

```
<sequence>
```



```
<operation name='Check Inventory'>
```

```
<participant name='InventoryService' />
```

```
<output message='checkStockRequest'>
```

```
<assign to='ISBN'>1-56592-488-6</assign>
```



```
</output>
```

```
<input message name='checkStockResponse'>
```

```
<assign to='foundBook' from='InStock' />
```



```
</input>
```

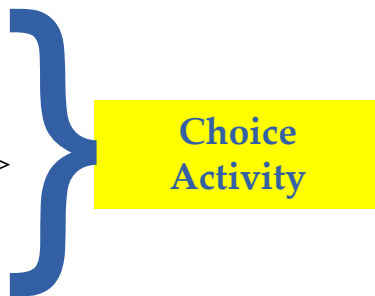
```
</operation>
```

```
<choice>
```

```
<select>
```

```
<case ref="BookInStock" activity="proceed"/>
```

```
<case ref="BookInStock" negative="true" activity="stop"/>
```



```
</select>
```

BPML's Unique Features

Business Process Management Made Easy

- ❖ End-to-End Process Modeling
- ❖ Control-Flow/Data-Flow Separation
- ❖ Produce/Consume Messaging
- ❖ Dynamic Control-Flow
- ❖ Transparent Persistence
- ❖ Embedded Business Rules
- ❖ Nested Processes
- ❖ Distributed Transactions
- ❖ Process-Oriented Exception Handling
- ❖ Underlying Mathematical Model



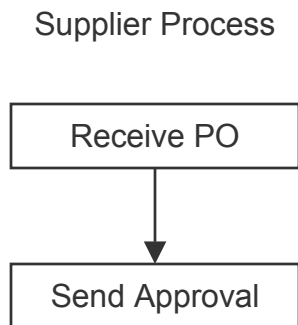
Illustrating These Features

End-to-End Process Modeling

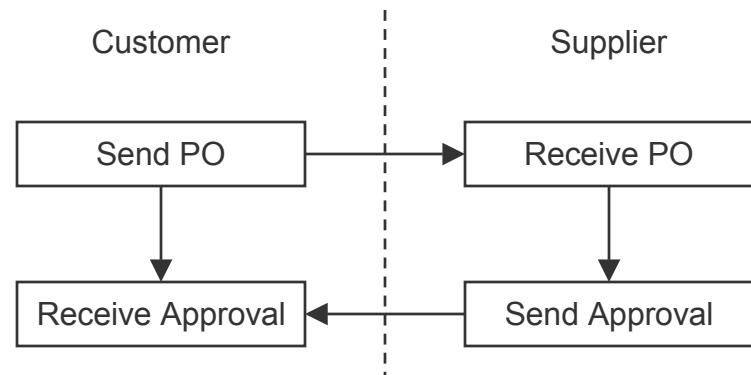
Get the Full Picture

- ❖ End-to-end business processes involve multiple participants
- ❖ Participants can be business partners but also IT systems
- ❖ Traditional approaches only show one participant's view
- ❖ BPML can show the end-to-end process

❖ Without BPML



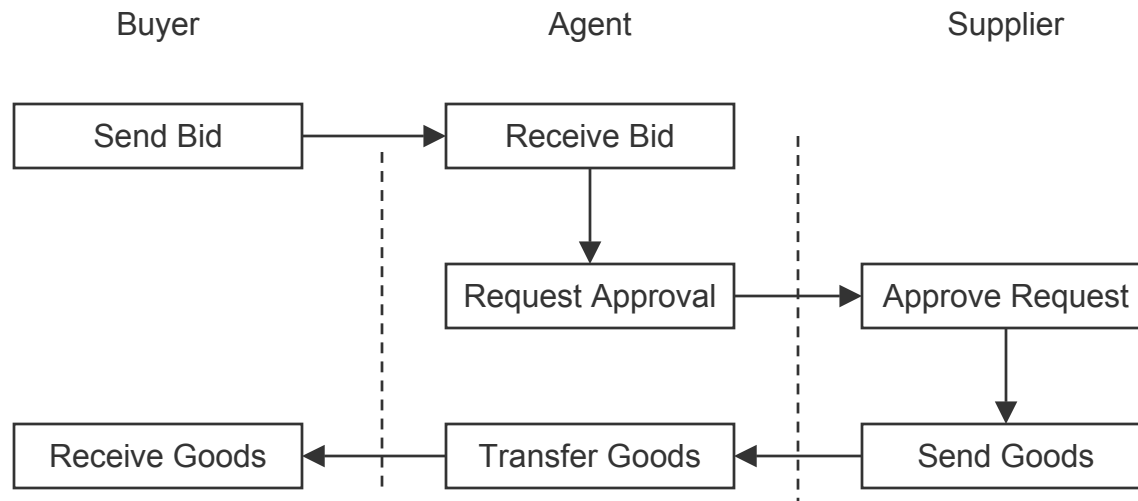
❖ With BPML



End-to-End Process Modeling

Another Example

❖ An escrow agent facilitates trading between enterprises

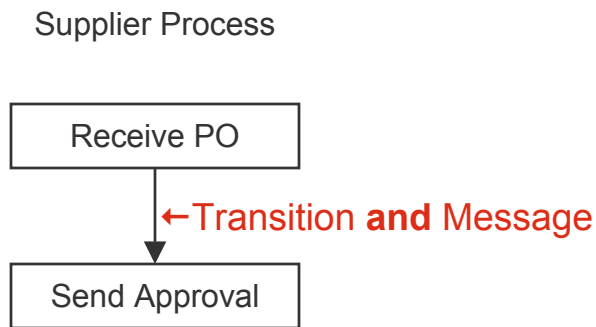


Control-Flow/Data-Flow Separation

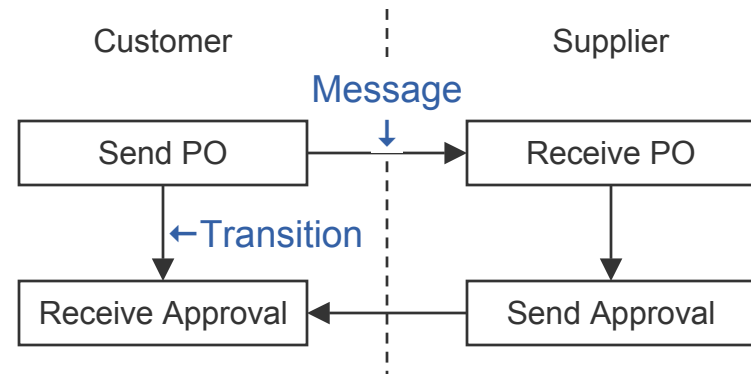
Get a Clear Picture

- ❖ Control-Flow and Data-Flow are two separate concepts
- ❖ Traditional approaches treat them as one and the same
- ❖ This leads to poor process maintainability and reuse
- ❖ BPML cleanly separates Control-Flow and Data-Flow

❖ Without BPML



❖ With BPML



Produce/Consume Messaging

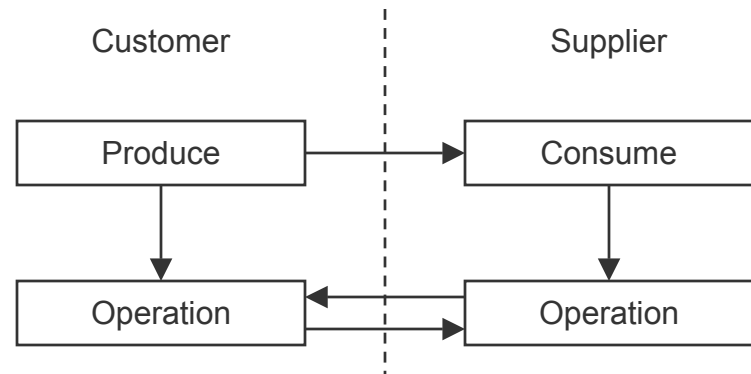
Messaging Made Easy

- ❖ Traditional approaches make EAI very complex
- ❖ Developers must deal with multiple messaging paradigms
- ❖ Different APIs dictate different messaging patterns
- ❖ BPML makes it simple with the Produce/Consume pattern

❖ Without BPML

- ❖ Synchronous Transaction
- ❖ Asynchronous Transaction
- ❖ Remote Procedure Call
- ❖ Request/Reply
- ❖ Publish/Subscribe
- ❖ You-Name-It

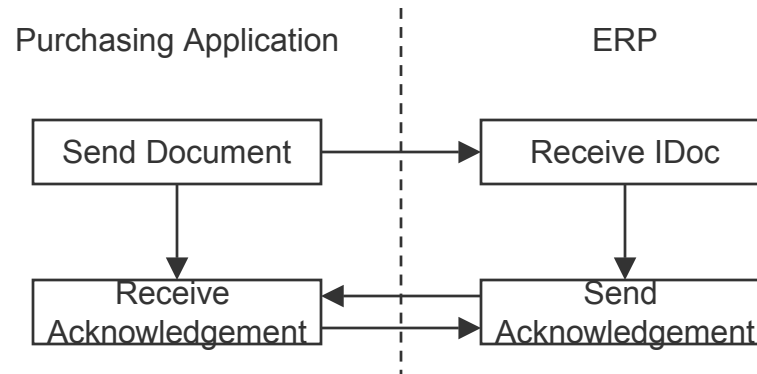
❖ With BPML



Produce/Consume Messaging

Another Example

❖ Application Integration



Dynamic Control-Flow

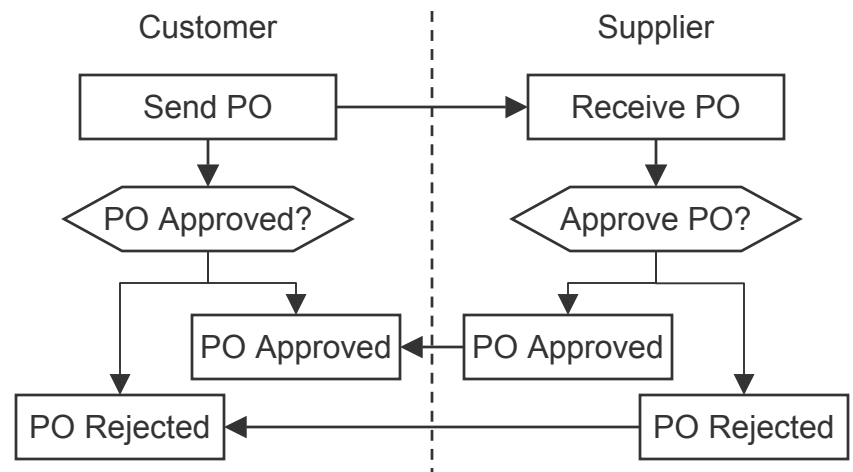
Modeling Dynamic Business Processes

- ❖ Business processes are dynamic
- ❖ Traditional approaches usually support static processes only
- ❖ BPML supports dynamic branching based on message types
- ❖ BPML supports dynamic branching based on participants

❖ Without BPML

- ❖ One queue + software logic?
- ❖ Multiple queues + synchronization?
- ❖ Hardcode EAI adapters?
- ❖ Hardcode B2B adapters?
- ❖ How many tools must I use?
- ❖ How can I synchronize everything?

❖ With BPML



Transparent Persistence

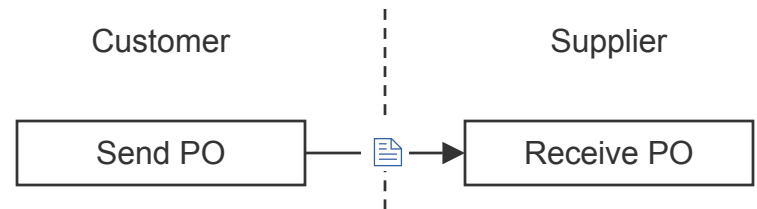
Do you really need to be a Rocket Scientist to store a Purchase Order?

- ❖ Business processes are long-running transactions
- ❖ Process data must be made persistent for later reuse
- ❖ Traditional approaches ask developers to hardcode persistence
- ❖ BPML supports transparent process data persistence

❖ Without BPML

- ❖ Write an entity EJB
- ❖ Map the XML message onto the EJB
- ❖ Create a new SQL database table
- ❖ Map the EJB onto the SQL table
- ❖ Deploy the EJB
- ❖ Do it all again for new XML schema

❖ With BPML



When received,
the PO becomes part
of the process state

Embedded Business Rules

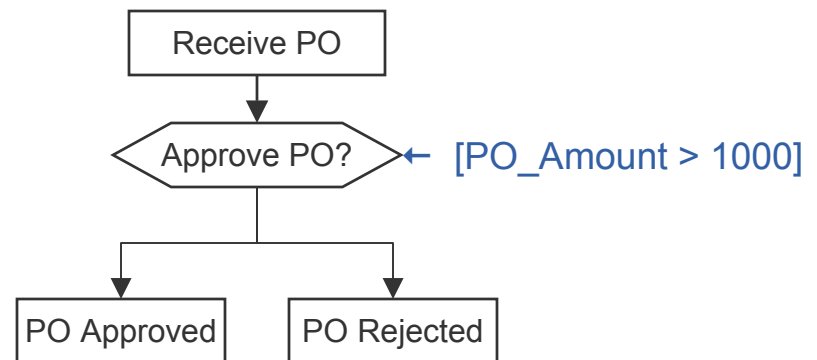
The Business Logic in Business Processes

- ❖ Business rules drive the execution of dynamic business processes
- ❖ Business rules are usually updated frequently
- ❖ Traditional approaches hardcode rules externally to the process
- ❖ BPML supports embedded business rules

❖ *Without BPML*

- ❖ Hardcode the business rule
- ❖ Map the business rule to a branching
- ❖ How do I synchronize everything?

❖ *With BPML*



Nested Processes

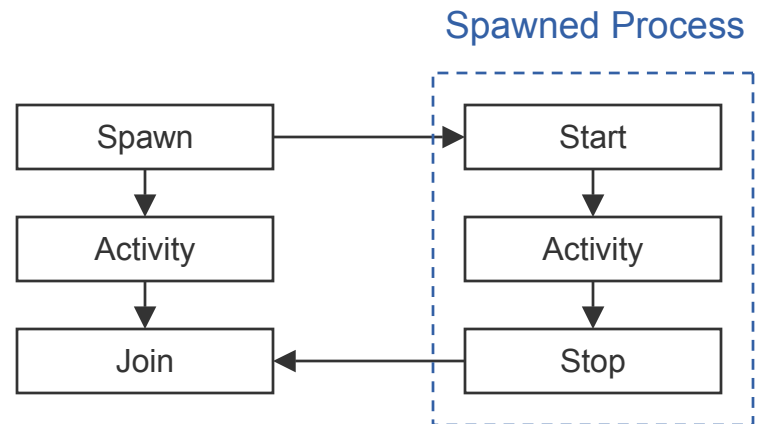
Where Business Process Management Meets Distributed Computing

- ❖ A nested process shares the data of its parent process
- ❖ Nested processes are required when things happen in parallel
- ❖ Traditional approaches only support sub-processes
- ❖ BPML supports sub-processes and nested processes

❖ Without BPML

- ❖ Develop two independent processes
- ❖ Hardcode a shared state mechanism
- ❖ Synchronize the two processes
- ❖ Get a degree in parallel computing?

❖ With BPML



-
-
-

Nested Processes

Additional uses

❖ Function Reuse

❖ Decision Making

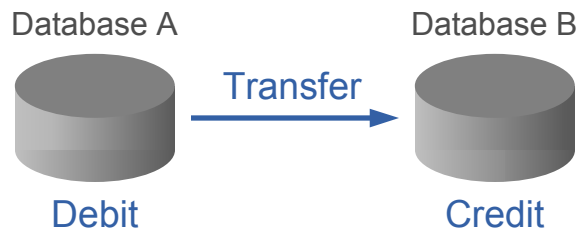
- Spawning processes based on process state

❖ Asynchronous activities

Distributed Transactions

Business Process Modeling for Mission-Critical Processes

❖ Mission-critical processes deal with distributed transactions



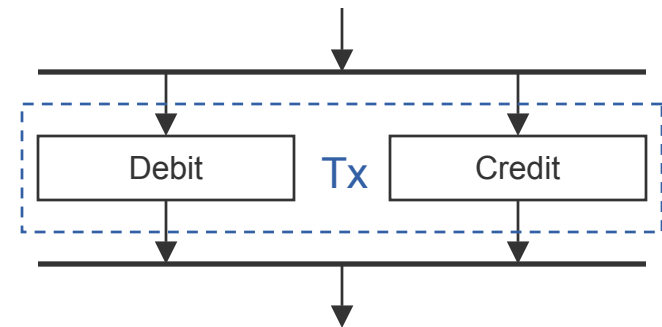
- ❖ Atomic
- ❖ Consistent
- ❖ Isolated
- ❖ Durable

The process must preserve the ACID properties of distributed transactions

❖ *Without BPML*

- ❖ Write a transaction with a TP Monitor
- ❖ Write a compensating transaction
- ❖ Deploy transactions
- ❖ Map transactions to the process
- ❖ Propagate transaction contexts

❖ *With BPML*



Process-Oriented Exception Handling

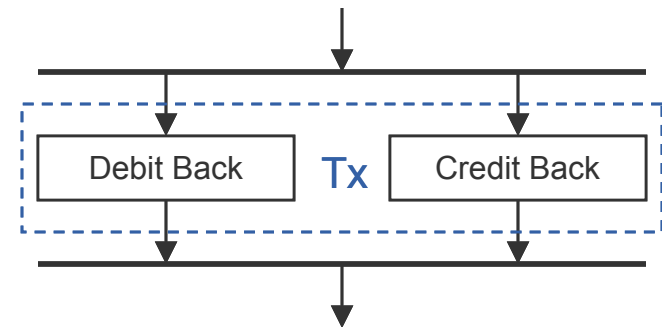
The Devil is in the Exceptions

- ❖ Business processes remain simple as long as they don't break
- ❖ Exceptions (failures) make processes much more complex
- ❖ Traditional approaches offer poor support for exception handling
- ❖ BPML supports process-oriented exception handling

❖ Without BPML

- ❖ Hardcode compensating transactions
- ❖ Map transactions to the process
- ❖ Propagate transaction contexts

❖ With BPML



-
-
-

Underlying Mathematical Model

What makes BPML really work

- ❖ BPML is based on the Pi-Calculus Mathematical Model
- ❖ Pi-Calculus offers consistency checking
- ❖ Pi-Calculus offers deadlock detection
- ❖ Pi-Calculus offers bottleneck detection
- ❖ Pi-Calculus enables process optimization
- ❖ Pi-Calculus is also used by Microsoft's XLANG

Take Away Points

What should be known about BPML

- ❖ Modeling complex business processes with BPML is possible
- ❖ BPML makes it easier than any other approach
- ❖ BPML bridges the gap between modeling and execution
- ❖ BPML can be used by business analysts and software engineers
- ❖ BPML solves problems artificially created by technology
- ❖ BPML is abstracted from the underlying IT infrastructure
- ❖ BPML does not sacrifice performance or reliability
- ❖ BPML has a smooth learning curve

-
-
-

Single Take Away Point

What should really be known about BPML

BPML is available Today!

Log on to www.bpml.org