6th IPROMS Virtual Conference
15-26 November 2010

Interaction- and Event-Based Management of Processes in Service-Oriented Infrastructures

Thomas Schlegel, Sebastian Dusch, Krešimir Vidačković

Agenda

- Motivation
- Basics
- Event Model
- System Architecture
- Technology and Implementation
- Test Scenario
- Conclusion

Motivation

- Rising complexity of service infrastructures
  - New services permanently emerging
  - Services integrated into cross-company business processes (e.g. claims settlement process of insurance companies)
  - Interactions with integrated services and users
- Monitoring of decentralized service platforms as key problem
  - No central control
  - Different stakeholders
  - Unforeseen events and interactions
- Architecture required to support:
  - Exchange of information and triggers between processes
  - Management platform for monitoring and user interactions

Basics

- Execution of distributed business processes
  - WS-BPEL (Web Services Business Process Execution Language) as de-facto standard for executable IT-based business processes
  - Monitoring difficult, especially on different engines and servers

- Complex Event Processing (CEP)
  - Technology for systematic processing of event objects from different sources
  - Support of loose coupling through publish/subscribe-concept
  - Support of real-time systems through immediate reactions on detected relations between incoming events (event patterns)

- **Utilization of CEP technology for monitoring of distributed business processes on multiple engines**

Basics

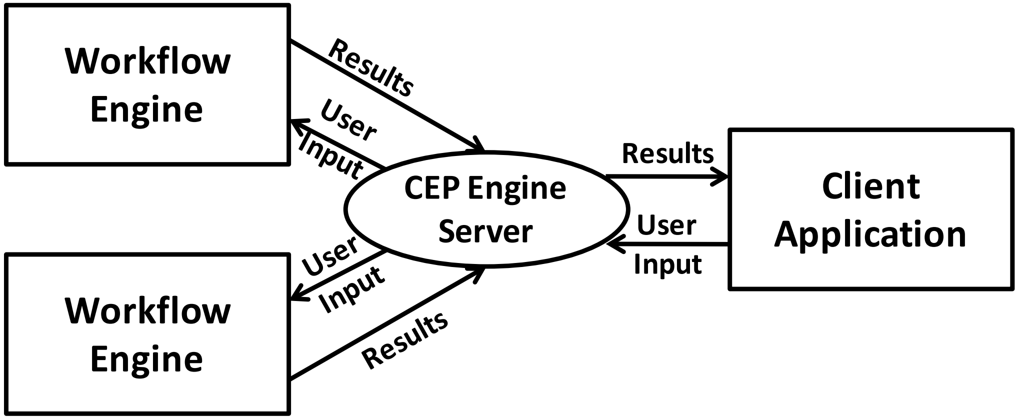
- Interaction with process infrastructures
  - Explicit interactions
    - Foreseen interactions with the executed business process
    - Modeling of human interactions using WS-HumanTask and BPEL4People
  - Implicit interactions
    - Interactions in unforeseen situations, between processes or outside the process

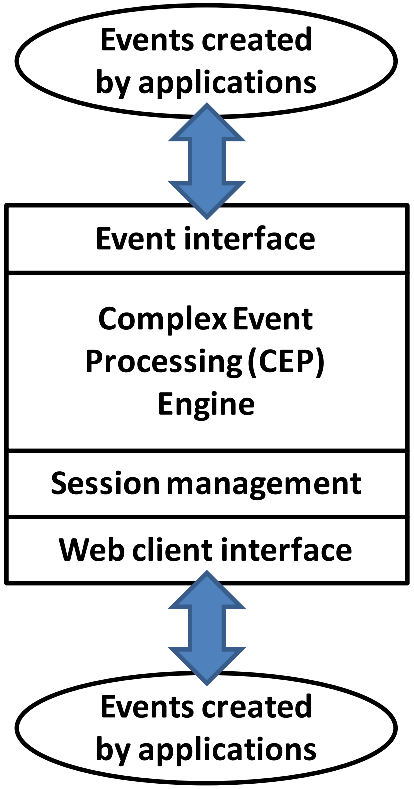
- **Utilization of CEP technology for user interactions with process infrastructures using a dashboard as graphical user interface**

Event Model

- Events as main concept for monitoring distributed business processes over different process engines
- Events are generated by different BPEL engines
- Event model defined using XML schema
- Different event types:
  - **Notification event:** information regarding status changes
  - **Time event:** event query may be embedded within event object regarding elapsed time frames
  - **Interaction event:** user interface embedded within event object

System Architecture

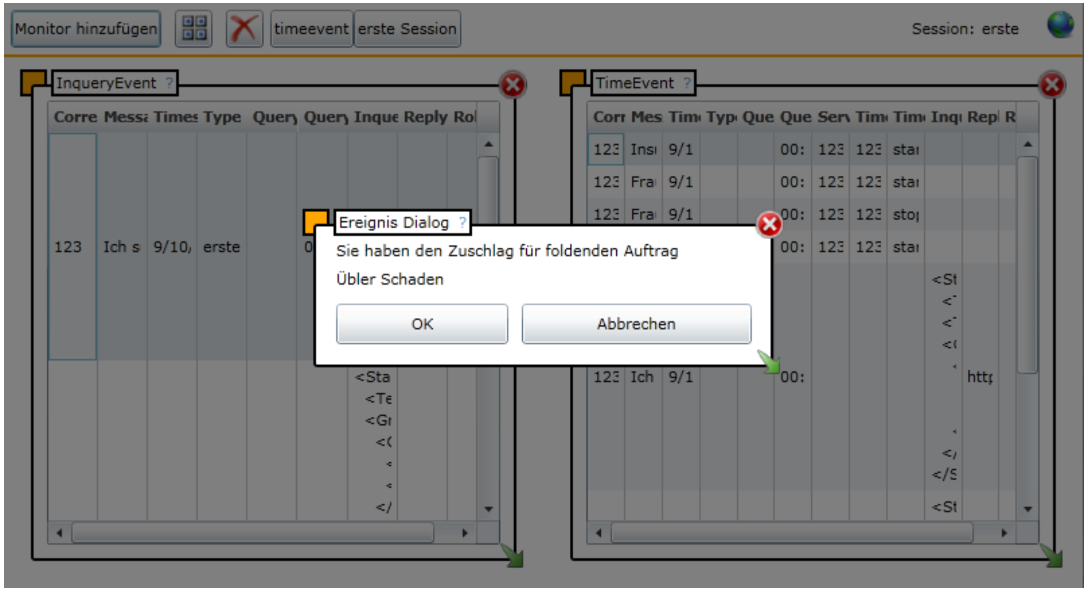



Events created by applications
Event interface
Complex Event Processing (CEP) Engine
Session management
Web client interface
Events created by applications

- Different BPEL engines generate events during process execution which are sent to CEP engine
- CEP engine processes events and sends query results to user interaction client (dashboard)
- User interactions are sent back to CEP engine which routes them to corresponding BPEL engine influencing process flow

Technology and Implementation

- Apache ODE (open source) as BPEL engine
- Esper (open source) as CEP engine
- XAML (eXtensible Application Markup Language) for graphical user interface embedded within event objects
  - Directly displayable and executable by runtime environment
- Microsoft .NET and Silverlight for dashboard environment



Corre	Messa	Times	Type	Query	Query	Inque	Reply	Rol
123	Ich s	9/10,	erste	0		<Sta <Te <Gr <(< < </		

Corr	Mes	Tim	Typ	Que	Que	Serv	Tim	Tim	Inq	Repl	R
123	Insı	9/1			00:	123	123	star			
123	Fra	9/1			00:	123	123	star			
123	Fra	9/1			00:	123	123	stoj			
					00:	123	123	star			
123	Ich	9/1			00:				<St < < <(</ </S <St	http	

Sie haben den Zuschlag für foldenden Auftrag

Übler Schaden

Technology and Implementation: Definition of Events

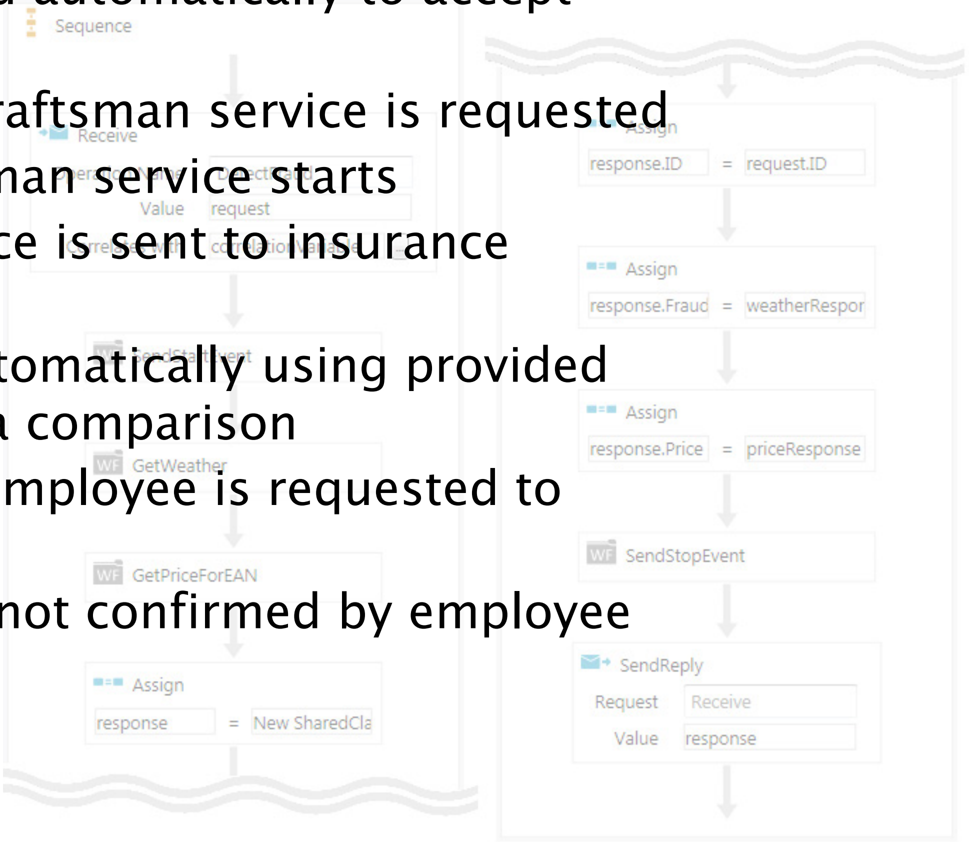
- Events are defined on the basis of XML
- Example – Interaction Event definition:

```
<xs:complexType name="InquiryFormType" mixed="true">
 <xs:sequence>
  <xs:element name="Content">
   <xs:complexType>
    <xs:sequence>
     <xs:any namespace="##any" processContents="skip" />
    </xs:sequence>
   </xs:complexType>
  </xs:element>
  <xs:element name="ReturnControlNames" >
   <xs:complexType>
    <xs:sequence>
     <xs:element name="ReturnControlName" type="xs:string" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
   </xs:complexType>
  </xs:element>
 </xs:sequence>
 <xs:attribute name="lang" type="xs:string" />
</xs:complexType>
```

Test Scenario and Evaluation

- Solution evaluated with test scenario from insurance domain
- Distributed claims settlement process of insurance company

1. Several services of craftsmen are provided on service marketplace
2. Selected craftsman is requested automatically to accept order
3. If defined time elapsed, next craftsman service is requested
4. If craftsman accepts order, human service starts
5. When craftsman finishes, invoice is sent to insurance process
6. Invoice is checked for fraud automatically using provided web service with reference data comparison
7. On potential fraud, insurance employee is requested to check invoice
8. Invoice is only paid, if fraud is not confirmed by employee



Conclusion

- Monitoring of distributed business processes on different engines and servers accomplished with CEP technology
- Independence from specific BPEL engine
- Notifications and user interactions via event objects
- Graphical user interface automatically generated using XAML embedded within event objects from CEP engine

6th IPROMS Virtual Conference
15-26 November 2010

Interaction- and Event-Based Management of Processes in Service-Oriented Infrastructures

Jun.-Prof. Dr. Thomas Schlegel
Technische Universität Dresden
+49 351 463-39177
Thomas.Schlegel
[at]tu-dresden.de

Krešimir Vidačković
Fraunhofer IAO
+49 711 970-2340
kresimir.vidackovic
[at]iao.fraunhofer.de