



Implementierung einer Datenbankanbindung zur Speicherung und Abfrage von Mikroprozessereignissen

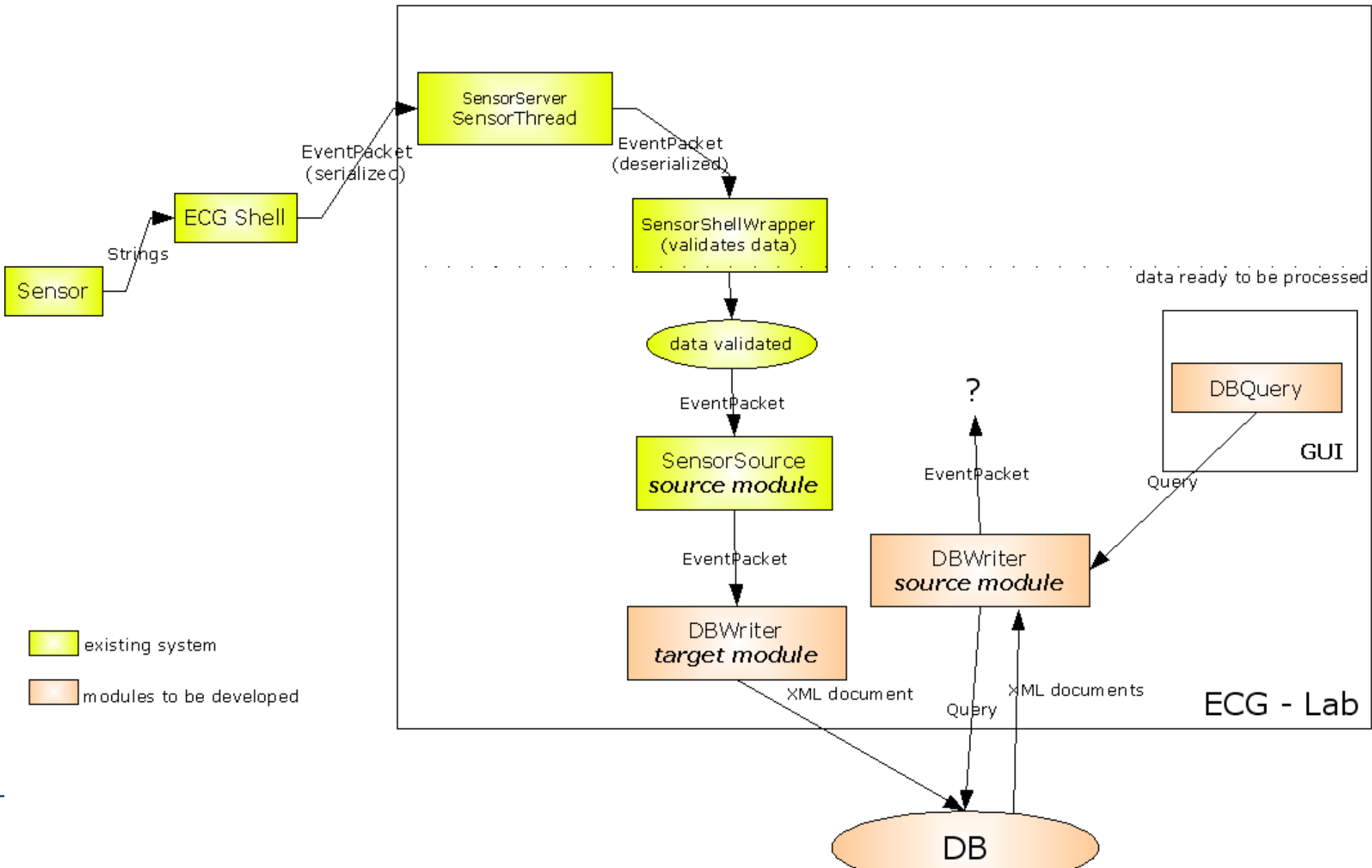
Julia Schenk

Institut für Informatik

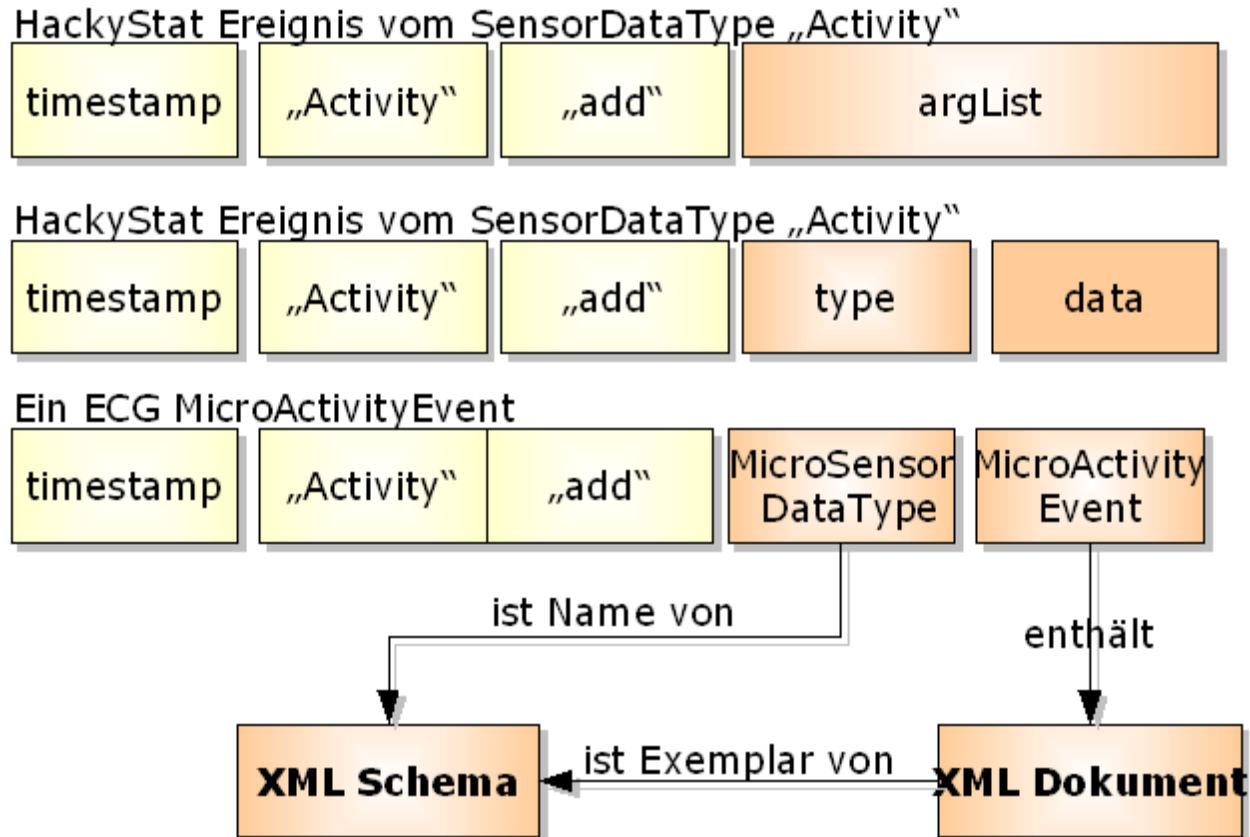
FU Berlin

09.02.2006

- Anforderungen
- ActivityEvent
- EventPacket
- Klassifizierung von XML Dokumenten
 - datenzentriert / dokumentzerntriert
- Speicherung von XML
 - native XML Datenbank: Apache Xindice (zeen-dee-chay)
 - XPath Beispielanfrage



- Auswahl eines geeigneten DBMS
 - Datenbankschema, Schlüssel, etc.
 - Anfragen über Zeitspannen
 - Speicherung zukünftiger Ideen (z.B. Episoden)
- DBWriter (Zielmodul)
 - Zerlegung
 - Abbildung EventPacket → DB
- DBReader (Quellmodul)
 - Zerlegung
 - DB Zugriff, DB Anfragen
- DBQuery
 - kein Modul im Sinne von ECG
 - interaktive Datenbankabfragen



```
Tue 10.01.2006 11:17:53 CET#Activity#;add;msdt.editor.xsd;
<?xml version="1.0"?>
<microActivity>
  <commonData>
    <username>FleißigerUser</username>
    <projectname>>null</projectname>
  </commonData>
  <editor>
    <activity>opened</activity>
    <editorname>
      TourBusRoutesView.java
    </editorname>
  </editor>
</microActivity>
```

toString()

EventPacket
-id : int
-timestamp : Date
-sensorDataType : string
-argList : List
+toString() : void
+isEqual() : bool

• datenzentriert

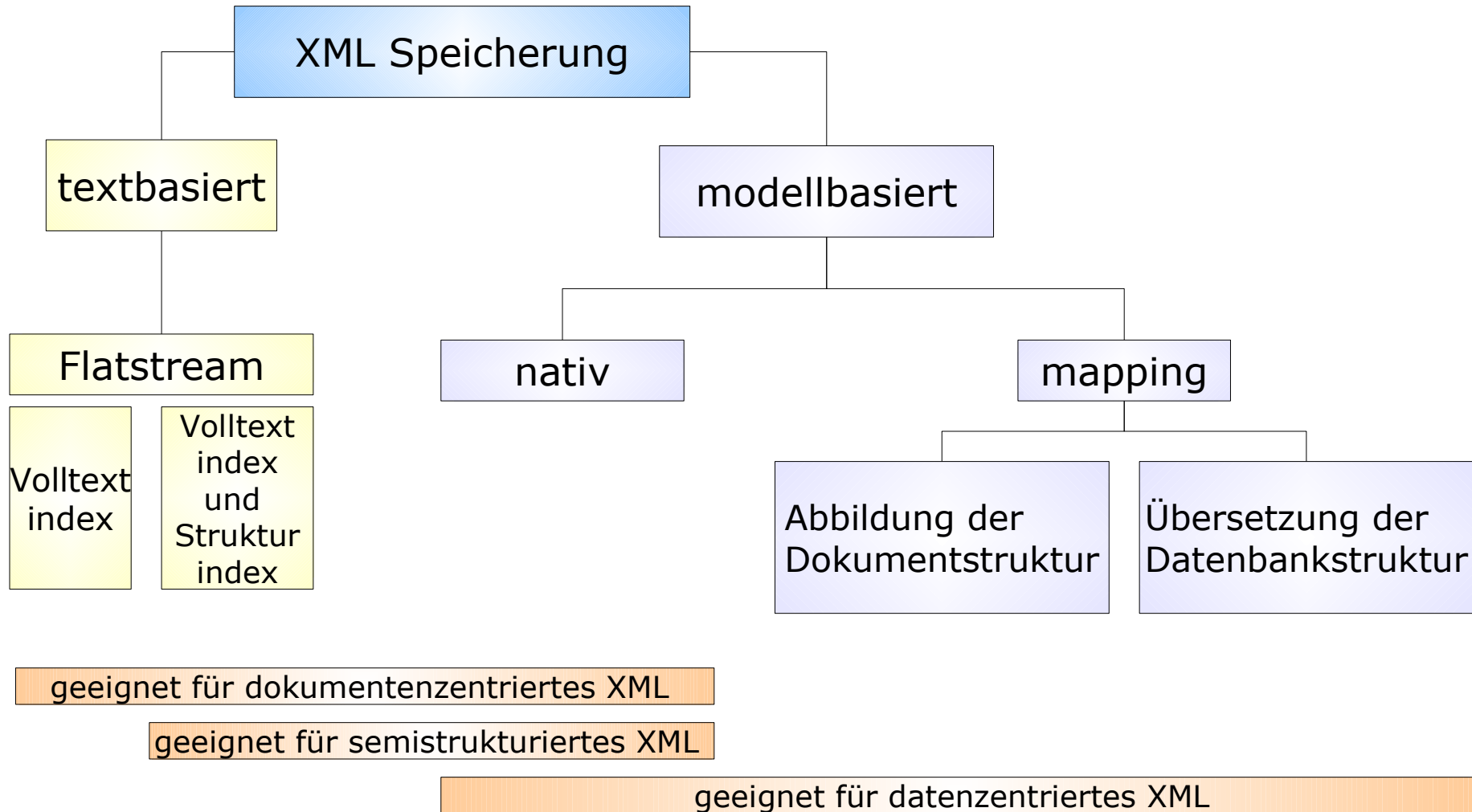
- für Maschinenlesbarkeit optimiert
- sehr regelmäßige Struktur
- beinhaltet Daten mit flachen Hierarchien
- Informationen genügen meist einem festen Schema
- Bsp: Bestellungen, Flugpläne
- Ideal für relationale oder objektorientierte Datenbanken

• dokumentenzentriert

- für menschlichen Gebrauch entworfen
- XML hat Dokumentencharakter
- weniger Struktur
- meist kein festes Schema
- „mixed content“
- Bsp: Bücher, Konfigurationsdatei
- geeignet für native XML Datenbanken

• semistrukturiert

- enthält sowohl strukturierte (datenzentrierte) als auch unstrukturierte (dokumentenzentrierte) Bereiche
- Bsp.: HTML Seiten
- geeignet für native XML Datenbanken



- Ausgelegt für die Speicherung von XML Dokumenten
- keine Abbildung zwischen unterschiedlichen Datenmodellen ("XML in – XML out")
- Features:
 - Document Collections
 - Dokumente werden in sog. Collections gespeichert
 - XPath als Anfragesprache
 - XUpdate
 - für serverseitige Änderungen an den Daten
 - XML Indexing
 - erlaubt Definition von Indizes auf Element- und Attributwerten
 - Java XML:DB API Implementierung
 - bietet Portabilität von XML Datenbankanwendungen, (ähnlich JDBC für relationale Datenbanken)

- Beispiel: /db/events Collection
 - Finde alle Ereignisse des Programmierers FleißigerUser

```
xindice xpath -c /db/events -q  
"/microActivity/commonData[username='FleißigerUser'] text()"
```

```
Tue 10.01.2006 11:17:53 CET#Activity#;add;msdt.editor.xsd;  
<?xml version="1.0"?>  
<microActivity>  
  <commonData>  
    <username>FleißigerUser</username>  
    <projectname>>null</projectname>  
  </commonData>  
  <editor> <activity>opened</activity>  
    <editorname>TourBusRoutesView.java</editorname>  
  </editor>  
</microActivity>
```

Vielen Dank!