Usage of open standards in a document-based service framework.

Fraunhofer Institut Angewandte Informationstechnik

Usage of open standards in a document-based service framework.

The EU4ALL eServices-server

Jaroslav Pullmann

Fraunhofer Institute for Applied Information Technology FIT

FIT
Fraunhofer Institut Angewandte Informationstechnik

Business case

- EU4ALL FP6 project
  - Enhancing accessibility in higher education
- eServices-server
  - Integration platform
    - Long-running processes
    - (A)synchronous interaction
  - Web Services (SOAP), web applications (REST)
  - End users
    - Different roles, types and stages of involvement

Requirements

– Ease of integration at different levels:
  - Data and operations
  - Accessible user interfaces
– Openness and extensibility
  - Standards
  - Open source software

→ Declarative, data-driven architecture

Architecture

– Cornerstones:
  - native XML DB (eXist)
  - XQuery(P) + Java: application development
  - SCXML: work-flow definition
  - XSLT 2.0
    - XML transformations
    - Work-flow rule engine
  - XForms, XUL and VXML (planned): user interfaces

SCXML: work-flow definition

– Enhancements
  - access data model (id, role, capability)
    - basic resource patterns, e.g. (re)allocation, delegation
  - custom actions (SOAP, XML manipulation etc.)

– Approach:
  - States → tasks
  - Event transitions → publicly exposed actions
    - Event payload (input) updates state's data model
    - Action visibility according to current state + access rules
  - Event-less transitions → transition upon data test

SCXML: work-flow definition

– Validity test and storage of (partial) input data

```
<transition event='setPlatform'>
  <if cond="$_eventdata/platform != ''">
    <assign expr="$_eventdata/platform"
           location="$data/root/platform"/>
  </if>
</transition>
```

– Traversal upon data validity and completeness

```
<transition cond="$data/root/platform and
 count($data/root/userId) &gt; 1" target="getResources"/>
</transition>
```

SCXML: work-flow definition

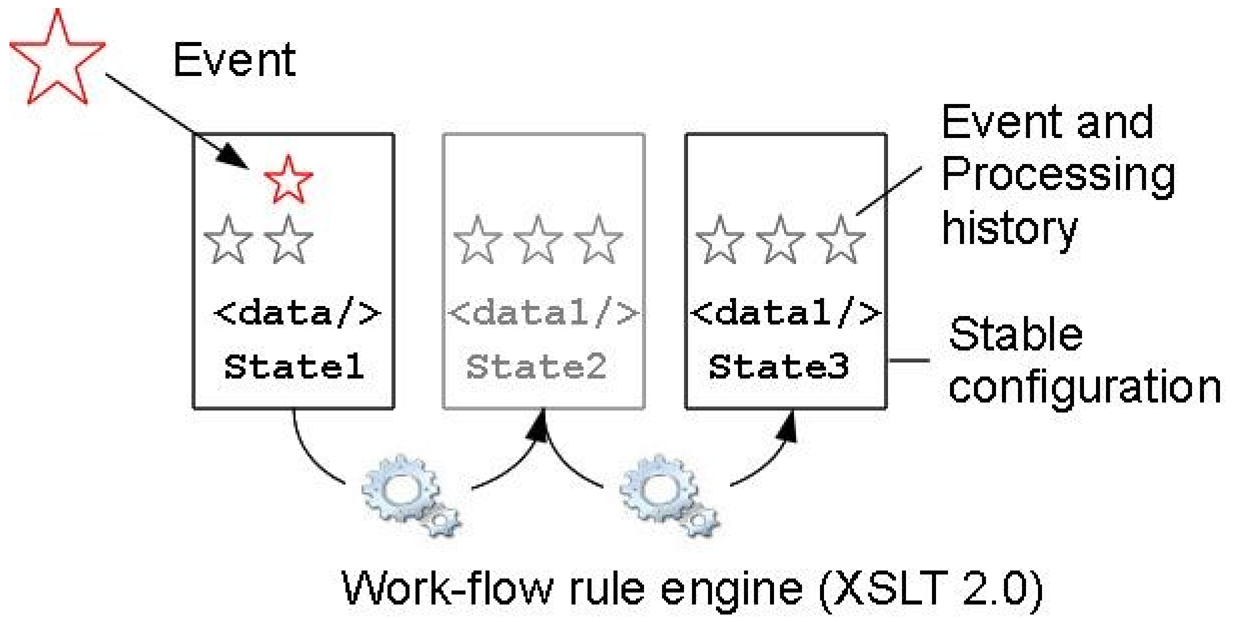
- Static model: XOR (<state>), AND (<parallel>)
  - Decomposition can't be changed at runtime
- Not suited for a data-driven flow
  - Only the first enabled transition is followed

XSLT 2.0: work-flow rule engine

– Stateless transformation (interpreter)
  - Updates and interprets a flow document
  - Dispatched event triggers the execution
  - All matching templates are executed in a pass
    - `<next-match/>`
    - Defaults to identity transformation
  - Repeated until a stable configuration is reached
    - The results of subsequent transformation do not differ

– State-full flow document
  - Processing history (events, execution count etc.)

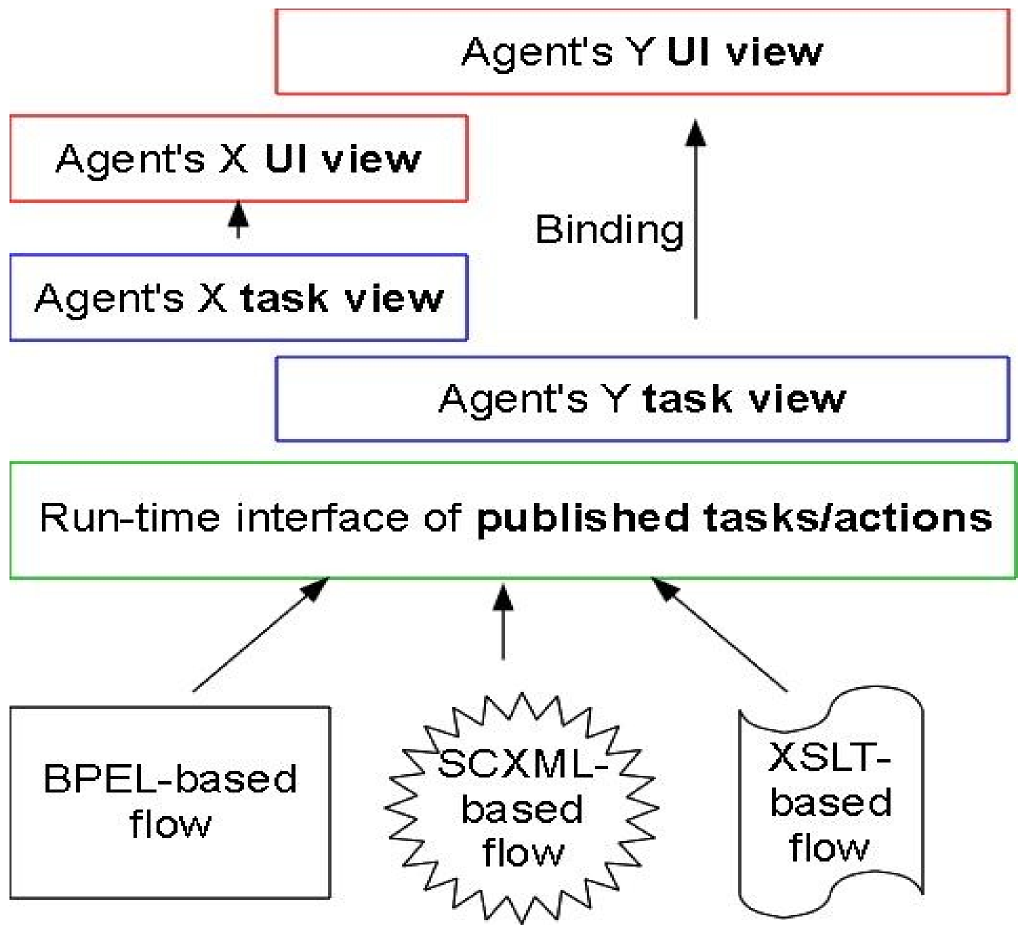
XSLT 2.0: work-flow rule engine



Task interface for data-level integration

- Unified interface abstracts from:
  - Task language in use
  - Task handler implementation
- Actions at *server*, *service*, *session*, *task* level
  - Groups separated by name-spaces
- Generated at run-time, tailored to agent
- Exposed via a selective interface
  - WSDL, WADL or UI
- Interacted via SOAP or REST

Task interface for data-level integration



Fraunhofer Institut Angewandte Informationstechnik

User interface level

– XForms
  - Rich, integrated interface
    - Model item properties (calculate, relevant, type etc.)
    - Wizards/tabbed interface (switch/case)
    - Node binding (repeat/nodeset, select/itemset)
  - Missing - synchronization
    - Triggering external events on model (notification)
      - » Will be reflected by UI via dependency graph
    - Handling data model events
      - » Deliver xforms-value-changed to model-level handlers