Adapting a Generic Data Synchronization Framework for YAWL to Access Clinical Information Systems at the Task Level

Holger Meyer\textsuperscript{1}, Sebastian Schick\textsuperscript{1}, Jan-Christian Kuhr\textsuperscript{2}
Andreas Heuer\textsuperscript{1}

\textsuperscript{1}University of Rostock, Dept. of Information Technology
\textsuperscript{2}GECKO mbH, Rostock
Evaluation in Clinical Setting

Krankenhaus Hetzelstift • Neustadt / Weinstraße • Germany
Test Case: Surgical Procedure

Procedural events

Simplified sequence of key events that each map to business process tasks

- Patient admitted
- Indication for surgery given
- Procedure registered
- Surgical procedure done
- Patient discharged
Procedural events

Events that are captured by existing clinical information systems

Patient admitted

Indication for surgery given

Procedure registered

Surgical procedure done

Patient discharged

Hospital Information Systems (HIS)
Test Case: Surgical Procedure

Procedural events
Events that are captured by YAWL

- Patient admitted
- Indication for surgery given
- Procedure registered
- Surgical procedure done
- Patient discharged

Process Execution Layer

Workflow Management System
Data synchronisation

Bidirectional data exchange needed

- Patient admitted
- Indication for surgery given
- Procedure registered
- Surgical procedure done
- Patient discharged

HIS Layer

Process Execution Layer
**Business process specification**

Enhancement of end-to-end operational process awareness and value-chain control
Use case for read-mode synchronisation

Making the patient’s admission data available to YAWL

HIS Layer

Process Execution Layer

Patient Mgt System (e.g. SAP IS-H)
Clinical data integration technology

HIS subsystems usually communicate with each other via exchange of HL7 messages that travel along a message bus.
Loose coupling of HIS and YAWL

YAWL can communicate with the HIS layer by reading data from and/or writing data to the message bus, respectively.
Data synchronisation scenario

To avoid interferences with the live HIS, the clinical test has been

- read only
- via the file system
Sample admission message

The event type ADT^A01 carries information about the patient’s master data as well as visit-related case data.

**HL7**

```
MSH|\|\|SAP-ISH|020^111|ITB111||
20110415185632|NP11I0|ADT^A01|0003807782
|P|2.1
```

**Event Type**

```
EVN|A01|20110415183500
```

**Master Data**

```
PID|||0111140419|0111266201|Doe^John|| |19
700119|M|||Sample Street
123^^Sampletown^^12345^UK||35784||D|M|02
```

**Case Data**

```
PV1|00001|I|1115.1^^|111CU|NO|||||1A||
|97^07|||NO|||11194||S|111194
2111|||20110415183500
```

**XML**

```
<masterData>
  <firstName>John</firstName>
  <lastName>Doe</lastName>
  <address>
    <city>Sampletown</city>
    <country>UK</country>
  </address>
</masterData>

<caseData>
  <admissionType>admissionType</admissionType>
  <patientId>patientId</patientId>
  <caseNo>caseNo</caseNo>
</caseData>
```
Anatomy of HL7 Messages

Extracting data from HL7 messages

Some clinical data (HL7) may be mapped to the business process level (XML)

### HL7

**MSH**
```
MSH|^~\&|SAP-ISH|020^111|ITB111||20110415185632|NP11I0|ADT^A01|0003807782|P|2.1
```

**EVN**
```
E VN|A01|20110415183500
```

**PID**
```
PID|||0111140419|0111266201|Doe^John||19700119|M||Sample Street
123^Sampletown^12345|UK||35784||D|M|02
```

**PV1**
```
PV1|00001|I|1115.1^111CU|NO|||1A|||97^07|||NO|||S|111194
2111|||20110415183500
```

### XML

```
<masterData>
  <formOfAddress>[0..1] (formOfAddressType)
  <firstName>[0..1] string
  <middleName>[0..1] string
  <lastName>[0..1] string
  <maidenName>[0..1] string
  <residence>[0..1] string
  <zipCode>[0..1] string
  <sex>[0..1] (sexType)
  <age>[0..1] (ageType)
  <nationality>[0..1] string
</masterData>

<caseData>
  <admissionType>[0..1] (admissionTypeType)
  <patientId>[0..1] (patientIdType)
  <caseNo>[0..1] (caseNoType)
  <insurance>[0..1] string
  <familyDoctor>[0..1] string
  <referringDoctor>[0..1] string
</caseData>
```
Task variables
Definition of an external task variable that is bound to HIS data
Mapping

DAF is called on enablement and on completion of a work item to facilitate mapping of data to/from external task variables.

YAWL Engine

\[ \text{def ext var } x \]

Data Gateway

Data Access Framework (DAF)

map(x)

HIS

Synchronisation Principle
Domain-specific DAF extension
Asynchronously reads clinical data from the HL7 message bus
Domain-specific DAF extension

Synchronously correlated case level parameter with clinical data

YAWL Engine

T

def var p

def var p

def ext var x

map(p)

xml

HIS

Data Access Framework (DAF)

DAF Extension

pass(p)

correlate(p, x[p])

return(x)

pass(x)
Bridging the gap between the HIS layer and YAWL

In case of the task „Admit patient“, data are generally entered into the system via the HIS.
Bridging the gap between the HIS layer and YAWL

Adapter and plug-in make up the domain-specific HL7 middleware

Introduction of a domain-specific middleware
Functional building blocks

The component „Data Extractor“ executes an XPath query to select business process relevant data from the XML representation of the HL7 message.
Separation of concerns

Domain-specific metadata are decoupled from the business process specification.
✓ Process automation allows for an **end-to-end operational control** of patient cases associated with surgical procedures

✓ The process execution engine should be **interoperable** with the hospital’s IT infrastructure

✓ Interoperability should be **bidirectional**

✓ Depending on the individual task, **either HIS or WfMS** can be the principal data-capturing system
Thank you!

jan-christian.kuhr@gecko.de

People involved

**IT Professionals**
Sebastian Schick (co-author)
Holger Meyer (co-author)
Andreas Heuer (co-author)
Markus Bandt
Viktor Baidinger
Erik Nieswandt
Thomas Becker

**Clinicians**
Dierk Vagts
Bernd Wössner
Willi Peter