

Semi-automatic generation of Web services and BPEL processes – A Model-Driven approach

Rainer Anzböck,
Schahram Dustdar

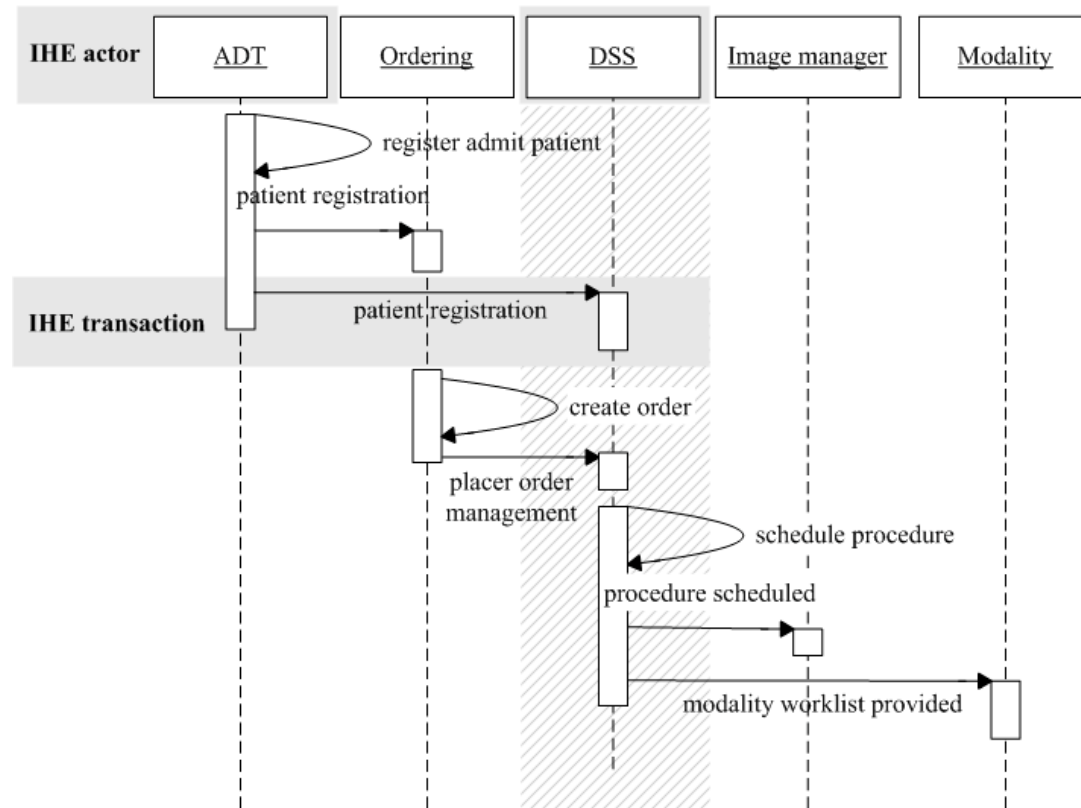
Outline

- Introduction to the domain
- Motivation
- Model-driven approach
- Modeling process steps
- Conclusion

Before we start...

- **HL7:** message based medical communication standard by HL7.org
- **DICOM:** service-oriented medical communication standard by RSNA (Radiological Society of Northamerica) (primarily used in imaging and surgery departments of hospitals)
- **IHE:** no standard, but implementation profile of HL7 and DICOM
 - kind of „best practice“ guide to the use of HL7 and DICOM
 - which messages to use, which optional fields to fill and how to combine messages for specific workflows scenarios
- **Medical Web services?** try to provide healthcare-workflows on Internet –scale (e.g., current CEN TC251 efforts)
- Why do we want this **semi-automatic generation?** -> cost effective way to generate medical WS solutions

Motivating Example – Patient registration



Hospital-wide Patient registration

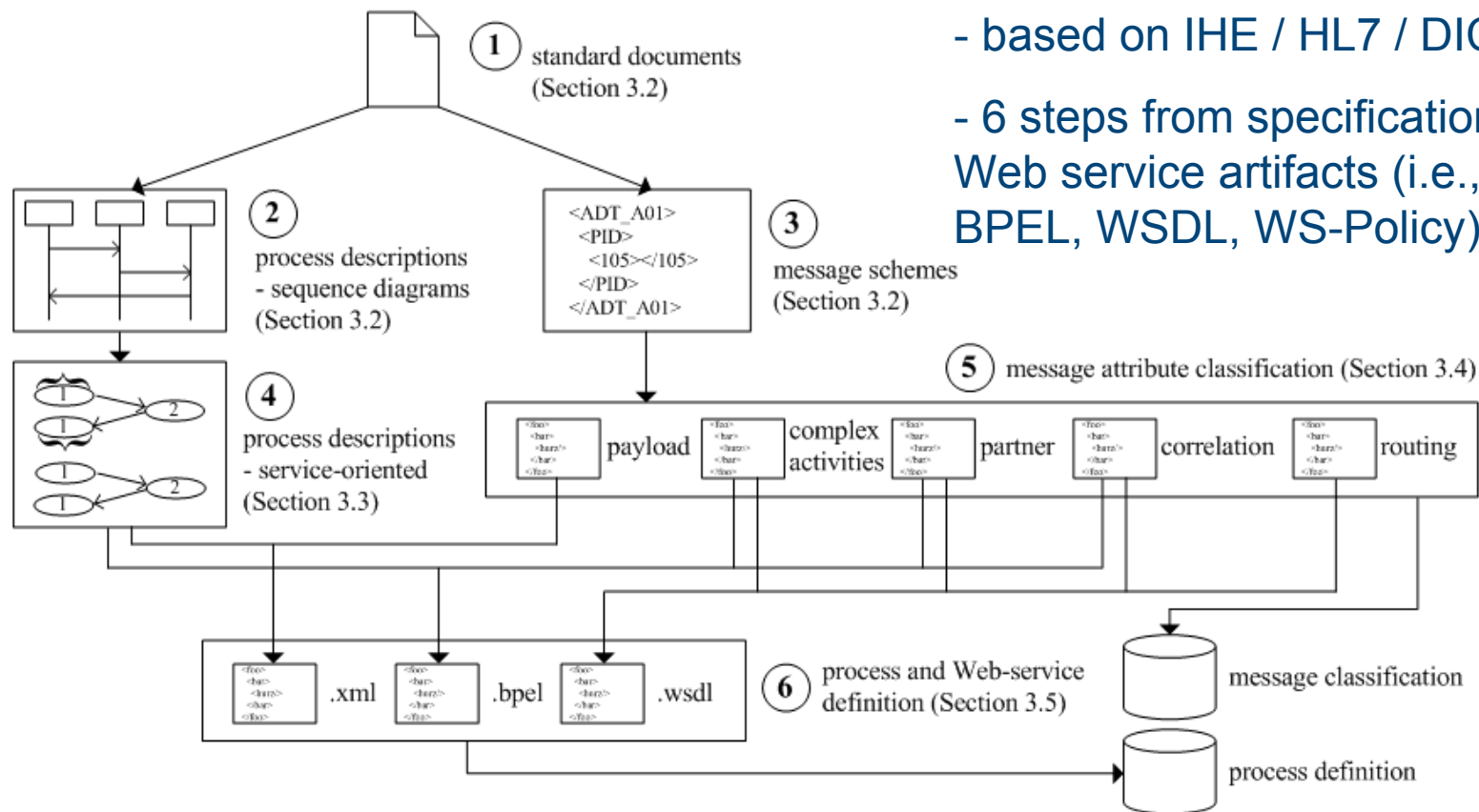
Applications:

- ADT: Patient registration
- Ordering: submitting order
- DSS: Schedules examinations for patients
- Image Manager: store/archive image data
- Modality: examine patients & generate image data

Model driven approach - Goals

- i) MD-approaches to WS development should be based on orchestration languages
- ii) define a modeling process for Web service orchestration (through (semi-)automatic transformations)
- ii) reduce complexity of specification process
- iii) support dynamically invoked Web service orchestrations
- iv) integration of security and transaction properties
- v) complement design-time process with a run-time perspective (gain better understanding of orchestration execution)
 - enable implementation of Internet-scale healthcare workflows by reducing the complexity of creating Web service specifications

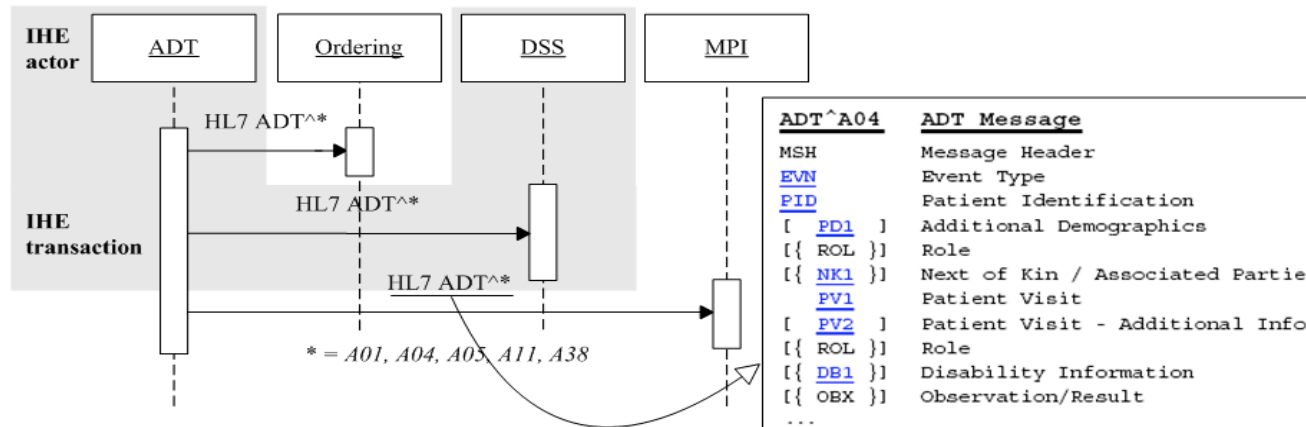
Modeling process



- based on IHE / HL7 / DICOM
- 6 steps from specification to Web service artifacts (i.e., BPEL, WSDL, WS-Policy)

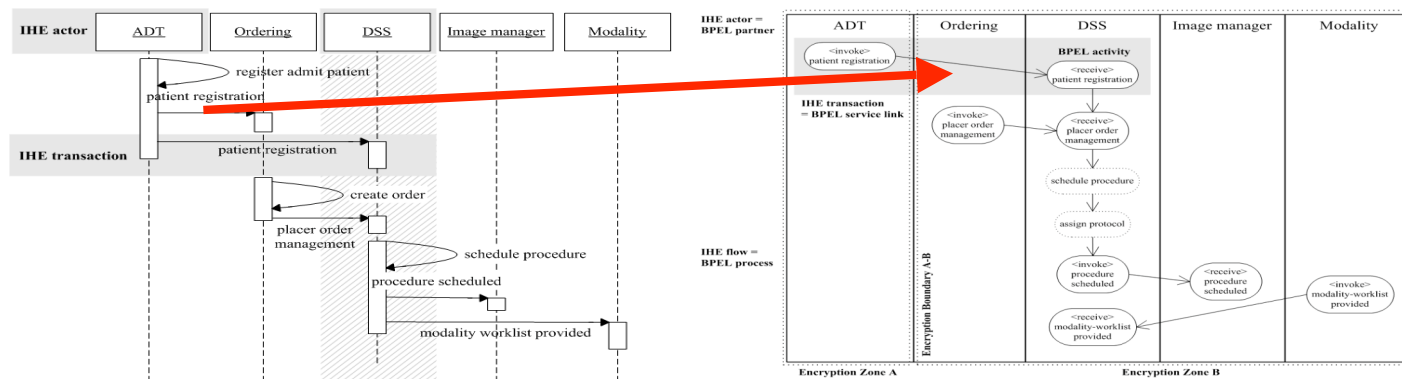
Step 1-3 – Standard documents

- create digital source representation (resolve errors in the standard document)
- extract the process from the sequence diagrams
- extract design-time and run-time properties from the messages
- map to generic XML message schema with classified attributes with HL7 messages in the payload

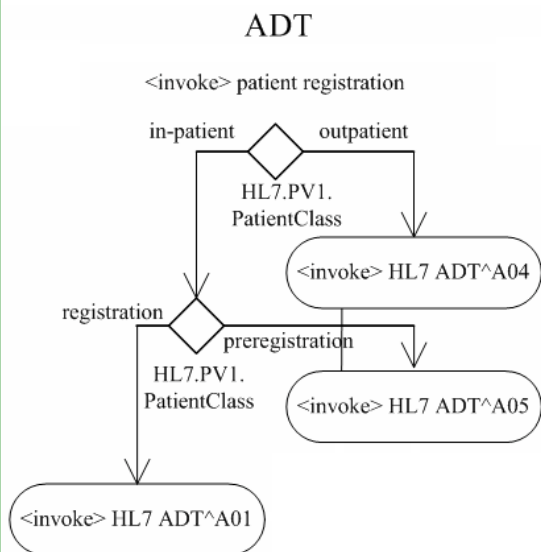


Step 4 – Process description

- select business partner for public process and skip private activities
- IHE / BPEL mappings are performed using predefined Service Ontologies (e.g., IHE workflow = BPEL process, IHE actor = BPEL partner, IHE transaction = BPEL service link)
- convert sequence diagram into activity diagrams (1:n mapping) extended with Encryption Zones and Encryption Boundaries (Zones define if/which WS-Security policies have to be enforced)
- insert security properties (WS-Security, WS-Trust, domains, trust models)

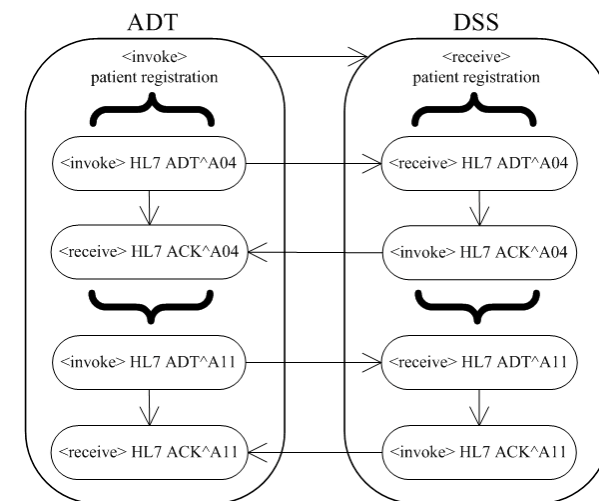


Step 4 – Process description (continued)



- extend the diagram to represent different control flows (take textual descriptions of standards)
- message attributes that control the workflow are part of the generic message format
- used control shapes have to conform to orchestration language tool

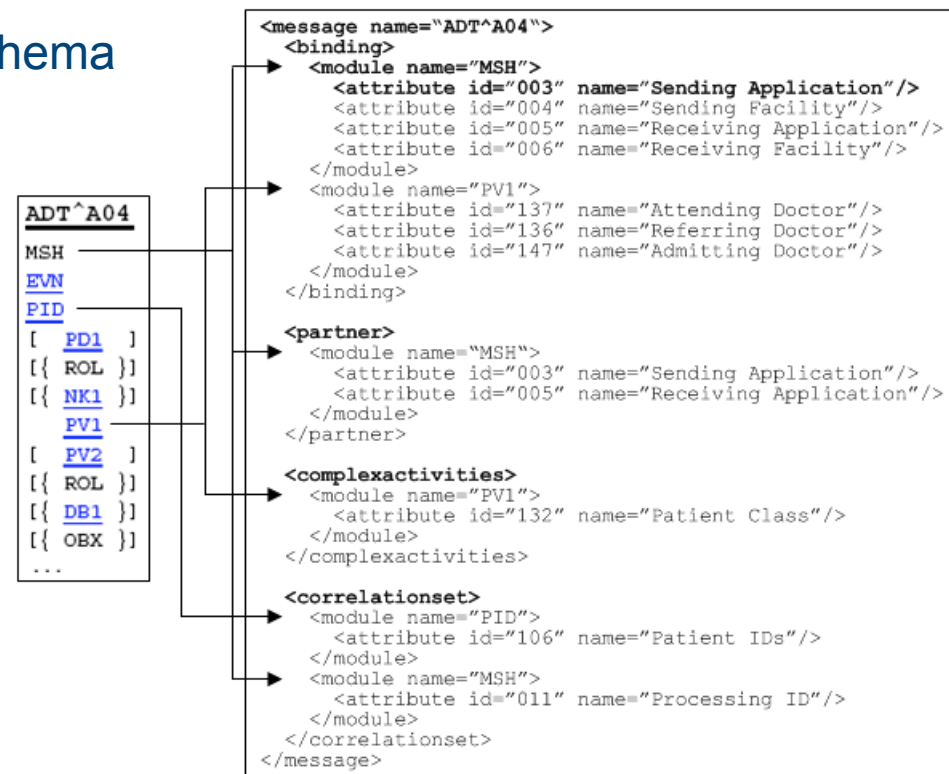
- transactions & annotations, the next level of detail
- extend diagram to represent acknowledgements
- insert transaction requirements and compensation activities



Step 5 – message classification

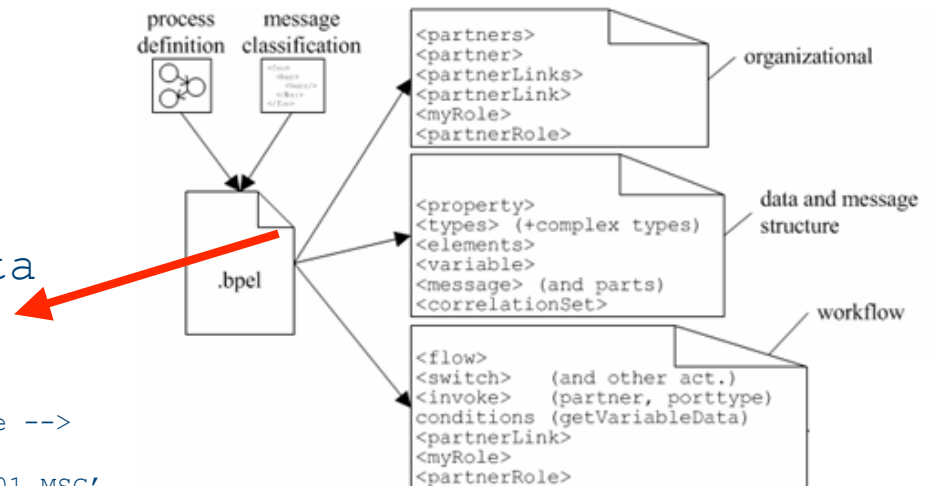
all attributes of a message schema are classified:

- binding section
- partner section
- complex activities section
- correlation set section
(-e.g., Processing ID is unique in HL7 across instances and Patient ID is not)
- ... or unimportant



Step 6 – Web service artifacts

- convert process diagram and message schema + classification to BPEL: e.g.: BPEL <flow> section
- variables used are extracted using `bpws:getVariableData`



```

<flow> <!-- patient registration flow -->
  <switch> <!-- switching PatientClass attribute -->
    <!-- in-patient -->
    <case condition= "bpws:getVariableData('HL7_A01_MSG',
'in-patient'")>
      ...
    <switch> <!-- switching PatientClass attribute -->
      <!-- registration -->
      <case condition= "bpws:getVariableData('HL7_A01_MSG', 'HL7_A01_PART', 'HL7_A01_TYPE/PV1-132') =
'registration'">
        <invoke partner="IHEActorDepartmentSystemSchedulerPartner"
          portType="IHETransactionPatientRegistrationCallbackPortType"
          operation="HL7_A01" inputvariable="HL7_A01_MSG">
          <correlations>
            <correlation set="HL7_A01_CS">
          </correlations>

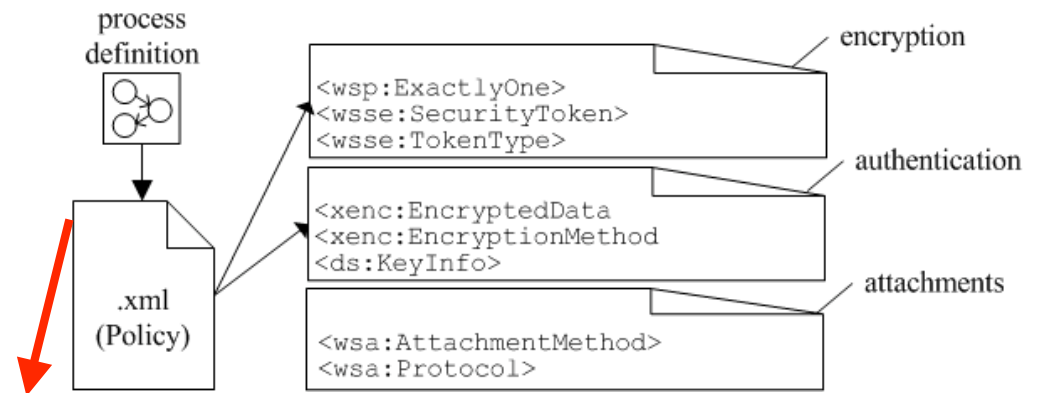
```

Step 6 – Web service artifacts (continued)

- Example: using WS-Policy and WS-SecurityPolicy for each business partner

- cover encryption and authentication requirements

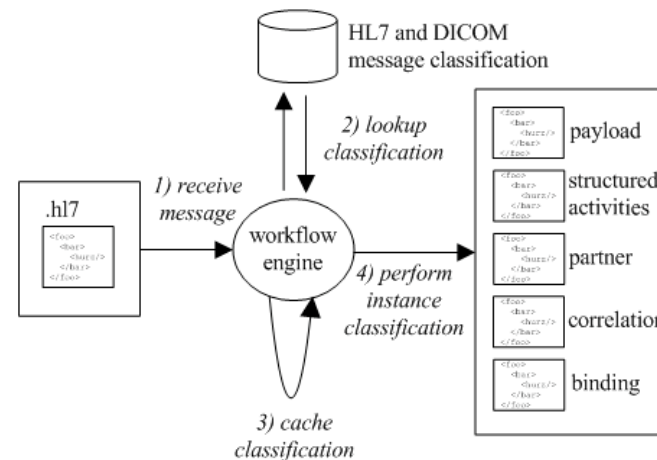
- static and dynamic configurations possible (e.g., Policy file can be filled with attribute values from message classification “partner section” slide 9)



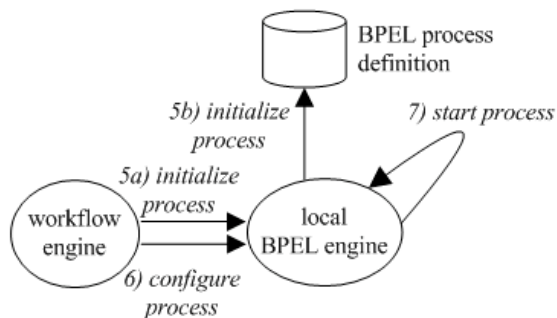
```
<wsp:Policy xmlns:wsse="..." xmlns:wsp="...">
  <wsp:ExactlyOne>
    <wsse:SecurityToken wsp:Usage="wsp:Required" wsp:Preference="1">
      <wsse:TokenType>wsse:X509v3</wsse:TokenType>
    </wsse:SecurityToken>
  </wsp:ExactlyOne>
</wsp:Policy>
<xenc:EncryptedData MimeType="Application/Dicom">
  <xenc:EncryptionMethod
Algorithm="http://www.w3.org/2000/09/xmlsig#rsa-sha1"/>
  <ds:KeyInfo>
    <ds:KeyName>CN=ADT_TEST, C=AT</ds:KeyName>
  </ds:KeyInfo>
  <xenc:CipherData>
    <xenc:CipherReference URI="dicom:image"/> ...
```

Web service process at run-time

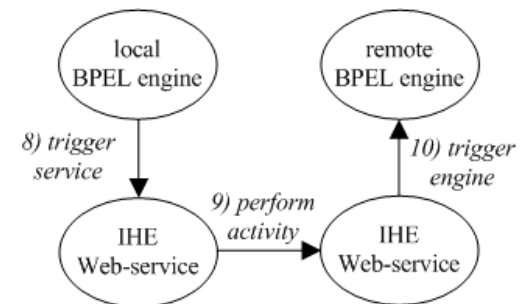
Run-time phase 1
(receiving a message):



Run-time phase 2
(start BPEL process):



Run-time phase 3
(execute BPEL process):



Conclusion

Goals reached / lessons learned?

- Evaluation: execution of workflow on two different middleware platforms (Microsoft Biztalk and Tomcat / ActiveBPEL)
- problems with WSDL compatibility -> we plan to integrate WS-I basic profile recommendations
- problems with process definition: ambiguities in standard documents, missing modeling standards for WS-Security, WS-Transactions

Future work - integrated MDA process

- tool for service ontology mapping
- UML profile for IHE workflows
- process modeling tool support
- integration of CEN domain model (EU standard 1Q/2006 enable union-wide exchange of electronic healthcare records)



Thanks for your attention