SOA approach for integration of departmental systems

Case Satakunta
Timo Itälä, 7.12.2011
Agenda

- Departmental applications, need for integration
- Business processes and business services
- Web services, identification and implementation
- Enterprise service bus
- Discussion
- Add-On: Event Driven Architecture SOA
Satakunta health district, project

- Satakunta Health District
  - Specialized care services
  - Owned by the municipalities
  - Area: 275,000 inhabitants
- SerAPI Tekes Finnwell research project
- Oct 2006 - Aug 2007
- SOLEA Tekes research project 2008 - 2011

Timo Itälä, 7.12.2011
The application map

Timo Itälä, 7.12.2011
THE RESEARCH QUESTION

Can we use Service Oriented Architecture (SOA) for integration of the departmental applications?

Reusable solution?
Functions, processes and (departmental) applications

Patient Admin:
MUSTI, Seniori

Patient Record:
WebKert, Seniori

Scheduling
Personell
Material

Reporting, BI

Patient Care Process

Resourcing Process

Management, Planning and Control Process

Outpatients
Inpatients
YKert
MLII
WebLAB
Samba
QPAti
Radu
WinRadu
TOTI
HAIKARA
Jne.

Departmental applications

Departmental applications

Departmental applications

Departmental applications

Timo Itälä, 7.12.2011
Can we use Service Oriented Architecture (SOA) for integration of the departmental applications?
The Method: Top-Down and Bottom-Up

- Care Processes
- Diagnostic services
- Generic services
- Enterprise service bus
- Applications with interfaces
THE RESULTS

SOA approach for integration of departmental systems
How to model a generic process?

- Take and store images
- Redocard findings as notes
- Print images and notes
The care process within specialized care

- Care process: tasks related to the care itself
- A generic model: all different cases are covered
- The model of the care process
  - Process consist of activities and their sequence
  - Process is triggered by an event
  - The customer of the process is the patient
  - The owner of the process: The doctor in charge
- The instance of the care process: An episode of a patient
The care process plans, executes and assesses the care
The care process is a consumer of several services
Any activity can make request for a service
The care process and services

- Service requests are made by the doctor
  - Lab tests, radiology tests, endoscopy tests etc...
  - Medication orders, therapies etc are requested services
  - Visit to outpatient clinic can be seen as a service which is requested by the receiving doctor himself
  - Inpatient episode can be seen as a service
  - A care process consists of multiple services

Timo Itälä, 7.12.2011
A care process, a service process and a task

Orders:
Referral, test orders

Assessment:
Results, Dictation

Delivery of care
Bookings, Work lists, Registration, Procedures, Recording the results, Transcription

Tasks
Capture an image, Record findings
Example: Endoscopy
Task: An examination

- Examinations:
  - Endoscopy
  - Capsule camera
  - Retinal image screening
  - Video and still imaging in operating theatre
  - Audiogram
  - Dermatology
  - EKG
  - etc...
An examination task: Endoscopy

Care Process
Referral for diagnosis

Assessment:
The doctor makes a request for an endoscopy examination

Care Service Process
Booking for the examination
Registration at the reception

Task
Capture an image, Record findings
Example: Endoscopy

Endoscopy Examination
Capture an image, Record findings

Timo Itälä, 7.12.2011
Can we use SOA for integration of the departmental applications?
A service can be a task, a sub-process or a process
Modeling web services

- A service consists of one or multiple operations
  - A Message Exchange Pattern (MEP) is related to an operation
  - **Request-Response** operation
    - A service receives a request message and sends a reply message
  - **Solicit-Response** operation
    - A service sends a request message and waits for a reply message
  - **One-way** operation
    - A service receives a message
  - **Notification** operation
    - A service sends a message
  - A fault message can be replied (Fault)
  - Synchronous - Asynchronous
Identifying web services

Service: RequestExamination
Operation: SendRequestMessage

Service: ExecuteExamination
Operations:
- ReceiveRequestMessage
- SendEndoscopyRequestMessage
- ReceiveEndoscopyResultMessage
- SendExaminationResultMessage

Service: ExecuteEndoscopyExamination
Operations:
- ReceiveEndoscopyRequestMessage
- SendEndoscopyResultMessage

Service: PatientRecord
Operations:
- ReceiveExaminationResultMessage

Timo Itälä, 7.12.2011
Description of web services

WSDL – Web Service Description Language

- Data Types
- Messages
- Services (portType) and their operations
- Binding: Transmission
- Location of services (Endpoint)

Timo Itälä, 7.12.2011
Request Examination: WSDL

Data Types
Messages
Services and operations
Binding
Location
PartnerLink
Properties of web services

- Loosely coupled - minimize dependencies
- Autonomy - encapsulation
- Abstraction - hide logic
- Reusability - generic services
- Composability - services are composed of services
- Stateless - stateful
- Synchronous - asynchronous
- Service contract - service description
- Service role: requestor or provider
Directory of examination services

- What service is used for certain examination?
- Directory of examinations is also a service
- Execute Examination -service is using directory to find out the proper examination service
- BPEL code has a service call for directory service
Implementing web services

Notification
- Request Examination
- SendRequestMessage
- ReceiveRequestMessage
- SendEndoscopyRequestMessage
- ReceiveEndoscopyRequestMessage
- SendExaminationResultMessage
- ReceiveExaminationResultMessage

Control
- Execute Examination
- Endoscopy Request Message
- Endoscopy Result Message

Action
- Execute Endoscopy Examination
- Examination Result Message

Entity
- Patient Record
- Orchestration
- Endoscopy

WSDL

BPEL Engine

Timo Itälä, 7.12.2011
BPEL (Business Process Execution Language)

BPEL orchestrates the services used by the process

- BPEL Process Model
  - Partners and roles
  - Variables
  - Fault handlers
  - Process Execution Logic
Execute Examination -service

<?xml version="1.0" encoding="UTF-8"?>
  <partnerLinks>
    <partnerLink name="Tutkimus" partnerLinkType="tns:TeeTutkimusType" myRole="TeeTutkimusServiceProvider"/>
    <partnerLink name="Tahystystutkimus" partnerLinkType="tns:TeeTahystystutkimusType" partnerRole="TeeTahystystutkimusServiceProvider" myRole="TeeTutkimusServiceProvider"/>
    <partnerLink name="Potilaskertomus" partnerLinkType="tns:PotilaskertomusType" partnerRole="PotilaskertomusServiceProvider"/>
  </partnerLinks>
  <variables>
    <variable name="input" messageType="tns:Tutkimuspyyntosanoma"/>
    <variable name="Tahystystutkimuspyynto" messageType="tns:Tahystystutkimuspyyntosanoma"/>
    <variable name="Tahystystutkimustulos" messageType="tns:Tahystystutkimustulosanoma"/>
    <variable name="output" messageType="tns:Tutkimustulosanoma"/>
  </variables>
  <faultHandlers>
    <catchAll> <sequence> <exit/> </sequence> </catchAll>
  </faultHandlers>
  <sequence name="main">
    <receive xmlns="http://docs.oasis-open.org/wsbpel/2.0/process/executable" name="receiveInput" partnerLink="Tutkimus" portType="tns:TeeTutkimus" operation="LueTutkimuspyynto" variable="input" createInstance="yes"/>
    <sequence name="TeeTahystystutkimus">
      <assign name="Muodosta_Tahystystutkimuspyynto">
        <copy>
          <from variable="input" part="Tutkimuspyyntosanoma"/>
          <to variable="Tahystystutkimuspyynto" part="Tahystystutkimuspyyntosanoma"/>
        </copy>
      </assign>
      <invoke name="TeeTahystystutkimus" partnerLink="Tahystystutkimus" portType="tns:TeeTahystystutkimus" operation="LueTahystystutkimuspyynto" inputVariable="Tahystystutkimuspyynto"/>
      <receive name="VastaanotaTahystystutkimustulos" partnerLink="Tahystystutkimus" portType="tns:TeeTahystystutkimus" operation="LahetaTahystystutkimustulos" variable="Tahystystutkimustulos"/>
      <assign name="Muodosta_Tahystystutkimuspyynto">
        <copy>
          <from variable="Tahystystutkimustulos" part="Tutkimuspyyntosanoma"/>
          <to variable="output" part="Tutkimustulosanoma"/>
        </copy>
      </assign>
      <invoke name="LahetaTutkimustulos" partnerLink="Potilaskertomus" portType="tns:Potilaskertomus" operation="LueTutkimustulos" inputVariable="Tutkimustulos"/>
    </sequence>
  </sequence>
</process>
ENTERPRISE SERVICE BUS (ESB)
All departmental applications are connected to LAN
Message broker (Ensemble)
Enterprise Service Bus (ESB)

- ESB
- Message Oriented Middleware
- Web Services
- Intelligent Routing based on Content
- XML Data transformation
- Endpoint = WSDL
  Port:Operation

Timo Itälä, 7.12.2011
Bottom-Up, Transport: Message broker

Ensemble message broker

Timo Itälä, 7.12.2011
Satakunta ESB

- Enterprise Service Bus will connect all applications together and hide the technical details
- Orchestration service will execute the processes
DISCUSSION

SOA approach for integration of departmental systems
Generic integration of departmental applications

- Service Oriented Architecture: Integrating a departmental application
  - Departmental applications are integrated using their native interfaces
  - Departmental applications are described as web services using WSDL descriptions
  - Directory of examination services is updated
  - BPEL code is updated and tested
  - GoLive

- Benefits
  - Standardization of the integration of departmental applications
  - Reusable processes are created
  - Standardization of processes
Future research

- Data model
  - ESB confirmed data, unique identifiers

- Service Directories
  - How to build, use and maintain?

- Naming standards
  - Services, operations, messages, data items

- Testing procedures
  - Services and processes
  - Faults, exceptions and recovery routines
  - Load and stress testing

- Versioning
  - Version control
  - Environment control
  - Agreements, SLAs
ADD ON

Event Driven SOA, EDA
Publish-Subscribe Model
Case Emergency Clinic

- Problem: Patients wait too long
  - Why do they have to wait?
  - What could be done about that?

- First model the process so that you understand the problem
Process and services execute in parallel

Many services can be ordered at the same time

- Laboratory tests
- Radiology examinations
- Consultations
- Operations
- etc

But how to know that the result is available?
Solution: Notification of the result

BPMN notation: The Service sends signal

But how to model receiving the signal?
Receiving the signal of the results

BPMN notation: "Non-Interrupting event subprocess". Subprocess can receive results while main process is active.
Other interested recipients of the signal

- One event can be of interest to many subprocesses
- One subprocess can be interested of many different events
How to distribute signals

- How to solve many-to-many messaging?
Event Driven Architecture (EDA)

Publish-Subscribe Model

Notation: ArchiMate
Event Driven Architecture (EDA)

Publish-Subscribe Model

Notation: ArchiMate
ESB can implement EDA

ESB can implement distribution service
Web Services described by WSDL documents

Timo Itälä, 7.12.2011
DATA MODEL
Data Model to follow the care?

Analogy:
Product Data Management
- Type = Kind of Bike (Product code)
- Instance = Individual Bike (serial number)

Process Data Management
- Type = Kind of Process (Process Type Code)
- Instance = Individual execution of Process (Process Instance ID)
QUESTIONS?

timo.itala@aalto.fi