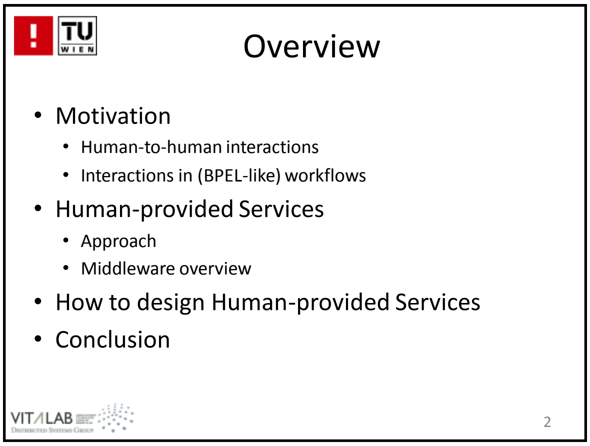
On Supporting the Design of Human-provided Services in SOA

Daniel Schall, Christoph Dorn, Hong-Linh Truong,
Schahram Dustdar

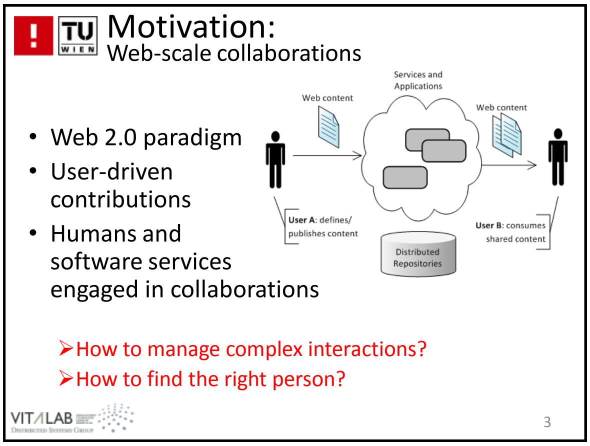
Presented by Martin Treiber

https://www.vitalab.tuwien.ac.at/autocompwiki/index.php/Human-provided_Services



Overview

- Motivation
  - Human-to-human interactions
  - Interactions in (BPEL-like) workflows
- Human-provided Services
  - Approach
  - Middleware overview
- How to design Human-provided Services
- Conclusion

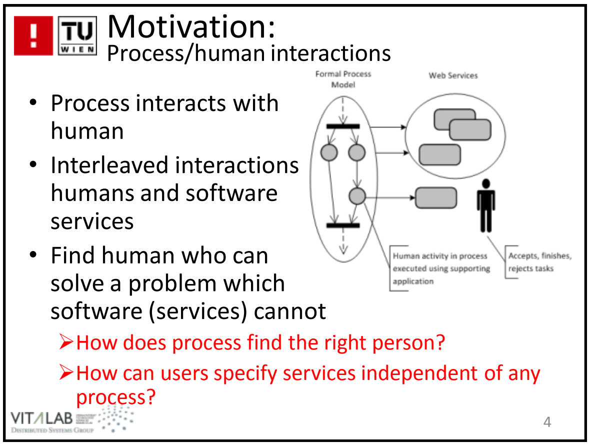


Motivation:
Web-scale collaborations

- Web 2.0 paradigm
- User-driven contributions
- Humans and software services engaged in collaborations

Services and Applications
Web content
Web content
User A: defines/ publishes content
User B: consumes shared content
Distributed Repositories

➢How to manage complex interactions?
➢How to find the right person?

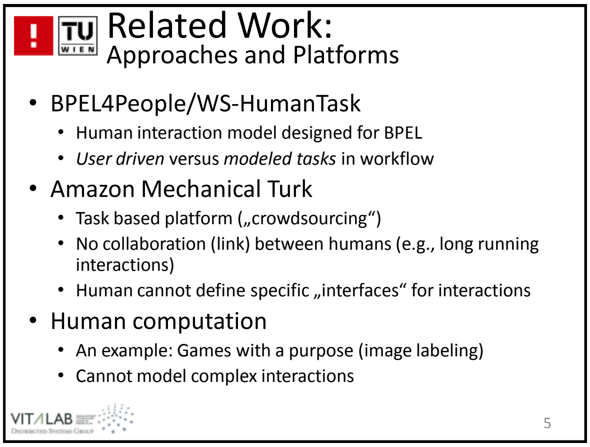


Motivation:
Process/human interactions

- Process interacts with human
- Interleaved interactions humans and software services
- Find human who can solve a problem which software (services) cannot

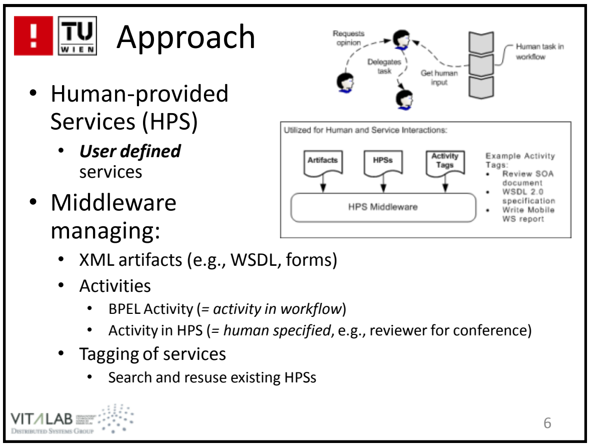
Formal Process Model
Web Services
Human activity in process executed using supporting application
Accepts, finishes, rejects tasks

➢How does process find the right person?
➢How can users specify services independent of any process?



Related Work:
Approaches and Platforms

- BPEL4People/WS-HumanTask
  - Human interaction model designed for BPEL
  - *User driven* versus *modeled tasks* in workflow
- Amazon Mechanical Turk
  - Task based platform („crowdsourcing")
  - No collaboration (link) between humans (e.g., long running interactions)
  - Human cannot define specific „interfaces" for interactions
- Human computation
  - An example: Games with a purpose (image labeling)
  - Cannot model complex interactions



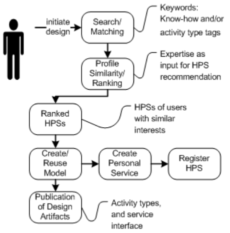
Approach

- Human-provided Services (HPS)
  - ***User defined*** services
- Middleware managing:
  - XML artifacts (e.g., WSDL, forms)
  - Activities
    - BPEL Activity (= *activity in workflow*)
    - Activity in HPS (= *human specified*, e.g., reviewer for conference)
  - Tagging of services
    - Search and resuse existing HPSs

Requests opinion
Delegates task
Get human input
Human task in workflow
Utilized for Human and Service Interactions:
Artifacts
HPSs
Activity Tags
HPS Middleware
Example Activity Tags:
• Review SOA document
• WSDL 2.0 specification
• Write Mobile WS report

Designing HPSs

- Novelty of HPS:
  Users can specify activities as (Web) services
  - E.g., consulting service
- Find existing HPSs
  Reusability
  - Search and resuse existing HPSs
  - Similarity ranking based on user profile information
- Create New HPS
  - User tools hiding underlying complexities („Mashup“ like HPS design)
  - Personal Services = User profile + activities + artifacts (WSDL)


initiate design
Search/ Matching
Keywords: Know-how and/or activity type tags
Profile Similarity/ Ranking
Expertise as input for HPS recommendation
Ranked HPSs
HPSs of users with similar interests
Create/ Reuse Model
Create Personal Service
Register HPS
Publication of Design Artifacts
Activity types, and service interface

Example

- (1) User specifies activity: „Document review“
- (2) Existing service?
  - Reuse or create new definition („Human-provided review service“)
- (3) Framework supports
  - Automatic translation into low interfaces (WSDL, XForms)
- (4) Runtime interactions
  - Through HPS access layer (= message dispatcher/router)

- XML Example:
  - <wsdl:message name="GetReview">:
    Definition of ***what*** a user (HPS) expectes to ***perform activity***
  - <wsdl:portType name="HPSReviewPortType">:
    Definition of ***how*** the activity is mapped onto an ***action***
  - <wsdl:binding name="HALSOAPBinding" type="HPSReviewPortType">:
    Technical binding of HPS to middleware access layer

HPS Discovery

- (1) Query string specified by service requester
- (2) Matching of HPS capabilities
  - Return interfaces for interactions (e.g., depending on requester WSDL or forms based representation)
  - XML Example:
  - Atom Feed referencing resources associated with HPS

News-reporter service:

```
<entry>
  <title>News Reporters</title>
  <link rel="alternate" type="application/atom+xml"
                href="/atom/newsreporter.xml" />
  <summary>News-reporter services.</summary>
</entry>
```

- (3) Ranking „best available“ HPS
  - Criteria such as expertise
  - Context dependent (e.g., location)
- (4) Interactions (runtime) enabled by middleware

Conclusion

- *Human interactions in SOA important issue!*
- Human-provided Services supporting versatile collaborations
  - Integration with BPEL4People
  - Ad-hoc collaborations
- Mashup-like tools to support the design of HPSs
  - Model driven approach based on XML standards (WSDL, XForms, etc.)
- Future work
  - How to compose complex interactions between humans and services
  - Generate HPSs based on user profiles

Thanks!

Martin Treiber (treiber@infosys.tuwien.ac.at)
Daniel Schall (schall@infosys.tuwien.ac.at)

https://www.vitalab.tuwien.ac.at/autocompwiki/index.php/Human-provided_Services