Human Interactions in Mixed Systems - Architecture, Protocols, and Algorithms

Daniel Schall
Distributed Systems Group
Institute of Information Systems
TU Wien

http://www.infosys.tuwien.ac.at
http://www.infosys.tuwien.ac.at/prototyp/HPS/HPS_index.html

27. Feb. 2009
Overview

• Motivation
• Related approaches and challenges
• The Human-Provided Services (HPS) framework
• Ranking and discovery in interaction networks
• Experiments
• Conclusion and future work
Motivation

- **Open dynamic ecosystems**
  - People and software services integrated into evolving “solutions”

- **Communications and coordination**
  - “Anytime-anywhere“ pervasive infrastructures and mobility

- **Mass collaboration**
  - Knowledge sharing and social interaction
Related Work: Approaches and platforms

- **BPEL4People/WS-HT**
  - User driven versus modeled tasks in workflow

- **Collective intelligence**
  - „Crowdsourcing“ (e.g., Amazon Mechanical Turk)
  - No collaboration link between humans

- Modeling of human interactions in dynamic service-oriented systems
- Reputation mechanism and expertise ranking in large-scale systems
Challenges (HPS)

- Current service-oriented systems only support the definition (interactions with) software services
- How to integrate human capabilities in SOA?
- How can people create services?

My Approach

- Human-Provided Services (HPS) to define human capabilities as (Web) service interfaces.
- *User-driven* approach to creating HPSs
- Ability to support interactions
• Human-Provided Services (HPS)
  – Users-define services
  – Activity modeling

• Unified view
  – Human and software service depicted using Web services standards
Architecture
Example
Designing HPSs

• **Novelty of HPS:**
  Users can specify activities as (Web) services
  • E.g., consultant or reporter

• **Find existing HPSs**
  **Reusability**
  • Search and reuse 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)
Designing HPSs

Template for activity types

User controls enable the definition of types

Output (XSD, XSI, XForm)
Overview Metrics

- Classification of Metrics

- Task
  - Claimed/Assigned Tasks
  - Completed Tasks
  - Processing Time

- Service
  - Availability
  - Request/Response Ratio
  - Responsiveness

- Interaction
  - Intensity
  - Context
  - Rating
  - Opportunistic/Long Running

- User
  - Profile Metrics (Membership)
  - Skill Level
  - Competency
Challenges (Ranking)

• How to find the most relevant expert?
• How to calculate the expertise of people in an automated manner?
• How to account for changing interests and the skill level in different fields of interest?

My Approach

• Dynamic Skill and Activity-based PageRank
• Interaction mining using link-intensity weights
• Personalization based on interaction context
• Aggregated importance using query terms
Page et al. (1999), The PageRank Citation Ranking: Bringing Order to the Web.
I will contact User 2 depending on the link weight $w_{1,2}$. The link weight is based on strength and intensities of interactions.

I will contact *some* other user. For example, to start a new collaboration by relaying a message.
Ranking Algorithm: Interaction context

• Users interact in different contexts with different intensities

  context 1 (e.g., topic = WS Addressing)  context 2 (e.g., topic = WS Policy)

  Interaction intensity context 1  Interaction intensity context 2

• Personalize ranking (i.e., expertise) for different contexts
Ranking Algorithm:

Calculate the importance of u, v, z:

\[ DSA(u) = \alpha \sum_{v \in \text{inlinks}(u)} \frac{W_{v,u}}{W_S} DSA(v) + (1 - \alpha) \sum_{w_m \in W_M} w_{m,p_m(u)} \]

Iterative algorithm to compute rankings (convergence criteria)

<table>
<thead>
<tr>
<th>Iteration</th>
<th>PR(u)</th>
<th>PR(v)</th>
<th>PR(z)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0.75</td>
<td>1.125</td>
</tr>
<tr>
<td>2</td>
<td>1.0625</td>
<td>0.765625</td>
<td>1.1484375</td>
</tr>
</tbody>
</table>
• **Approach**: Expertise mining in weighted subgraph

**Theorem linearity (Haveliwala 02):**

\[ w_1 PR(p_1) + w_2 PR(p_2) = PR(w_1 p_1 + w_2 p_2) \]
Context-dependent DSARank

- (1) Identify context of interactions ("tags")
- (2) Select relevant links and people
- (3) Create weighted subgraph (for context)
- (4) Perform mining

\[ DSA(u;C') = \sum_{c \in C'} w_c DSA(w_1 p_1(u) + \ldots + w_n p_n(u)) \]

Calculated offline
E.g., \( p(u) = w_1 \text{IIL}(u) + w_2 \text{availability}(u) \)

User 1’s expertise in context 1
User 1’s expertise in context 2

Combined online based on preferences
Results (1/2)

- Real dataset (Reality mining)
- High interaction intensity influences importance rankings
- Expected informedness of users

<table>
<thead>
<tr>
<th>ID</th>
<th>Rank (DSA)</th>
<th>Rank (PR)</th>
<th>Intensity (out)</th>
<th>Intensity (in)</th>
</tr>
</thead>
<tbody>
<tr>
<td>43</td>
<td>1</td>
<td>6</td>
<td>2.74</td>
<td>0.58</td>
</tr>
<tr>
<td>187</td>
<td>2</td>
<td>90</td>
<td>6.60</td>
<td>8.85</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>10</td>
<td>93</td>
<td>4.38</td>
<td>3.14</td>
</tr>
</tbody>
</table>
• Real dataset (Email)
• High interaction intensity reveals key people
• *Best informed* users

<table>
<thead>
<tr>
<th>ID</th>
<th>Rank (DSA)</th>
<th>Rank (PR)</th>
<th>Intensity Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>37</td>
<td>1</td>
<td>21</td>
<td>7.31</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>253</td>
<td>4</td>
<td>170</td>
<td>2.07</td>
</tr>
<tr>
<td>347</td>
<td>5</td>
<td>282</td>
<td>1.39</td>
</tr>
</tbody>
</table>
Conclusion

• Human-Provided Services supporting versatile collaborations

• Global importance ranking based on interaction intensities and context

• Future work
  • How to compose interactions between humans and services
  • Generate HPSs based on user profiles
Thanks for your attention!

Daniel Schall
Distributed Systems Group
Institute of Information Systems
TU Wien

http://www.infosys.tuwien.ac.at
http://www.infosys.tuwien.ac.at/prototyp/HPS/HPS_index.html
Backup
• To manage artifacts, resources and HPSs
• To manage collaboration structure
  • Actions and related messages
  • Activities and associated resources
  • Activities and related subactivities
Task Model

- To control the status of interactions
  - Start time, end time, progress
  - Notifications
  - Stakeholders (groups)
HPS WSDL 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)

XML Example:
- `<wsdl:message name="GetReview">:
  Definition of what a user (HPS) expects 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 Access Layer as interaction proxy
Discovery and Ranking

- (1) Logging interactions
- (2) Create interaction graph (offline)
- (3) Aggregate ranking results based on preferences (online)
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
    
    <entry>
    <title>News Reporters</title>
    <link rel="alternate" type="application/atom+xml" href="/atom/newsreporter.xml"/>
    <summary>News−reporterservices</summary>
    </entry>

• (3) Ranking „best available“ HPS
  • Criteria such as expertise
  • Context dependent (e.g., location)

• (4) Runtime interactions HPS Access Layer
  Message dispatcher/router
Ranking Algorithm: Metrics (1/2)

- Availability
  \[ \text{availability}(u) = \sum_{v \in A(u)} \sum_{\text{call} \in (u,v)} t_{\text{call}} \]

- Link Intensity
  \[ i(l) = \left( \prod_{\text{call} \in l} t_{\text{call}} \right)^{1/|l|} + \kappa \]

- Interaction Intensity
  \[ i(l; u) = i(l) |l| \left[ \sum_{l \in \text{links}(u)} i(l) \right]^{1/|l|} \]

- Skill/expertise
  \[ \text{SE}(u; c) = \sum_{v \in \text{inlinks}(u)} \frac{w_{v,u}}{w_S} \text{SE}(v; c) \]
• Interaction Intensity Level (IIL)

\[
IIL(u) = \left[ \beta^2 \left( \sum_{l \in \text{outlinks}(u)} i_{out}(l;u) \right)^2 + (2 - \beta)^2 \left( \sum_{l \in \text{inlinks}(u)} i_{in}(l;u) \right)^2 \right]^{1/2}
\]

• IIL Imbalance

\[
imb(IIL) = \sum_{l \in \text{inlinks}(u)} i_{in}(l;u) - \sum_{l \in \text{outlinks}(u)} i_{out}(l;u)
\]

• Passive involvement: \(imb(IIL) = 1\)

• Active involvement (all interactions outgoing): \(imb(IIL) = -1\)
Implementation: Interaction Mining

- Interaction Logs
  Email, telephone, etc.

- Interaction Graph
  Filter by context

- Algorithms
  Interaction mining

- Results
  - Context dependent rankings
  - Query service (online aggregation)