Socially enhanced Services Computing

Schahram Dustdar
Distributed Systems Group
TU Wien

Joint work with:
Florian Skopik, Daniel Schall, Martin Treiber, Harald Psaier,
Lukasz Juszczyk, Hong-Linh Truong
Evolution of Large-Scale & Collective Problem solving
Autonomic Nervous System
Co-Evolution

Impact on all Institutions & Society

Communication

Collaboration

Wealth
Crowdsourcing & Social Computation
Hi there, Sign In to make a comment.

**Ideas so far**

You know better than anyone else what you want from Starbucks. So tell us. What's your Starbucks Idea? Revolutionary or simple – we want to hear it. Share your ideas, tell us what you think of other people's ideas and join the discussion. We're here, and we're ready to make ideas happen. Let's get started.

**SHARE. VOTE. DISCUSS. SEE.**

**Most Recent Ideas**

- 44 Min(s) Ago  Colorful Tumblers
- 2 Hour(s) Ago  Expand Owatonna store.
- 4 Hour(s) Ago  Re starbuck Lids Plastic Toxicity
Amazon MTurk

Mechanical Turk is a marketplace for work.
We give businesses and developers access to an on-demand, scalable workforce. Workers select from thousands of tasks and work whenever it’s convenient.

112,613 HITs available. View them now.

Make Money by working on HITs

HITs - Human Intelligence Tasks - are individual tasks that you work on. Find HITs now.

As a Mechanical Turk Worker you:
- Can work from home
- Choose your own work hours
- Get paid for doing good work

Find an interesting task Work Earn money

Get Results from Mechanical Turk Workers

Ask workers to complete HITs - Human Intelligence Tasks - and get results using Mechanical Turk. Register Now

As a Mechanical Turk Requester you:
- Have access to a global, on-demand, 24 x 7 workforce
- Get thousands of HITs completed in minutes
- Pay only when you’re satisfied with the results

Fund your account Load your tasks Get results

or learn more about being a Worker
Fundamentals

- **Open** and dynamic Internet-based environment
  - Humans **and** software resources (e.g., Web services)
  - **Joining/leaving** the environment **dynamically**
  - Humans perform **activities**

- **Massive collaboration** in **Crowds, Services & Clouds**
  - Large sets of **humans** and software **resources**
  - Dynamic **compositions**
  - Distributed communication and coordination

- **Understanding the dynamics**
  - Future interactions
  - Resource selection
  - Compositions & Adaptation of actors
  - Disclosure of information
**Motivating Scenario**

Q1: How do actor **discovery** and **selection** mechanisms work?
Q2: How can actors be flexibly involved (**ranked**)?
Q3: How can interactions and service compositions become **adaptive**?

Skopik, F., Schall, D., Dustdar, S. *Trusted Interaction Patterns in Large-scale Enterprise Service Networks*. 18th International Conference on Parallel, Distributed, and Network-Based Computing. Pisa, 2010. IEEE.
Approach

- Social Services Computing
- People & Metrics
- Self-Adaptation
- Monitoring & Trust Services
- Ranking Approach & Algorithm
Human-Provided Services (HPS)

- User contributions modeled as services
  - Users define their own services
  - Reflect willingness to contribute
- Technical realization
  - Service description with WSDL (capabilities)
  - Communication via SOAP messages
- Example: Document Review Service
  - Input: document, deadline, constraints
  - Output: review comments

Schall, D., Dustdar, S., Blake, B.M. A Programming Paradigm for Integrating Human-Provided and Software-Based Web Services
IEEE Computer, July 2010

HPS – Framework & Middleware
Overview Metrics

Metrics: ranking and selection of services
Ranking Algorithm: Interaction context

- Users interact in different contexts with different intensities

context 1 (e.g., topic = ABC)  context 2 (e.g., topic = XYZ)

Interaction intensity context 1  Interaction intensity context 2

- Personalize ranking (i.e., expertise) for different contexts

Ranking Algorithm: Context-aware DSARank (Dynamic Skill Activity)

Approach: Expertise mining in weighted subgraph

- “Tags” identify the interaction context.
- Each context tag may have different weights (e.g., frequency).
- For a given context (e.g., c1) create a subgraph.
- Perform ranking based on weighted links in subgraph.

• Linearity Theorem (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

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

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
Delegation Factory/Sink

- **Factory**
  - $a$ accepts and delegates tasks frequently
  - $a$ processes few tasks and has a low task-queue

- **Sink**
  - $d$ accepts too many tasks
  - $d$ processes slow (capability vs. overload)

- **Misbehavior impact**
  - Produces unusual amounts of task delegations
  - Tasks miss their deadline
  - Leads to performance degradations of the entire network

(Mis)behavior monitoring

• Open System with varying participation
• All services use the communication infrastructure
• Interaction logging:
  – Log the exchanged messages and process their content
• Logs provide information on:
  – Task properties: id, tags, etc.
  – Type, skills, and interests of services
Similarity Service

- Cos-similarity to determine the similarity of two services’ profile vectors:
  \[ \text{sim}_{\text{profile}}(p_u, p_v) = \cos(p_u, p_v) \]

- **Trust mirroring**: “similar minded” nodes tend to trust each other more than random nodes

- **Trust teleportation**: the past trust relation \((u,w)\) “teleports” to others having similar interests.
  - Note: \(u\) and \(w\) have different profile, e.g., different roles

Misbehavior adaptation

initial state
- $b$ queue overload detected
- find alternative/similar service
- (i) 1$^{st}$ support $b$ mirroring of trust
- (ii) 2$^{nd}$ avoid $b$ teleportation of trust
Self-adaptation concepts

- feedback loop design for misbehavior healing
- MAPE loop of autonomic computing:
  - monitor interactions and queue threshold
  - analyze behavior and compare to misbehavior models
  - update behavior registry (part of knowledge)
  - plan adaptive actions
  - execute channel regulations and redirections
VieCure framework

- Interaction logging updates monitoring db and behavior registry.
- Policy Store and Similarity Service determine the adaptations
- Admin tools allow to fine-tune the framework
Some Software

HPS – Human-Provided Services
http://www.infosys.tuwien.ac.at/prototyp/HPS/HPS_index.html

VieTE – Trust Emergence Framework
http://www.infosys.tuwien.ac.at/prototyp/VieTE/VieTE_index.html
1. **Trust-based Discovery and Interactions in Mixed Service-Oriented Systems**  

2. **Modeling and Mining of Dynamic Trust in Complex Service-oriented Systems**  

3. **Programming Human and Software-Based Web Services**  

4. **Unifying Human and Software Services in Web-Scale Collaborations**  

5. **Runtime Behavior Monitoring and Self-Adaptation in Service-Oriented Systems**  
Conclusions

Socially enhanced Services Computing requires novel “programming model” (concepts, primitives) for composing collaborative HPS and SBS:

1. Delegation principle & Interaction Models
2. Social Trust & Patterns
3. Monitoring & Adaptation principles
4. Incentive & Reward structures and mechanisms
5. Dynamic Role Models
Thanks for your attention

Schahram Dustdar
Distributed Systems Group
TU Wien

Joint work with:
Florian Skopik, Daniel Schall, Martin Treiber, Harald Psaier,
Lukasz Juszczyk, Hong-Linh Truong