

Concert of Human Proficiency in Tune-up-Leaning Systems

¹N.Tulasi Radha, ²G.Vijay Kumar

^{1,2}Dept. of IT, Kaushik College of Engineering, Visakhapatnam, AP, India

Abstract

In this paper, we motivated the trend towards socio-technical systems in SOA. In such environments social implications must be handled properly. With the human user in the loop numerous concepts, including personalization, expertise involvement, drift interests, and social dynamics become of paramount importance. Therefore, we discussed related Web standards and showed ways to extend them to fit the requirements of a people-centric Web. In particular, we outlined concepts that let people offer their expertise in a service-oriented manner and covered the deployment, discovery and selection of Human-Provided Services. In the future, we aim at providing more fine-grained monitoring and adaptation strategies. An example is the translation service presented in this paper, where some language options are typically used more often, or even more successfully than others. In that case, data types could be modified to reduce the number of available language options in the WSDL interface description and to restrict input parameters. Harnessing delegation patterns that involve various participants, a complex social network perspective is established in which connections are not only maintained between one client and an avatar, but also among avatars.

Keywords

Design, Human Factors, Management, Performance

I. Introduction

Web services have paved the way for a new type of collaborative system. Services let us design collaborative systems in a modular way in a distributed environment, adhering to standard interfaces using, for example, the Web Services Description Language (WSDL) [6]. Users can create collaborative features by (re) using and composing Web services. Services already play an important role in fulfilling organizations' business objectives because process stakeholders can design, implement, and execute business processes using Web services as well as languages such as the Business Process Execution Language (BPEL). Web and are increasingly found in Web-scale collaborations. Web services are tools that users and developers can reuse in various applications by exposing well-defined interfaces and APIs. The spectrum of collaboration ranges from process centric to ad hoc collaboration models [4]. Process-centric collaboration defines process models and follows a top-down approach. The business analyst or process architect must fully understand the processes before modeling and then enacting (instantiating) them. Such models' reusability is generally high, because we can apply process models several times. However, flexibility is rather limited, because if changes occur (such as exceptions), process architects have to remodel the process. On the other hand, ad hoc collaboration (for example, situations in which people or businesses must act spontaneously and creatively) follows a bottom-up approach. It's more flexible but less reusable, because many aspects depend on the actual players (that is, humans) involved in the process. However, Web-scale collaborations demand a flexible yet reusable approach because they might involve numerous people and software services. Here, we introduce Human-Provided Services, which you can use in ad hoc or process-centric collaborations. The HPS framework helps integrate humans into service-oriented infrastructures, thus

promoting reusability and flexibility.

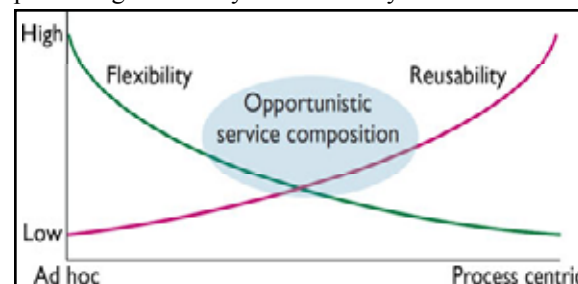


Fig. 1:

II. Existing System

Avatars are a computer user's representation of himself/herself, e.g., in form of a nickname or icon, in Internet communities. More advanced models further include interests and capabilities, such as in online gaming platforms. This makes an avatar the ideal metaphor to represent humans and their capabilities in service-oriented systems. Furthermore, an avatar does not only represent a human's services in an SOA environment, but can also actively act on behalf of the human it represents. Based on contextual constraints, such as the current load and assigned expertise areas, that software component can automatically categorize or reject requests. This process is configured through policies and rules in advance to shape the behavior of services and unburden the human from frequent but simple decisions. depicts the conceptual overview and explains our notion of avatar. HPS Layer. HPS [14], enhances the traditional SOA-based systems by enabling people to provide services with the very same technology as used by implementations of Software Based Services (SBS). Various operations for different collaborative activities indicate a provider's ability (and willingness) to participate in ad-hoc as well as process-centric collaborations. The HPS Framework provides predefined data types (XML schemas), operations, and compiled interfaces to provide particular services. The design of services is supported via a Web-based 'toolbox' (graphical user interface) enabling users to create services in a simplified manner. The creation of services does not require any knowledge related to SOA, Web services standards, or SOA runtime aspects. Based on the designed HPS, a script is parameterized to create and deploy corresponding avatars in the Genesis Hosting Environment (detailed in the following section). Composition Layer. People who provide their expertise as services on the Web, select the required features, e.g., Web service interfaces to interact with clients, and compose them, thus, predefine the capabilities of the instances managed by their avatars. While these initial decisions represent the rather static properties of an avatar, personal profiles.

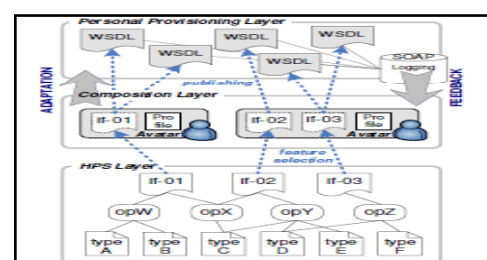


Fig. 2:

(Modeled as FOAF1) are periodically updated by our system to reflect social aspects, such as interests, interaction behavior and provided service quality. Together these static and dynamic properties characterize the avatar. The link between situation dependent profiles and composition decisions of the owner define the avatar's current providable instances. A high current load of the owner, for example, must not only update the current profile but also influence the avatars deployment strategy. Furthermore, to propagate the current situation to its instances the avatar provides instances with a connection to the current profile's state. Personal Provisioning Layer. Clients discover avatars by accounting for:

1. Functional Properties (FPs), i.e., the type of supported interfaces.
2. Non-Functional Properties (NFPs), i.e., social aspects. Here, for each single client an own service instance is deployed (peer-to-peer style). Clearly, humans providing services cannot serve thousands of concurrent requests as software services do. However, publishing dedicated instances enables our system to personalize them gradually for each individual client that has a long-term contract with the corresponding avatar.

III. Proposed Work

We discuss our approach to adaptive service provisioning by highlighting the fundamental building blocks, and in particular the adaptation of service instances itself.

The major components are organized in three layers:

A. Monitoring Layer

SOAP interactions and environment events are logged and processed. Identified composite events are triggered and forwarded to the adaptation module.

B. Infrastructure Layer

The adaptation module checks pre-defined rules to take appropriate steps, i.e., adapting the HPS templates in the HPS registry if a service does not provide sufficient QoS or adapting the deployed services in the G2 hosting environment [9], e.g., removing unused or expired operations from a service instance and its WSDL interface (as shown later).

C. User Portal

Users can discover potential services using the discovery module; and interact with particular instances through the interaction module. These interactions are logged to trigger future adaptations. Parts of this system are described by references, for instance the G2 hosting environment [9], and event triggering based on SOAP monitoring [13]. Therefore, in this paper, we revisit the interaction monitoring concept in SOA environments from a technical point of view; deal with service descriptions in terms of functional and non-functional properties to support the discovery process; and demonstrate how to enable run-time adaptations in the Genesis hosting environment.

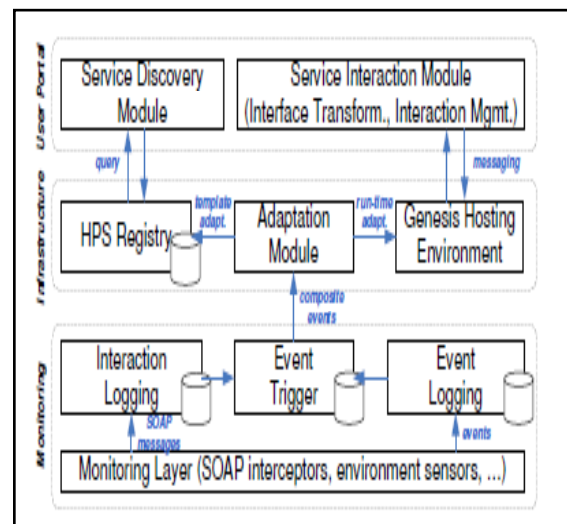


Fig. 3:

Interaction Model. Avatars are not statically bound to clients but are discovered at run-time. Thus, interactions are ad-hoc and dynamically performed with often not previously known partners. In SOA, interactions are typically modeled as SOAP messages. Besides standard SOAP structures we use various header extensions, such as WS Addressing 6, temporal properties (timestamps, deadlines), and contextual annotations. The latter are realized through tags/keywords that are assigned to messages to annotate interactions. An excerpt of a typical SOAP request is depicted.

```

1 <soap:Envelope
2   xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
5   xmlns:types="http://socsoa.infosys.tuwien.ac.at/Type"
6   xmlns:ts="http://socsoa.infosys.tuwien.ac.at/TS">
7   <soap:Header>
8     <types:timesent value="2010-09-01T15:13:21"/>
9     <types:deadline value="2010-09-06T12:00:00"/>
10    <types:context tags="Web,Services,SOA,research,paper"/>
11    <wsa:MessageID>uuid:722B1240-...</wsa:MessageID>
12    <wsa:ReplyTo>http://socsoa....</wsa:ReplyTo>
13    <wsa:From>http://socsoa....</wsa:From>
14    <wsa:To>http://socsoa....</wsa:To>
15    <wsa:Action>http://socsoa....</wsa:Action>
16  </soap:Header>
17  <soap:Body>
18    <!-- applied document translation request -->
19    <!-- schema details omitted -->
20  </soap:Body>
21 </soap:Envelope>

```

Our system utilizes temporal properties of SOAP calls to infer behavior metrics, such as the average time required to process a request, availability or responsiveness metrics (see [13, 16] for details). As demonstrated in our previous papers, metrics are calculated using the most recent history, and updated with a sliding window approach. Thus, old data ages out automatically. For the sake of simplicity, we only consider simple request-response patterns. A request can be accepted by a service and further processed by the corresponding avatar; or rejected immediately (e.g., due to the lack of free capacities). More complex, long-running interactions consisting of numerous intermediate responses are not in the scope of this paper. Dynamic Behavior Profiles. Since interests and skills of people regarding their capabilities to process requests from different domains usually widely vary,

behavior metrics are context sensitive, i.e., bound to particular expertise areas. Collections of these behavior metrics are used to calculate NFPs and finally, to calculate service capabilities. For instance, someone may be highly rewarded for providing a document translation service while his/her document review service for scientific papers is not highly ranked. Moreover, the document translation service might be successfully used for research papers in computer science, while it is not frequently used to translate business documents. Human skills and expertise evolve over time. Furthermore, interests alter and drift. Thus, our monitoring and mining approach is the key to timely compensation of behavior changes.

IV. Results

In order to implement these three algorithms on the basis on time complexity and space complexity.

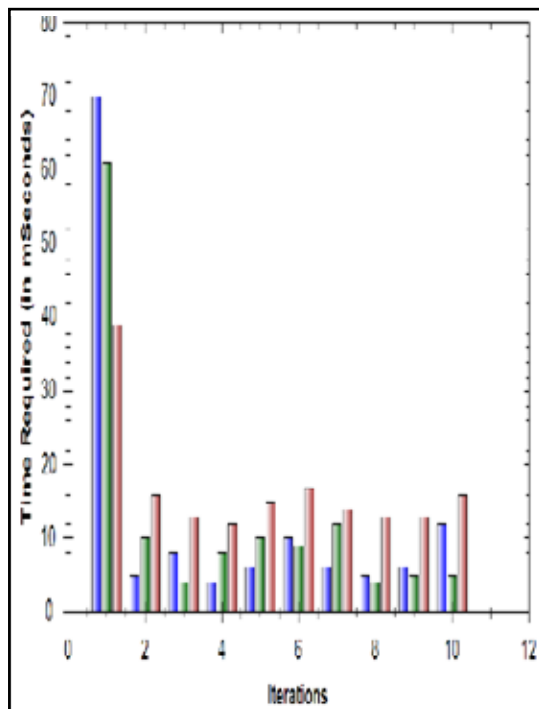


Fig. 4: Bar Chart on the Basis on Time Complexity

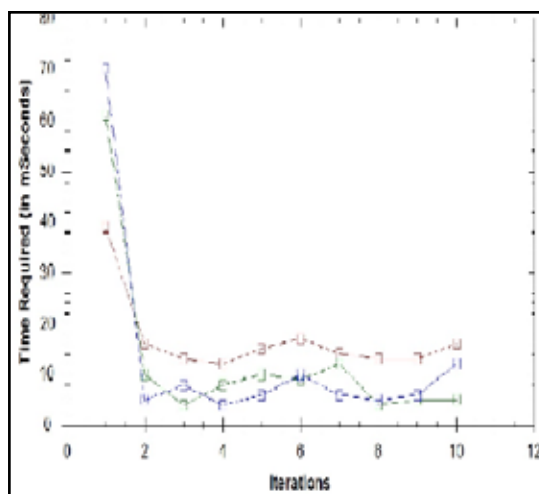


Fig. 5: Line Chart on the Basis on Time Complexity

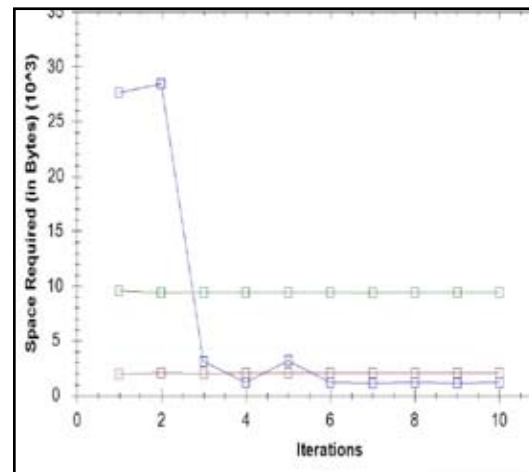


Fig. 6: Line Chart on the Basis on Space Complexity

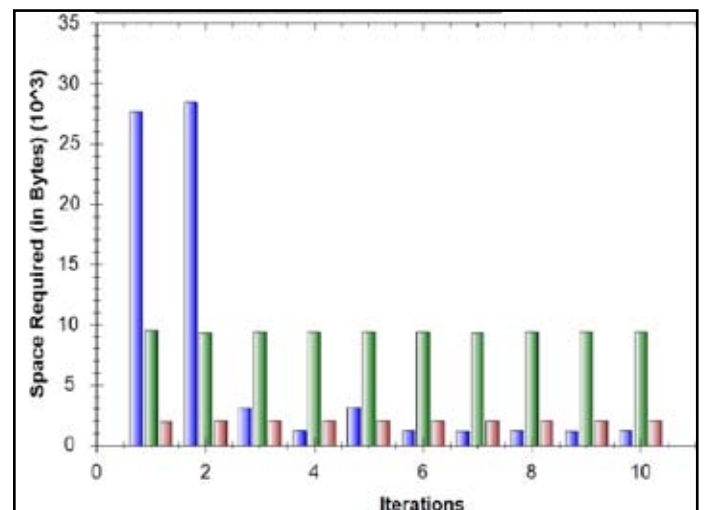


Fig. 7: Bar Chart on the Basis on Space Complexity

V. Conclusions

Web-based collaborations have become essential in today's business environments. Due to the availability of various SOA frameworks, Web services emerged as the de facto technology to realize flexible compositions of services. While most existing work focuses on the discovery and composition of software based services, we highlight concepts for a people-centric Web. Knowledge-intensive environments clearly demand for provisioning of human expertise along with sharing of computing resources or business data through software-based services. To address these challenges, we introduce an adaptive approach allowing humans to provide their expertise through services using SOA standards, such as WSDL and SOAP. The seamless integration of humans in the SOA loop triggers numerous social implications, such as evolving expertise and drifting interests of human service providers. Here we propose a framework that is based on interaction monitoring techniques enabling adaptations in SOA-based socio-technical systems

References

- [1] A. Agrawal et al., "Ws-bpel extension for people (bpe 14 people)", version 1.0., 2007.
- [2] M. Amend et al., "Web services human task (ws-human task)", version 1.0., 2007.
- [3] D. Brabham, "Crowdsourcing as a model for problem solving: An introduction and cases", *Convergence*, Vol. 14(1), pp. 75, 2008.

- [4] V. Bry, P. Giorgini, "Self-configuring socio-technical systems: Redesign at runtime", *Int'l Trans. on Syst. Science and App.*, Vol. 2(1), pp. 31–40, 2006.
- [5] A. Cherns, "The principles of sociotechnical design", *Human Relations*, Vol. 29(8), pp. 783–792, August 1976.
- [6] E. Di Nitto, C. Ghezzi, A. Metzger, M. Papazoglou, K. Pohl, "A journey to highly dynamic, self-adaptive service-based applications. *Autom. Softw. Eng.*, 2008.
- [7] S. Dobson et al., "A survey of autonomic communications", *ACM Trans. Auton. Adapt. Syst.*, Vol. 1(2), pp. 223–259, 2006.
- [8] IBM, "An architectural blueprint for autonomic computing", Whitepaper, 2005.
- [9] L. Juszczak, S. Dustdar, "Script-based generation of dynamic testbeds for soa", In *ICWS*, pp. 195–202, IEEE, 2010.
- [10] J. Kleinberg, "The convergence of social and technological networks", *Commun. ACM*, Vol. 51(11), pp. 66–72, 2008.
- [11] Object Management Group, "Uml profile for modeling quality of service and fault tolerance characteristics and mechanisms specification", Version 1.1., 2008.
- [12] H. Psai, L. Juszczak, F. Skopik, D. Schall, S. Dustdar. "Runtime behavior monitoring and self-adaptation in service-oriented systems", In *SASO*, pp. 164 – 173. IEEE, 2010.
- [13] H. Psai, F. Skopik, D. Schall, S. Dustdar, "Behavior monitoring in self-healing service-oriented systems", In *COMPSAC*, pp. 357 – 366. IEEE, 2010.
- [14] D. Schall, H.L. Truong, S. Dustdar, "Unifying human and software services in web-scale collaborations", *Internet Comp.*, Vol. 12(3), pp. 62–68, 2008.
- [15] D. Shahaf, E. Horvitz, "Generalized task markets for human and machine computation", In *AAAI*, 2010.
- [16] F. Skopik, D. Schall, S. Dustdar, "Modeling and mining of dynamic trust in complex service-oriented systems", *Information Systems*, Vol. 35, pp. 735–757, 2010.
- [17] J. Zhang, R. J. Figueiredo, "Autonomic feature selection for application classification", In *ICAC*, pp. 43–52. IEEE, 2006.



Mrs. N. Tulasi Radha, well known author and excellent teacher received M.Tech (CSE) from Jawaharlal Nehru Technological University is working as Associate Professor and HOD, Department of Information Technology, Kaushik College of Engineering and Technology, Visakhapatnam. She has 9 years of teaching experience in various engineering colleges. To her credit

couple of publications both national and international conferences/journals. Her area of Interest includes Data Warehouse and Data Mining, Information Security, Computer Networks, Operating systems and other advances in computer applications. She has been guiding five M.Tech projects.



Mr. G. Vijay Kumar is a student of Kaushik College of Engineering & Technology, Visakhapatnam. Presently he is pursuing his M.TECH (CSE) from this college and he received his graduation (CSE) from Jawaharlal Nehru Technological University.