Daniel Schall

Vienna University of Technology
Argentinierstrasse 8/184-1
1040 Wien
Austria

T +43-1-58801-18453 F +43-1-58801-18491

schall@infosys.tuwien.ac.at http://www.infosys.tuwien.ac.at/staff/dschall/

Formation and Interaction Patterns in Mixed Systems

In this document, we introduce advanced formation principles and interaction patterns with the application of these patterns in mixed systems. We apply formation patterns using the notion of Human-Provided Services (HPS) *(2008c)*. The novelty of the HPS approach is that Human-Provided Services can be discovered like software-based services. Here we introduce various concepts and patterns to realize socially-based formations and interactions in mixed systems.

Human-Provided Services

Web-based collaboration platforms have been evolving towards Web services-based architectures. In such platforms, collaborations include both humans and software services. The challenge of composing these new type of services is that interactions are highly dynamic and context-dependent. A fundamental issue is that existing collaboration platforms do not support the provisioning of *human capabilities and expertise as services*. We outline the steps to support HPS based interaction and collaboration scenarios. The HPS framework[1] *(2008b)* provides fundamental features:

- Definition of services: anyone can define his/her capabilities which are exposed as services and corresponding interfaces.
- Specification of interactions: users are able to specify their personal interaction protocol.
- Provisioning of HPSs: services can be published and provisioned in ad-hoc collaboration scenarios as well as formalized processes (e.g., BPEL).
- Discover and interact with other users/processes: by discovering services provided by humans, a user can include other HPSs in his/her processes.

HPS provides fundamental techniques for humans to express their capabilities as services and to collaborate with each other through these services. HPS is a flexible approach supporting versatile collaboration scenarios. Thus, we can utilize this concept in various (dynamic) environments.

Activity-centric Collaboration

The basic model of activity-centric interactions allows collaborations to be structured based on the concept of flexible activities. Examples of activities include creating documents or reviewing papers. An activity model describes the management aspects such as responsible and involved users, time constraints, skill requirements of involved people and applicable resources, for example services. The action concept enhances activity design-time aspects with 'runtime information'. Such runtime information includes a set of actions such as delegation, coordination and communication. The action concept provides the fundamental input for deriving various collaboration metrics.

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

Description of Formation Patterns

Basic Notation

In the following discussions and figures we denote a human or a software service as circles, lines between circles means that there is a connection between two entities, say between human *a* and human *b*. A dashed line with arrows at both ends depicts interactions, for example, with the purpose of information and context sharing.

HPS abbreviates Human-Provided Services which are denoted by a special compound symbol - document shaped symbol with embedded diamond symbol (denoting a human activity) and a user icon. A set of entities usually operate (e.g., affiliated with) in a certain scope. We denote those scopes by surrounding entities and connections with spheres. Notice, we make no assumptions how these scopes are determined or how the implementation of such scopes looks like.

Advanced Interactions

We introduce interaction concepts to model human and service interactions across various contexts. Such contexts include, for example, cross-enterprise collaborations and interactions, e.g. Virtual Organizations (VO). These interaction scenarios demand for concepts such as information sharing, flexible control, and abstraction of human capabilities as HPSs. Let us first start with a discussion on various interaction scenarios depicting the need to support context-awareness and versatile interaction scenarios. Such interaction scenarios typically span humans and (software) services.

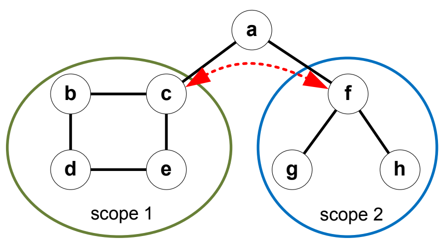

Figure 1: Broker concept connecting independent scopes.

The first concept illustrated in Figure 1 can be described as a broker. The basic principle of the broker concept is based on the idea of 'structural holes' *(2004)* and strategic formation *(2008a)* in social networks. The set of *entities b, c, d*, and *e* are connected (operate) in *scope 1*. The *broker a* controls the information and context exchange between *scope 1* and *scope 2*. This can be accomplished by interactions with entities in *scope 1* (*entity c*) *scope 2* (*entity f*).

However, in this example we assume that the broker does not necessarily attempt to inform entities in *scope 1* and *scope 2* respectively about its control of information and contexts. In some cases such separation of scopes is well desirable, but other collaboration and interactions scenarios may demand for shared context scopes.

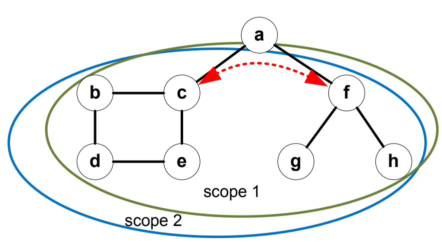

Figure 2: Broker enabling shared context views.

In Figure 2 we show a broker which merges two independent scopes with the goal of establishing a shared context between *scope 1* and *scope 2*. However, merging of contexts may not only cause conflicts, but also privacy and security concerns.

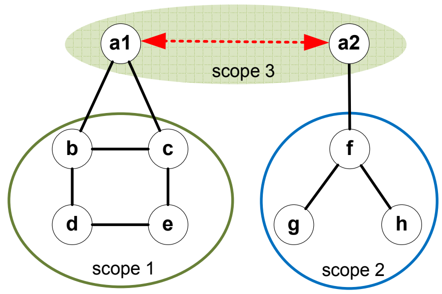

Figure 3: Shared scope between delegates.

In Figure 3 we show *scope 3* established for the purpose of syncing entities *a1* and *a2*. Such synchronizations are done on behalf of entities residing in *scope 1* and *scope 2* respectively. We call such interactions scenario 'delegates with shared, abstracted views'.

In the following in Figure 4, we introduce an interaction scenario which is more rigorous in terms of connecting entities within *scope 1* and *scope 2*.

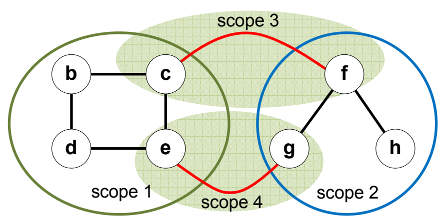

Figure 4: Mashing connections between entities.

Entities *c* and *f* as well as *e* and *g* are connected with each other, thus introducing stronger ties between both scopes. However, both connections that were introduced for the purpose of merging, for example, the ability to interact with entities in different scopes can still be restricted to operate under certain conditions (*scope 3* and *scope 4*). The problem of mashing connections could be formulated as the 'link prediction problem' *(2003)* in social networks.

HPS Support

Here we emphasize how previously introduced concepts can be realized using Human-Provided Services and context-based information sharing techniques. In Figure 5, the concepts described in Figure 1 and Figure 2 are detailed enabled through activities and services (HPSs for example).

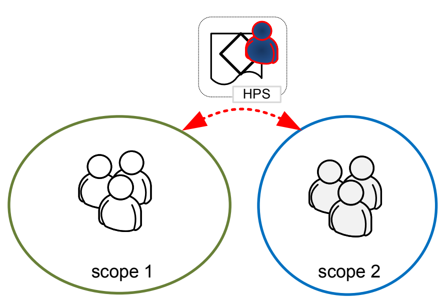

Figure 5: Supporting the HPS broker pattern (broker with separated views).

In Figure 5 we show an HPS acting as broker for two scopes. Such scopes might comprise a set of users (e.g., teams or VO). The HPS broker connects both scopes without establishing a shared context, whereas in Figure 6 we show an HPS whose goal is to connect both scopes through information-sharing techniques.

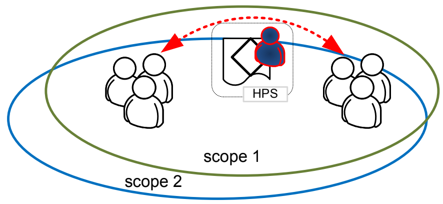

Figure 6: Supporting the HPS broker pattern (broker with shared views).

Next, the concept delegates Figure 7 with shared, abstracted views, as depicted in Figure 3. Users situated in each scope may be nominated to act as delegates (e.g., representatives) using *scope 3* to share information, context, and perform interactions. Specifically, if organizational structure as well as details regarding collaboration structure may not be exposed and shared across scopes, we favor such architectural views.

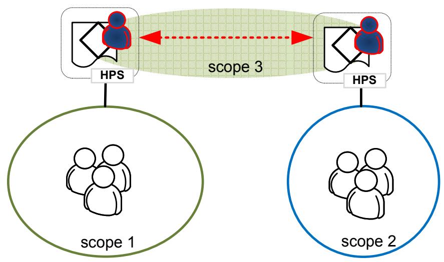

Figure 7: Supporting delegates.

Finally, based on the definition of a mashup-like scenario (i.e., Figure 8), we demonstrate the HPS support for such scenarios in Figure 4. Multiple users offering HPSs are connected with each other, therefore introducing multiple interfaces for exchange of information and context.

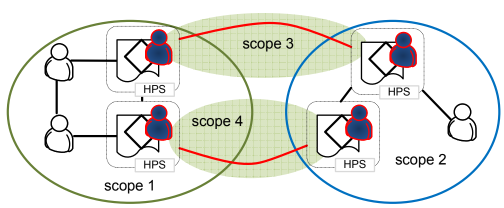

Figure 8: Mashing connections.

However, various information sharing techniques (e.g., permission to access information at certain granularity level or even document routing across scopes) help to prevent unauthorized access (e.g., delimited by *scope 3* and *scope 4*).

References

2003 Liben-Nowell D., Kleinberg J. "The Link Prediction Problem for Social Networks". *Proc. 12th International Conference on Information and Knowledge Management (CIKM)*, 2003.

2004 Burt R. S. "Structural holes and good ideas". *American Journal of Sociology*, 110(2):349-399, Sept. 2004.

2008a Kleinberg J., Suri S., Tardos E., Wexler T. "Strategic network formation with structural holes". *ACM Conference on Electronic Commerce*, 7(3):14, 2008.

2008b Schall D., Truong H.-L., Dustdar S. "The Human-Provided Services Framework". *IEEE 2008 Conference on Enterprise Computing, E-Commerce and E-Services (EEE'08)*, Crystal City, Washington, D.C., USA, July 2008.

2008c Schall D., Truong H.-L., Dustdar S. "Unifying Human and Software Services in Web-Scale Collaborations". *IEEE Internet Computing*, vol. 12, no. 3, pp. 62-68, May/Jun, 2008.

2008d Schall D., Dorn C., Dustdar S., Dadduzio I. "VieCAR - Enabling Self-adaptive Collaboration Services". *34th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA)*, 2008. IEEE.