Human Task Management for RESTful Services

Daniel Schulte
Supervised by : Bernd J. Krämer

FernUniversität in Hagen, 58084 Hagen, Germany
Daniel.Schulte@FernUni-Hagen.de

Abstract. Human task management is an integral part of WS-* based service-oriented solutions like ESBs as human interactions are important for business processes. Also alternative REST-based service-oriented solutions provide support for business processes but lack interoperable human task management solutions.

In the PhD project we are investigating possibilities and limits of incorporating human tasks in REST-based service solutions. We present a first approach and discuss the design of the remaining research process including challenges, potential benefits, and open issues.

Keywords: Human Task Management, REST, Hypermedia.

1 Introduction

Business process engines automate business processes but do also require user interactions to perform not automatable tasks [6]. Common standards like BPEL solve the process automation for web services. WS-HumanTask and BPEL4People specify additional components for user interactions. But these solutions focus on established enterprise IT environments utilizing WS-* technology stacks.

An alternative service approach relies on REST [2,8] and utilizes the web technology stack to provide its services. Whereas RESTful service composition is addressed in recent research (e. g., JOpera, see [9]), human task management for RESTful services and web applications is not examined, although many web applications already contain human tasks (e. g., conference management systems) or support their execution (e. g., online office applications). Instead, proprietary portals and e-mail communication are used regularly to notify users that tasks are available to work off. Proprietary portals for task management lead to isolated solutions and applications for e-mails lack task management facilities.

To overcome the weaknesses of these stopgap solutions, we aim to find a systematic and easy to use human task management fitting in a web context. Our research investigates axioms and constraints guiding the architecture design of web applications (providing RESTful services) that will enable users to oversee and manage their tasks independently of applications involved and their vendors.

Section 2 discusses challenges for web-scale human task management as well as current approaches and introduces our research hypothesis. Section 3 identifies main components of our proposed solution and depicts further challenges. Section 4 concludes this paper naming expected impacts.

G. Pallis et al. (Eds.): ICSOC 2011, LNCS 7221, pp. 207–212, 2012.

2 Web-Scale Human Task Management

Human tasks are —broadly defined— actions that are carried out by humans. This spans simple data entry tasks controlled by rigid input masks but also creative tasks of knowledge workers.

The management of tasks comprises their collection, the maintenance of metadata like deadlines, the view and manipulation of these metadata by users as well as scheduling of tasks, and so on. When task-aware applications provide task descriptions incl. metadata to affected humans, task management can be (partially) automated.

Challenges for an (automated) web-scale human task management are discussed in sec. 2.1, related solutions and approaches are presented in sec. 2.2, and sec. 2.3 introduces the research hypothesis underlying our approach to web-scale human task management.

2.1 Challenges for Web-Scale Human Task Management

To survey and manage tasks efficiently, users should be able to access all their tasks by a *single worklist* and have access to *task management facilities* that can handle tasks regardless of the origin. Ultimately, users need only one task management application (not one task management application per organization, (virtual) enterprise, team, task provision application, etc.). To realize this vision in the web, some requirements have to be considered [12]:

- Task announcement automation: As tasks emerge in task-aware applications, these applications should *add tasks to users worklists* incl. important metadata like deadlines directly. As tasks change over time (state, deadlines, etc.), they should also *update these metadata* automatically.
- Autonomy of task execution: Human tasks may originate in various contexts with different execution steps, states, and security requirements. They are already contained in existing applications. Therefore, the task execution should be controlled by these self-contained applications.
- Autonomy of group management: The assignment of tasks within teams may depend on current work loads, on team member roles, group policies or on some other kind of agreement. To support individual assignment patterns, independent group management solutions are needed.
- Autonomy of task management: Respecting the freedom of users means that they should be in control of their own tasks, which includes the delegation and deletion of tasks without restrictions due to arbitrary web application.
- Flexibility of tasks lifecycle and model: Task types may span routine jobs, adaptive tasks and innovative ones [3], and may be executed by one or several persons cooperatively or competitively, to name a few options. Therefore, no common task lifecycle or model can be expected. Task management has to be flexible to support different not a priori known task lifecycles and models.
- Interoperability: As arbitrary task-aware web applications should provide task descriptions, interoperability between these applications and task management applications is crucial.

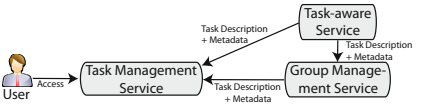

Fig. 1. Service Overview

Furthermore, web applications evolve independently, may be substituted or temporary offline. Human task management applications must handle these inherent characteristics of the web but may also benefit from its features like common authentication and authorization mechanism based on OpenID and OAuth.

Core services are depicted in fig. 1: A task management service allows a user to access and manage her tasks, task-aware services provide information about outstanding tasks and may support the task execution, and group management services enable one to consider group structures or the like for task assignment.

2.2 Approaches to Human Task Management

Existing task-aware web applications often use e-mail —a distributed, interoperable system with user-centered access possibilities— for task management although task management functions are absent. Therefore, several approaches try to extend it. Whittaker and Sidner propose to redesign e-mail for task management using threading and semantic clustering of mails and marking mails that require action together with the option to program reminders [13]. Whittaker, Bellotti & Gwizdka propose a combination of centralization (all personal information management should be done in email programs) with explicit PIM functions and information extraction [14]. Li et al. propose agents to process ingoing, outgoing, and existing e-mails for automation of manual work by intelligent applications [7] which can be adapted to extract task descriptions and metadata from e-mails to offer a worklist.

However, this approaches support, e. g., automated task description update only partial and separate task management from the web as they believe, that "it seems highly unlikely that users will abandon email for dedicated task-management tools because such tools fail to support the important reminding aspects of task management" [13]. But new browser plug-ins, notifications on smartphones and other developments improved the situation and can —as users use several devices in parallel— benefit from web managed task descriptions.

Another already mentioned approach originates in the WS-BPEL community: WS-HumanTask [1] specifies a human task concept with a detailed state and transaction system for tasks. It focuses on tasks executed by one person and is build on the WS-* stack. Questions regarding, e. g., distribution are not addressed. Task access and group management is solved proprietarily. Thus, it is appropriate for usage in enterprises with a distinct scope where IT departments manage deployment and people assignment but not applicable for a web-scale human task management.

Collaboration integration approaches like the action-centric HERMES [5] provide deep integrations at the expense of autonomy of applications (e. g., shared internal data) but do also not consider distribution and restrict the autonomy of integrated services.

Altogether, a web-scale human task management solution considering all named requirements and challenges is yet not established.

2.3 Research Hypothesis and Research Plan

As task-aware web applications rely on the web technology stack, we propose a solution that also utilizes this stack. As REST's constraints guided its development and fosters, among others, simplicity, independent evolvability, scalability and extensibility [2], it should also guided the development of a web-scale human task management solution.

Furthermore, "[t]he Web is intended to be an Internet-scale distributed hypermedia system" [2]. Hypertext links are a defining characteristic of the web and the simplicity of creating links "is partly (perhaps largely) responsible for the success of the hypertext Web as we know it today" [4]. Links allow the interconnection of information across multiple organizational boundaries and foster loose coupling and independent evolvability.

Automatically created worklists with some task metadata can link to task-aware web application, so that user can easly access them. This respects the autonomy of involved applications, and enables flexible tasks lifecycles and models as task management applications need no deep knowledge about tasks.

Nowadays, HTML and JavaScript provide not only navigable hyperlinks but also interactive controls via, e. g., HTMLs forms and JavaScipts XMLHttpRequests. Thus, task descriptions and metadata can be enriched with powerful application control information to enable even advanced task specific management facilities by hypermedia controls and code on demand.

Hence, our research hypothesis is that we can establish a web-scale human task management respecting REST constraints and making extensive use of its hypermedia as the engine of application state constraint. But to achieve interoperability and automate, e. g., the collection and update of task descriptions, some additional constraints have to be observed. Therefore, our research will

- investigate essential constraints for web-scale human task management,
- demonstrate its suitability in case studies with a prototype solution of a REST-conform human task management solution, and
- evaluate the solution using common workflow resource patterns, such as those patterns identified in [10] and already used to evaluate BPEL4People and WS-HumanTask [11].

3 Components and Challenges of a RESTful Solution

Figure 1 depicted a high level view on the task management problem domain which is further decomposed in fig. 2 and identifies the main components and resources of a RESTful solution.

A *task list* comprises all tasks of a user. As a resource, it is equipped with an URI and `POST`ing a task description to it will (a) add it to this list and (b) create a *task description* subresource. This subresource is equipped with an URI

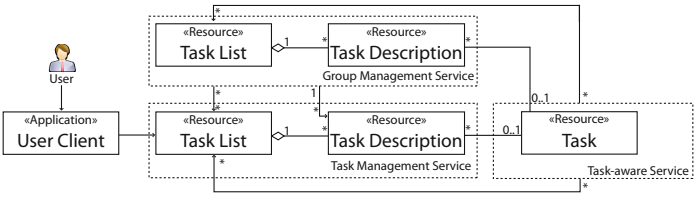

Fig. 2. Component Overview

itself which will be returned to the adding application for subsequent updates (by `PUT`ting the updated task descriptions to this subresource). Task descriptions contain metadata about tasks as well as application control information for advanced task specific management facilities.

If the execution of the task is supported by a web application, a *task* resource will be created which can be linked in the task description and invoked by a client (e. g., in a browser). And a *group management* can be realized as intermediary acting as a task management component for task-aware web applications and vice versa.

Consequently, the information flow between task management component and task-aware web application is modeled as a push architecture to reduce the workload of the task-aware web applications supporting scalability (a message exchange only occurs when an update is available), and to reduce latency of updates supporting accuracy (the update occurs as soon as an update is available).

On this technical level, further challenges need to be addressed, for instance, how to specify and describe the infrastructural parts to ensure interoperability but also independent evolvability of components. Typically, is should be based on media types and link relations to foster exploration of web applications and extensibility and should consider already established media types describing tasks as well as characteristics of the web infrastructure like caching for scalability. As the web includes malicious parties and task descriptions should only be modifiable by originators and its commissaries, they need to be secured.

Web applications which do not explicitly model tasks, like Google Docs, need to be made task-aware without changing their implementation. Other web applications which rely on e-mail to inform users about outstanding tasks need to be adapted.

A first task management component prototype is developed as Java application utilizing Jersey. An android client and a PHP based web client can access the managed tasks. A framework for using currently task-unaware web applications as task-aware ones noninvasively is under development.

4 Conclusion

Tasks of users in the web are spread over different web applications offered by independent organizations. Therefore, web-scale human task management should help users to overview and manage their tasks efficiently and independently and

has —compared to other task management scopes— some unique characteristics, in particular the distribution of task-related information and the autonomy of task-aware web applications, group management applications and task management applications. So far, no web-scale human task management is established. Our approach to this problem bases on the REST architecture style, the web technology stack and heavily relies on hypermedia.

Several challenges concerning this approach are introduced and —as the overall suitability of it— have to be studied in the future. But the expected outcomes are not restricted to the problem domain of task management as a better understanding of interoperability and evolvability may promote the interconnection of applications for other domains, too. The outcomes should further facilitate reusability and adaptability of web applications and processes in unforeseen ways.

References

1. Agrawal, A., et al.: Web Services Human Task (WS-HumanTask), Version 1.0. (2007)
2. Fielding, R.T.: Architectural Styles and the Design of Network-based Software Architectures. PhD thesis, University of California, Irvine (2000)
3. Hoffmann, F.: Aufgabe (german). In: Grochla, E. (ed.) Handwörterbuch der Organisation, Poeschel, Stuttgart, pp. 200–207 (1980)
4. Jacobs, I., Walsh, N.: Architecture of the World Wide Web, Volume One : W3C Recommendation (December 15, 2004). `http://www.w3.org/TR/2004/REC-webarch-20041215/` (August 18, 2011)
5. Kapos, G.-D., Tsalgatidou, A., Nikolaidou, M.: A Web Service-Based Platform for CSCW over Heterogeneous End-User Applications. In: PDCS 2004, pp. 462–469 (2004)
6. Kloppmann, M., et al.: WS-BPEL Extension for People – BPEL4People. IBM & SAP (2005)
7. Li, W., Zhong, N., Yao, Y., Liu, J.: An Operable Email Based Intelligent Personal Assistant. World Wide Web 12(2), 125–147 (2009)
8. Pautasso, C., Zimmermann, O., Leymann, F.: RESTful web services vs. "big" web services: making the right architectural decision. In: WWW 2008, pp. 805–814 (2008)
9. Pautasso, C.: Composing RESTful Services with JOpera. In: Bergel, A., Fabry, J. (eds.) SC 2009. LNCS, vol. 5634, pp. 142–159. Springer, Heidelberg (2009)
10. Russell, N., ter Hofstede, A.H.M., Edmond, D., van der Aalst, W.M.P.: Workflow Resource Patterns. Eindhoven University of Technology (2004)
11. Russell, N., van der Aalst, W.M.P.: Evaluation of the BPEL4People and WS-HumanTask extensions to WS-BPEL 2.0 using the workflow resource patterns. BPM Center (2007)
12. Schulte, D.: Web-Scale Human Task Management. In: Crnkovic, I., Gruhn, V., Book, M. (eds.) ECSA 2011. LNCS, vol. 6903, pp. 190–193. Springer, Heidelberg (2011)
13. Whittaker, S., Sidner, C.: Email Overload: Exploring Personal Information Management of Email. In: CHI 1996, pp. 276–283 (1996)
14. Whittaker, S., Bellotti, V., Gwizdka, J.: Email in personal information management. Communications of the ACM 49(1), 68–73 (2006)