

Mobile access to scientific event information: An Android tablet application for ginkgo

Wolfgang Reinhardt

University of Paderborn

Computer Science Education Group

Fürstenallee 11, 33102 Paderborn, Germany

wolle@upb.de

Christoph Suntrup

University of Paderborn

Computer Science Education Group

Fürstenallee 11, 33102 Paderborn, Germany

csuntrup@mail.uni-paderborn.de

Caius Cioran

University of Paderborn

Computer Science Education Group

Fürstenallee 11, 33102 Paderborn, Germany

caius@live.uni-paderborn.de

ABSTRACT

Researchers use an increasing number of social media tools and adopt practices of the Web 2.0. In the so-called Research 2.0, researchers are increasingly overwhelmed with the amount of available data and need to make educated decisions based on an analysis of the condensed information. Our previous research on awareness support for researchers led to the development of ginkgo, a social scientific event management system that combines classic features of conference management systems with features that are now well-known from many social networking applications. In this paper we report on the development of an Android tablet application for ginkgo that is driven by the researchers' requests for a more aggregated and easy-to-use, context-aware interface to the information in our system.

Author Keywords

research networks, research 2.0, API, conference management system, social interaction, social software, android, REST, mobile conference application, awareness support

INTRODUCTION

The adoption of Web 2.0 tools, technologies and practices in the context of research work is often labeled as Research 2.0, Science 2.0 or e-Science (Waldrop, 2008; Shneiderman, 2008). The application of Web 2.0 on research led (and still leads) to an increasing amount of published scientific content in conferences, journals as well as blog posts, wiki entries and tweets on early research results and comments to those outputs (cf. Rowlands and Nicholas (2005); Renear and Palmer (2009); Priem and Hemminger (2010). Priem and Hemminger (2010) point out that often too much work is being published which hinders the work of tenure and promotion committees. Renear and Palmer (2009) found out that today's researchers read 50% more papers than their 1970s fellows while spending less time on each single paper. Research is conducted in an increasingly networked manner and due to an ever-expanding amount of information created it is getting harder to filter the stream for relevant information. Recommender systems are one solution to overcome this. On the other hand, an increasing number of researchers are evermore relying on the power of their social network and active engagement in social relations to stay aware of relevant information. Thus, awareness support for researchers has both a technical and a social aspect: automatic recommender systems can make predictions of which content I might be interested in, based on the history of my own actions and the interactions with others while peers might send interesting links and objects because they know my area of interest.

The same issues exist in the context of scientific events: often researchers get to know of relevant call for papers briefly before or even after the deadline because they had no socio-technical support that would have alerted them earlier about the relevant event. In Reinhardt et al. (2011a) we have introduced the ginkgo system as one approach to make use of the very rich information that can be derived from the interactions with a conference management system and combine those information with the social power of social networking sites. In this paper we describe the derivation of requirements on a mobile application to support researchers' awareness in scientific events. Therefore, we investigate some existing mobile applications that target attendees of scientific events and list their strengths and weaknesses. From the requirements on the mobile application, we describe the development of a REST-based API for the ginkgo system (Reinhardt et al., 2011a) in order to provide data access on the mobile. Moreover, we describe the design and implementation of the ginkgo mobile tablet application for Android.

SOCIAL SCIENTIFIC EVENT MANAGEMENT WITH GINKGO

In Reinhardt et al. (2011a), we introduced *ginkgo* as a social scientific event management system that combines classic features of conference management systems (CMSs) with those known from well-known social networking sites like Facebook or Twitter. Ginkgo has been designed and developed as a novel approach to the organization and participation

in scientific events against the background of our previous studies on awareness support for researchers (Reinhardt, 2012; Reinhardt et al., 2012). The organization of scientific events is a very prestigious task that is accompanied by several tasks that have to be conducted before, during and after the event. Based on a literature study, we could identify a generic phase model of the organization of scientific events in which both organizers and attendees of the events are in need of technological support in order to fulfill their tasks (see Reinhardt et al., 2011a). The identified phases are 1) the *preparation phase* in which basic decisions regarding the name, focus, time period of the event have to be made and the call for papers has to be prepared and spread, 2) the *submission phase* in which interested authors can submit their manuscripts or other kind of contributions to the event, 3) the *review phase* in which the submissions have to be assigned to and reviewed by knowledgeable reviewers. These reviewers should not be biased towards the submission under review in order to produce an objective evaluation of the submission. In phase 4) the *selection phase*, the program committee has to select the submissions that will be selected for presentation at the event. In phase 5) the *session planning phase*, the overall schedule for the event will be planned. Typically the schedule has multiple streams and the accepted talks are assigned to one of those streams. The program committee has to consider time and room capacity constraints and to continuously update the session planning because of change requests by the presenters and organizers. The actual *event* is phase 6 of our generic phase model and followed by phase 7) the *post-processing* or *community awareness phase*.

The persons involved in the above-described phases possess different roles and interact with a wide range of other persons with same or different roles. The social interactions that take place in the context of scientific events are manifold but yet poorly supported by existing CMSs. Thus, many researchers rely on external social networking services like Facebook, Twitter or Crowdvine¹ to enhance the social experience during the whole timespan of the event (see Reinhardt et al., 2011a). In ginkgo, we support social interactions between organizers and potential attendees of an event through different means: researchers can start following each other in order to stay aware of the latest news from the other; they can engage in private chats and mark each other in their status updates. The same is possible for scientific events: researchers can simply follow an event in order to stay up-to-date about the latest news and changes and they can more particular indicate that they plan to attend the event which serves as indicator for others that they might be able to see the researcher at the respective event. If a researcher agrees to act as co-organizer or member of the program committee of an event, their activity stream will be updated accordingly, serving as a trigger for interaction with their friend and followers that might be interested in more detailed information about the event.

Since its debut, ginkgo has been used in a number of small scientific events (see <http://ginkgo.cs.upb.de/events>), which allowed us to evaluate the tool as such and to improve some of its functionality. The tests also show the potential of ginkgo together with indications for future enhancements of the tool. One reoccurring theme is that attendees of scientific events wish for location-aware mobile applications that interface with ginkgo and that they could use during the event in order to stay up-to-date of relevant information (Reinhardt, 2012; Reinhardt et al., 2011b). In the following section we review and compare three mobile applications that have been explicitly developed for the context of scientific events and point out their strengths and weaknesses. From this analysis we derived some feature requirements for a mobile application for ginkgo.

MOBILE APPLICATIONS FOR SCIENTIFIC EVENTS

In order to get the information given by a Conference Management System during an event the participant needs a mobile device and an Internet connection. This data can either be shown inside a browser or through a native application for the mobile operating system, which provides a better experience. In the following, we give an overview of existing mobile CMSs Apps and their features and points out why none of them satisfies all needs of participants.

As a participant I want to stay in contact with others and be informed about events. So the features can be divided into two major groups, namely Social Networking and Event Awareness. Social Networking contains features like getting information about each user by accessing their personal profile and contact them via personal messages or every follower by posting a status update. Event Awareness consists of being informed before, during and after an event took place. A user usually wants to browse through multiple events and their topics to decide if they are interesting for him. During an event he wants to stay informed about any changes and the presentations he would like to visit. In case of the user's wish to gather more information about speeches he should have the possibility to read the related papers and rate them.

Contrasting juxtaposition of applications

The *I-KNOW Conference Assistant*² was written exclusively for the i-Know 2011 conference³ and is available for both iOS and Android smartphones (see Figures 1 and 2). The application provides detailed information about the conference sessions and talks but it obviously does not contain any information about other events. Since the application lacks a personalization feature, the supported social networking features are also very minimalistic and can only be realized

¹ <http://crowdvine.com/>

² <http://itunes.apple.com/app/i-know-conference-assistant/id387869871?mt=8>

³ <http://i-know.tugraz.at/>

using external applications. The I-Know Conference Assistant features a well designed and seamlessly integrated note-taking feature that allows saving notes for each talk and that can be synced to a desktop application. Moreover, the application contributes to the fact that an increasing number of researchers use Twitter in the context of scientific events by assigning a dedicated hashtag to each and any talk in the overall schedule. Finally, the application contains a map view that shows the locations of all relevant event-related places and thus may help the user to find its way to the conference dinner or the main conference hall.

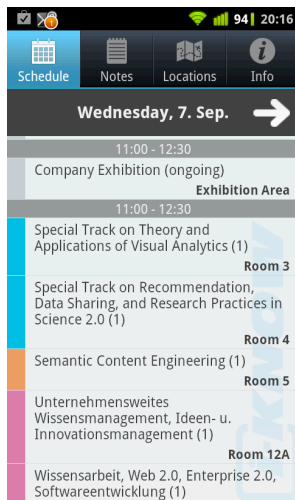


Figure 1 Schedule view in the I-KNOW Conference Navigator



Figure 2 Map view in the I-KNOW Conference Navigator

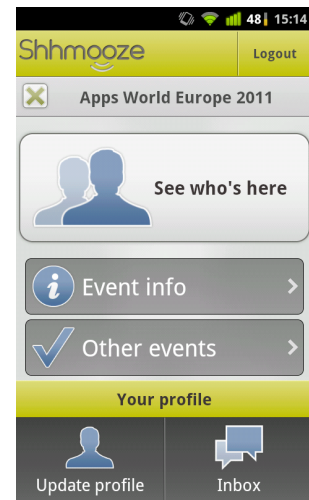


Figure 3 Event overview in Shhmooze

A different approach is used in *Shhmooze*⁴. Each user needs a personal account to use the provided services like browsing through registered events and seeing who attends (see Figure 3). In order to communicate with others and to see who is visiting the specific conference, the users have to be checked in at the same event. Thus, Shhmooze does not provide potential attendees of the event with an easy-to-use overview of who plans to be attending the event. Moreover, since there is no “friend functionality” provided by Shhmooze, there is no way to stay in contact with other researchers after having met at an event. In order to do so, the researchers have to rely on traditional means of communication. However, the application provides a function to register new events along with some basic information like date, location and topic. The information is sent to Shhmooze’s back office via email and processed there. Detailed information about sessions and talks at the event cannot be included.

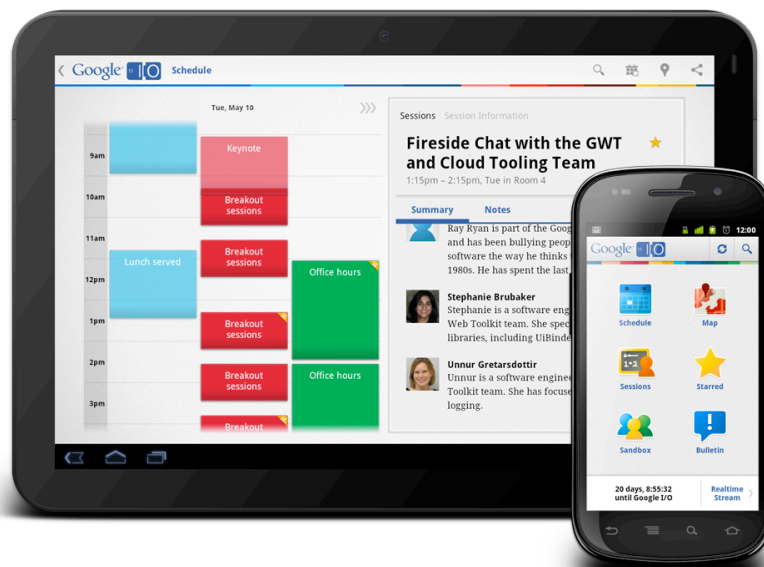


Figure 4 Google’s iosched on a tablet and smartphone

The third application that we investigated is Google’s *iosched*⁵ application that has been developed exclusively for the Google I/O developer conference. iosched offers a well-designed and easy-to-use overview of every session and the talks

⁴ <http://itunes.apple.com/app/shhmooze/id400523342?mt=8>

⁵ <http://code.google.com/p/iosched/>

in each of them. The application features a unique map application that shows which session takes place on which floor and in which room of the conference building. Moreover, it integrates a social media stream as a central element of the main interface. Finally, the application lacks a social network layer, which presumably will be added to an updated version for the 2012 I/O conference. The complete code of the application has been made publicly available by Google and can be used for own projects.

Recapitulatory comparison

After having investigated the above three mobile applications for scientific conferences, we have to conclude that so far there is no application that combines a detailed access to the information about the conference and a social network to further support information diffusion between researchers. Moreover, two of the applications are built for one dedicated conference only and thus cannot be used for multiple events at all. The three applications also have strengths that we do not want to gloss over such as the check-in feature of Shhmooze, the great note-taking feature and the dedicated Twitter hashtags of the I-KNOW Conference Assistant or the indoor maps and the beautifully designed schedule of Google's iosched. Table 1 provides a side-by-side comparison of the three applications with the developed ginkgo mobile.

	Shhmooze	I-KNOW CA	iosched	ginkgo mobile
System Features				
Support of multiple events	✓	✗	✗	✓
Express interest in an event	✓	✗	✗	✓
Browse through registered events	✓	✗	✗	✓
See where events take place on a world map	✗	✗	✗	✓
View the event schedule	✗	✓	✓	○
Take notes	✗	✓	✗	○
Social network features				
Create a user profile	✓	✗	✗	✓
Write status updates	✗	✗	✗	✓
Read activity streams	✗	✗	✗	✓
Write private messages	✓	✗	✗	✓
Add friends	✗	✗	✗	✓
Browse through followers	✗	✗	✗	✓
Event management features				
Create an event	✓	✗	✗	✗
Create reviews	✗	✗	✗	✓
Create an individual schedule	✗	✗	✗	✓
Download papers for talks	✗	✗	✗	○
✓ = feature is available ✗ = feature is not available ○ = feature is in preparation				

Table 1. Comparison between mobile CMSs applications.

Requirements for ginkgo mobile

From the analysis of the above mobile applications and in line with the features of the ginkgo web application (see Reinhardt et al, 2011a), we defined the following requirements for the first version of ginkgo mobile. First, users have to be able to login to the application. Since the web application allows to login via Twitter or using username and password, the same should be possible on the mobile application. Users should be able to access user and event profiles in order to browse through the respective activity stream, to follow other users and indicate their interest in scientific events by following them or indicating their plan to attend the event. When a user comes to the event venue, he should be enabled to check-in at the event, so that the other researches are informed about his arrival. The check-in feature is a direct enabler of social interactions amongst event attendees. Moreover, users of ginkgo mobile should be able to send private messages with other users to support their social interactions. Since social media coverage of scientific events has become commonplace today, users of ginkgo mobile should be enabled to send status updates and directly forward them

to Twitter. Finally, we choose to not enable users to create new events directly on the mobile application; the process requires much structured information such as the call for papers, deadlines and the invitation of co-organizers and reviewers, which is not easily to support in a tablet application.

DEVELOPING AN API FOR GINKGO

In order to realize the native mobile application described above and to enable an easy integration and propagation of data, it is essential to offer access to the ginkgo core system via an Application Programming Interface (API). An API is advantageous for both parties: API consumers and API providers. On the one hand API consumers get the possibility to process data independently of the provider's user interface whereas on the other hand the API provider gains popularity and traffic.

Choosing an appropriate architecture

There are several possibilities and styles to develop a web API. According to one of the biggest directories for Web 2.0 APIs (ProgrammableWeb, 2012) the architectural style that is used the most often for Web based services is Representational State Transfer (REST) (Fielding, 2000). ProgrammableWeb (2012) contains around 6.000 APIs whereof 70% use REST and 21% use SOAP.

REST became a de facto standard for web APIs because it simplifies processing heterogeneous data from different sources in a homogenous way. Every piece of information a REST API publishes is modeled as a resource, must be uniquely identified with a Uniform Resource Locator (URL) and supports the same set of operations (the Hypertext Transfer Protocol (HTTP) methods). These principles have a far-reaching effect on interoperability and data exchange (Battle & Benson, 2008). For example, on a video portal a video is not *commented* (verb) but rather a comment is *created* (noun). Because a comment is modeled as a resource (differently to SOAP, for example, where this would be an action) it is possible to reuse it in a completely different context. Thereby API consumers can choose the representation format, which fits best to their requirements (e.g., JSON, XML or HTML). Figure 5 schematically compares common interfaces with REST interfaces.

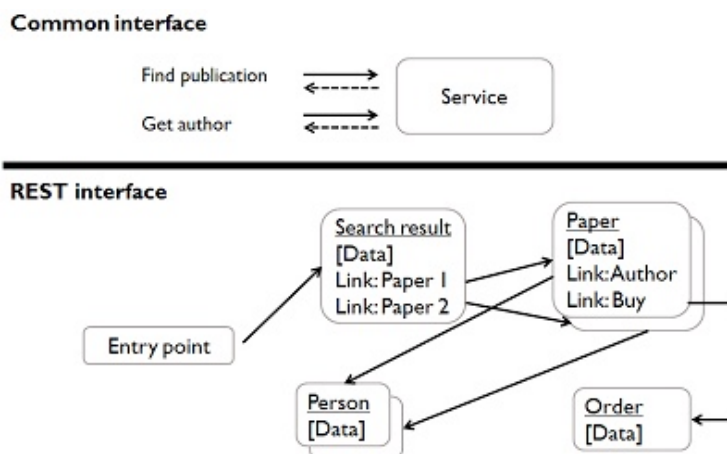


Figure 5 Comparison of common and REST interfaces

A common interface offers access to data via several functions (“Find publication”, “Get author”). Communication with a REST interface always starts by retrieving the representation of the entry point resource, which includes *hyperlinks* indicating the next possible actions for the API consumer (comparable with *transitions* in state machines). In Figure 5 we choose to search for a publication and receive the representation of a search result resource, which in turn contains hyperlinks to all of the found papers. With this architecture, exchanging particular pieces of data offered by the interface is simple, as every resource has a fixed URL. Furthermore, the integration of external data in our own service is as simple as referencing this data via a hyperlink in one of our representations. Furthermore, a performance evaluation of RESTful web services and SOAP based web services especially for mobile devices in Hamad, Saad and Abed (2010) reveals advantage of REST over SOAP. As a main result of the evaluation, the message size of RESTful web services is smaller than the size of SOAP messages. Additionally less response time per request leads to less power consumption of mobile devices and a better user experience.

The advantages mentioned above turn REST into the appropriate architectural style to develop an API for ginkgo.

Core concepts of the ginkgo API

As starting point of the API development a data model, source code and the running ginkgo web application are available. To develop the resources and representations needed, we roughly follow the steps Richardson (2007) proposes:

1. Derive resources from relevant parts of the data model of the ginkgo core system.

2. Add additional resources to support all desired functionalities.
3. Name the resources with URLs.
4. Choose the HTTP methods every particular resource should support.
5. Design the representation(s) for every particular resource.
6. Create connections between resources by adding hyperlinks to their representations.

Media type and versioning

The media type (also called mime type) constitutes the basic contract between client and server and defines how they should communicate. The description of a media type must contain the semantics of all elements within a representation, which belongs to this media type. To give a meaning to hyperlinks we use *rel* and *href* attributes like defined in Nottingham (2010):

```

{"rel": "self", "href": "https://api.ginkgosem.com/users/3jfe" }
{"rel": "https://api.ginkgosem.com/rels/collection/comment",
 "href": "https://api.ginkgosem.com/users/3jfe/comment" }

```

The three lines above show two links in JSON syntax. The link relations *self* and *https://api.ginkgosem.com/rels/collection/comment* identify the semantics of the corresponding target resource in the *href* attribute. *self* is globally standardized (see <http://www.iana.org/assignments/link-relations>) and indicates that the target contains the URL of the resource in the current context. The second link relation is ginkgo-specific, therefore has to be in the form of an URL and indicates that the corresponding link points to a list of comments.

Having implemented these semantics and following the rules HTTP defines, an API consumer navigates via hyperlinks through the API until he reaches the representation of a resource, which includes the desired information.

The ginkgo API uses the media type for versioning, too. Our media type (*application/vnd.ginkgosem-v1+json*) contains a version number, which will be incremented for upcoming new versions. Corresponding to the content negotiation mechanism in HTTP, an API consumer always requests and accordingly receives the version he understands. Versioning through the media type is to be preferred to including a version indicator in the URL of a resource, because a URL identifies a resource, independent of the version of its representation (Masse, 2011).

Security

In terms of API security, attention should be paid to authentication, authorization and request limiting. OAuth 2.0 (Hammer, Recordon & Hardt, 2012) is the emerging method for authentication in the field of web APIs. Using it for the ginkgo API, it enables a registered ginkgo user to access ginkgo via mobile applications or any mashup services without giving his or her credentials to third-party applications. All third-party applications must register with ginkgo once and receive a unique client ID. With this client ID and the accordance of a ginkgo user a third-party application can request an access token for this particular user from the API. This access token empowers to access the ginkgo API on the user's behalf.

In ginkgo, the client ID and the access token are also used to realize authorization and request limiting. As there is public information in ginkgo like Call for Papers or activity streams of events, there has to be a possibility to access this information without any user authentication. Therefore the ginkgo API allows requests, which only include the client ID for authentication ("public requests"). For every request it is checked if the current resource is not user specific and thus can be retrieved. Additionally a counter is incremented (and reset after an adjustable period of time) for every request from a specific access token and client ID to prevent that any API consumer overloads the API.

DESIGN AND IMPLEMENTATION OF GINKGO MOBILE

After implementing a REST-based API for ginkgo, the system offered all necessary functionality to realize the required features described above. In this section we report on the design and implementation of ginkgo mobile, an Android tablet application that serves as mobile interface to the social features and content available in ginkgo.

Design

ginkgo mobile's user interface follows the one of Google's iosched application (see Figure 6). After a successful login the main screen will be split into two parts. The left side of the screen (number 1 to 7) contains an array of buttons that – following the dashboard design pattern – stay visible and can be used during the whole working with the app. The right side of the screen (number 8 and 9) is a container for the different views that will be shown. Initially, the container includes the personal activity stream (8), which serves as personalized information stream and is composed of news items that individuals and events that the researcher follows have shared. The same area is later used to show user and event profiles, a list of events and follower as well as user and event activity streams. Touching the user profile icon (2) will bring the user to his own profile where he can see statistics about his social connections, post status updates and see his personalized news stream. The messages icon (3) leads the user to an overview of his private chats with other users of the

ginkgo tool. By pressing the event button (4), the user is presented a list of events that are organized using ginkgo. For this list he can navigate to the event profile pages where he can start following the event or can indicate that he plans to attend the event. Moreover, the event profile shows information about the call for papers and a news stream for the event. The event profiles are also reachable by pressing the check-in button (6). The main difference is that in this case the user is presented a map view where he can explore events that are close to his current GPS position. The single events are marked with pins on the world map and a researcher can choose to check-in at the selected event or to see the event profile. The followers and friends of the logged-in user can be explored after pressing the button 5. Some selected settings can be changed in the dialog that opens after pressing the settings button (7). Finally, the action bar on the upper area of the interface contains links to the main dashboard (1) and a shortcut to posting status messages (9).



Figure 6 Main elements of the ginkgo mobile interface (on Android 3)

Implementation

ginkgo mobile is currently developed using the Android 3 software development kit (SDK) and thus runs on all the latest Android tablets. During the development of ginkgo mobile, Google released the Android 4 SDK and we plan to port our code base to Android 4 in the near future. Since Google released the source code of iosched as Open Source, we originally planned to reuse large parts of the code for our own implementation. However, during the development process we stumbled upon several oddities of the code that made it harder to reuse the existing code rather than implementing it completely from scratch. We did, however, reuse the interface design of iosched, as it was well designed to have the dashboard like navigation on the left and enough room for exchangeable content on the right.

In Android, different UI states are created using so-called activities that have their own controllers and might have separate layout as well. The navigation through those activities is realized using an activity stack that enables the user to go back and forward in history of screens. The login is handled by a special activity that takes care of the OAuth 2.0 procedure (Hammer, Recordon & Hardt, 2012). Therefore, a web view is started where the user can provide his username and password and the generated access token is stored persistently and securely on the device. Upon successful authentication, the dashboard activity is started. This interface contains two so-called fragments and the action bar. The left fragment contains the navigation buttons as described above; the right fragment is the container for the context-dependent content. The communication with the ginkgo REST API is realized using helper classes that take care of networking and the deserialization of JSON data. The single activities have access to the helper classes and can make use of the provided functions. Moreover, the helper classes implement a quality of service layer in that they make sure HTTP errors are handled and calls to the API are repeated in case they fail.

CONCLUSION AND OUTLOOK

Starting from the far-reaching changes that came with the adoption of Web 2.0 tool, technologies and practices in research, in this paper we have made the case for enhanced management of scientific events. We outlined how adding a social layer to conference management systems may not only enhance awareness of the entailed researchers but also

serves as a rich source of information for recommender systems. With ginkgo we introduced our approach for a social scientific event management system that integrates classic conference management systems with well-known features from social networking sites. Our previous research showed that researchers wish for a handier tool for awareness support in the context of scientific events, which should be tailored to mobile use. In this paper we have performed a competitor analysis of three mobile applications for conference attendees' support and could derive requirements for the implementation of a tablet application of ginkgo. Our comparison shows that the three applications under investigation are all lacking the support for social networking of (potential) attendees and are mostly not prepared to support multiple events at the same time.

Based on the requirements of ginkgo mobile we developed a REST API for ginkgo that comes with authorization via OAuth 2.0 and an intelligent versioning concept that makes use of vendor-specific media types. Thus, each API consumer can request the data provided by the API in a version that it understands. This way, the application infrastructure does not break by updating the API to a new version. The ginkgo API provides data in JSON format, which have to be deserialized on the client and thus offers a broad compatibility with most programming languages.

We have described the visual design of ginkgo mobile that is based on Google's iosched application and provided insight to the basic implementation of ginkgo mobile. The application has yet to be evaluated with users from the intended user group. The feedback from this evaluation will be used for improving the overall user experience and usability of the app when we go on and change the code base to the Android 4 SDK. During this transition we plan to make use of the Calendar API, which allows making calendar entries for each talk that a user is interested in. Moreover, we plan to integrate social networking and check-in functionality based on Near Field Communication (NFC) into ginkgo mobile as soon as the first NFC-enabled tablets will be available (see Reinhardt et al. (2011b) for a description of the possibilities of NFC-based communication in scientific events).

REFERENCES

- Battle, R., Benson, E. (2008). Bridging the Semantic Web and Web 2.0 with Representational State Transfer (REST). *Journal of Web Semantics*, 6, pp. 61–69.
- Fielding, R. (2000). *Architectural styles and the design of network-based software architectures*. PhD thesis, University of California,
- Hamad, H., Saad, M., Abed, R. (2010). Performance Evaluation of RESTful Web Services for Mobile Devices. *International Arab Journal of e-Technology*, 72(1)
- Hammer, E., Recordon, D., Hardt, D. (2012). The OAuth 2.0 Authorization Framework. Available online at <http://tools.ietf.org/html/draft-ietf-oauth-v2> accessed 08 June 2012
- Masse, M. (2011). *Rules for REST API Design*. O'Reilly Media, Incorporated
- Nottingham, M. (2010). Web Linking. See also website: <http://tools.ietf.org/html/rfc5988>
- Priem, J. & Hemminger, B. (2010). Scientometrics 2.0: New metrics for scholarly impact on the social web. *First Monday*, 15(7).
- ProgrammableWeb (2012). Protocol Usage by APIs. See also website: <http://www.programmableweb.com/apis>
- Reinhardt, W.. *Awareness Support for Knowledge Workers in Research Networks*. PhD thesis, Open University of the Netherlands, available online at <http://bit.ly/PhD-Reinhardt>, 2012.
- Reinhardt, W., Maicher, J., Drachsler, H., Sloep, P.B. (2011a). Awareness Support in Scientific Event Management with ginkgo. In: *Proceedings of the 11th International Conference on Knowledge Management and Knowledge Technologies (i-Know'11)*, pp.40:1-40:8, New York, NY, USA. ACM.
- Reinhardt, W., Messerschmidt, T. & Nelkner, T. (2011b). Awareness Support in Scientific Events with SETapp. In W. Reinhardt, T. D. Ullmann, P. Scott, V. Pammer, O. Conlan, and A. Berlanga, editors, *Proceedings of the 1st Workshop on Awareness and Reflection in Learning Networks (ARNets'11) at the EC-TEL 2011 Conference*, volume 790 of *CEUR*, pp. 100–115.
- Reinhardt, W., Mletzko, C., Drachsler, H. & Sloep, P.B. (2012). Design and evaluation of a widget-based dashboard for awareness support in Research Networks. *Interactive Learning Environments* (in print).
- Renear, A.H. & Palmer, C.L. (2009). Strategic reading , ontologies, and the future of scientific publishing. *Science*, 325(5942).
- Richardson, L. (2007). *RESTful Web Services*. O'Reilly.
- Rowlands, I & Nichols, D. (2005). New journal publishing models: An international survey of senior researchers. A CIBER report. Available online at http://www.homepages.ucl.ac.uk/~uczciro/pa_stm_final_report.pdf accessed 08 June 2012.
- Shneiderman, B. (2008). Science 2.0. *Science* 5868(319), pp. 1349-1350.
- Waldrop, M. (2008). Science 2.0. *Scientific American* 298(5), pp. 68-73.