STEVIE—Collaborative Creation and Exchange of Events and POIs on a Mobile Phone

Max Braun
WeST, University of Koblenz-Landau, Germany
maxbraun@uni-koblenz.de

Daniel Schmeiß
WeST, University of Koblenz-Landau, Germany
dschmeiss@uni-koblenz.de

Ansgar Scherp
WeST, University of Koblenz-Landau, Germany
scherp@uni-koblenz.de

Steffen Staab
WeST, University of Koblenz-Landau, Germany
staab@uni-koblenz.de

ABSTRACT

We propose a model and mobile application for the collaborative creation and sharing of semantic points of interest (POIs) and events. We annotate events and POIs with semantic categories and support search through an ontology of these categories. By this, users are able to find, e.g., an event of a jazz jam session through the query string *concert* even though the event has only been annotated with the concept of a jam session. Unlike existing applications, each user can modify any event and POI stored in the system. To this end, we consider the creation, modification, as well as deletion operations on events and POIs as contributions within our system. Even with delete operations the actual events and POIs are not permanently removed from the database. This allows rollbacks, e.g., if data has been arbitrarily deleted by spammers. The model has been implemented in our mobile application for collaborative creation and sharing of events and POIs called STEVIE.

Categories and Subject Descriptors

K.4.3 [**Organizational Impacts**]: Computer-supported collaborative work; H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

design, human factors, management

Keywords

mobile collaboration, events, POIs, semantic annotation

1. INTRODUCTION

Mobile applications for creating and sharing points of interest (POIs) have gained widespread popularity in recent time. They allow the users to create and publish one's POIs and share them within a community of other users. However, the existing applications do not provide means for true collaborative creation and sharing of such POIs. A POI published by a user cannot be modified by another user of the community, e.g., to correct mistakes or to update it with new data. Events, which in the context of this work are considered to be temporally extended POIs, are typically not supported by existing systems. The collaborative creation and sharing of events and POIs, however, is useful in various domains such as emergency response and tourism.

In this paper, we present a model and mobile application for the collaborative creation and sharing of events and POIs, called STEVIE (short for Semantic Technologies for EVents and poInts of intErest). Unlike existing applications, each user can modify any event and POI stored in the system. To this end, we consider creation, modification, as well as deletion operations on events and POIs as contributions to the system. This means that delete operations on the actual events and POIs do not permanently remove them from the database and allows for targeted rollbacks, e.g., if data has been arbitrarily deleted by spammers. In contrast to existing applications, STEVIE additionally allows to annotate events and POIs with semantic categories. Thus, it supports searching events and POIs through an ontology of these categories. For example, when a user initiates the query string *concert*, the mobile application STEVIE will also return jazz jam sessions since the concept of a jam session is a subclass of a concert. Apart from the collaboratively created events and POIs, the categories and the category ontology can also be collaboratively modified.

In the following section, we motivate the need for collaborative creation and sharing of events and POIs. The model for representing collaboratively created events and POIs is presented in Section 3. The architecture and implementation of STEVIE is presented in Sections 4 and 5. Current state of the art is reviewed in Section 6, before we conclude the paper.

2. SCENARIOS

The following two scenarios are motivated from the WeKnowIt project[1]. The first one is an emergency scenario: Here the attention is focused on the fast and simple ad-hoc creation of events. A citizen discovers a fire located at a certain place. He marks this place as POI and annotates it with the concept fire. The fire POI is send to the emergency

[1] `http://www.weknowit.eu/`

hotline which asks a police officer who is closeby to verify the situation. The police officer confirms the incident, requests a fire engine to be sent out and adds the current time to the POI for logging reasons. By this, the POI becomes an event. It turns out that the fire is part of a larger chemical incident in an industrial area. Thus, the agents at the strategic emergency planning and coordination center are notified about this event and can react appropriately. They receive the location, time and other properties of the event such as categories. The benefits of such a mobile emergency application are fast communication between the officers and the emergency response entities and the logging of emergency incidents.

In the second scenario, a group of persons is planning a trip to a foreign city. They inform themselves about possible activities and events during the trip in the chosen city. To this end, they locate the city in the mobile application and set the focus of time to the period of the travel. Now the group is able to search and browse for events and POIs that happen at the chosen place and period of time. While on the trip, the group realizes that a certain POI does not exist anymore. Thus, they delete the POI from the application. However, they discover a nice wine festival that happens during their stay in the city and create a new event for that.

As the scenarios have shown, there is a need for a mobile collaborative creation and sharing of events and POIs. In the following section, we present a model how this information can be represented and stored.

3. REPRESENTING COLLABORATIVELY CREATED EVENTS AND POIS

For representing collaboratively created semantic events and POIs, we have created a set of ontologies for modeling the structure and content of the events and POIs, the user accounts of our STEVIE application, and most notably the history of all collaborative user activities. The *base ontology* defines the fundamental concepts such as POI, contribution, user, and their relationships. An event is considered a special case of a POI. It differs from a POI by having temporal information. This understanding of events is based on a pragmatic motivation as all events in our mobile system will have some spatial relation in order to be shown on the map of the mobile application. Thus, in the context of this work an event is considered an entity that comprises both spatial and temporal information similar to domains such as classical mechanics [1]. The base ontology cannot be changed by the users. The *vocabulary ontology* defines event categories such as concert, fair, and others and POI categories such as monument, park, and the like. The vocabulary ontology is open to direct collaborative modification by the users. The *event and POI ontology* comprises all instances of events and POIs and their associated metadata. As such, the ontology directly depends on the categories defined in the vocabulary ontology. Users of the STEVIE application primarily interact with this ontology during POI creation and POI retrieval. The *collaboration ontology* represents any kind of collective activity in the STEVIE application, whose instances are called *contributions*. A contribution is either a creation, modification, or deletion and refers to either an instance of an event or POI or to a category in the vocabulary ontology. Finally, the *user account ontology* contains all instances of users and their account information.

3.1 Base Ontology

The base ontology of our STEVIE application defines the fundamental concepts and their relationships which are instantiated and subclassed in the other ontologies. The major part of the base ontology concerns the definition of concepts used in the collaboration ontology, which are all related to the abstract concept of a *contribution*. The minor part of the base ontology defines concepts mainly used in the event and POI, vocabulary and user account ontologies, such as those of a POI, category, and user.

The class base:Poi is used in the vocabulary ontology as a superclass to all event and POI categories and is therefore instantiated in the POI vocabulary. It is defined here as a subclass of geo:Point, which puts base:Poi in the context of the Basic Geo vocabulary [8], where the location-related properties —geo:lat, geo:long, and geo:alt— are defined in the way they are used in the event and POI ontology. The class base:Poi is also a superclass of events, which is due to the fact that a POI can be turned into an event and vice versa by adding or removing the temporal information.

The base ontology further defines the class base:User, which is instantiated by all users in the user account ontology. It subclasses foaf:Person from the FOAF vocabulary [10] and thus enables instances of base:User to carry the properties foaf:nick for the username, foaf:name for the user's full name and foaf:mbox for an email address. Another property also used in the user account ontology and defined here is base:password, which stores the user's password in a hashed form as an xsd:string.

The content of the collaboration ontology is based on the notion of a contribution, which represents anything a user can contribute to the ontology. Thus, contributions are all indirect instances of the base ontology's class base:Contribution and the different types of contributions are represented by subclasses of base:Contribution. Each contribution type uses the properties base:issuedAt to state the date and time of the contribution and base:issuedBy to indicate which user issued the contribution. Both properties therefore have the generic base:Contribution as their rdfs:domain as well as xsd:dateTime and base:User as their respective rdfs:range.

The type of the resource affected by a contribution is modeled as specialization of the generic class base:Contribution. There is one subclass for instances of events and POIs, namely base:PoiContribution, and one for representing categories, which is base:CategoryContribution. Note that the class PoiContribution refers to instances of both events and POIs, as events differ from POIs only by having temporal attributes (see above). Each of these two contribution classes have an associated property referring to the affected resource. The property base:affectsPoi has the domain base:PoiContribution and ranges over instances of the class base:Poi. Analogously, the property base:affectsCategory is used together with base:CategoryContribution and the base:Category class.

The two contribution types represented by the classes base:PoiContribution and base:CategoryContribution are each further specialized to specify the type of operation, namely create, modify, and delete. For the class PoiContribution the three subclasses base:PoiCreation, base:PoiRemoval, and base:PoiModification are introduced. Analogously, the subclasses CategoryContribution are defined: base:CategoryCreation, base:CategoryRemoval, and base:CategoryModification.

Event and POI modifications have additional properties determining whether a modification instance changes or de-

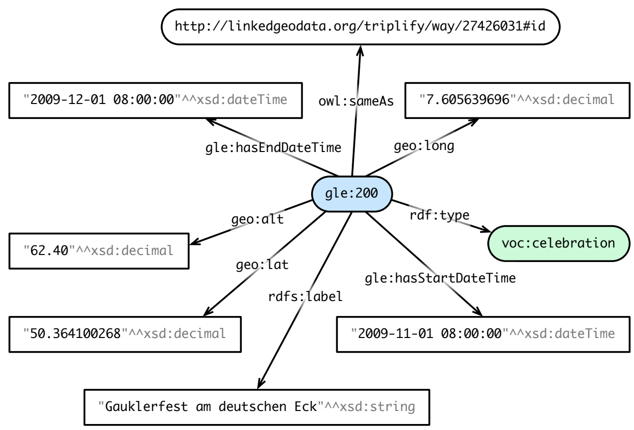


Figure 1: Example of the Event and POI Ontology Showing the Event gle:200.

letes an existing property or adds a new one. The properties for these contributions are base:changesPoiProperty, base:deletesPoiProperty and base:addsPoiProperty.

3.2 Event and POI Ontology

The event and POI ontology contains all information about specific events and POIs and their metadata. It is therefore modifiable by the users via the STEVIE mobile client application. Possible operations are creating, modifying, and deleting instances of events and POIs.

The instances of the event and POI ontology use the gle namespace, which is short for geo-located event. Figure 1 shows this ontology at the example of the event gle:200. Each geo-located event instance has properties describing its human-readable name, its location, and possibly its time or interlinks to instances from other Linked Data [5] sources such as DBpedia [2] or LinkedGeoData [3]. These interlinks to DBpedia and LinkedGeoData are used to connect STEVIE to the Linked Data cloud and to make the events and POIs in STEVIE accessible to other applications.

The human-readable name of an event or POI is expressed using the predefined rdfs:label property, which in this case refers to a literal of type xsd:string. There are three properties from the geo namespace used to describe the location of an event or POI. These are geo:lat for the latitude, geo:long for the longitude, and geo:alt for the altitude above sea level. Which categories the event or POI belongs to is expressed using the instance relation rdf:type referring to a subclass of base:Poi as defined in the vocabulary ontology. In the example, such a relationship is shown with the category voc:celebration. There is also one interlink to a resource in the LinkedGeoData dataset which is considered equivalent to gle:200. Here, Linked Data interlinks are expressed using owl:sameAs.

While the properties name, latitude, longitude, and altitude are definite, i.e., for each event or POI instance there is exactly one, the remaining two or more are contingent. Thus, there are possibly multiple or no categories and interlinks for one POI instance. If there is no category associated, an event or POI is always of the generic type base:Poi. Events additionally have the two temporal definite properties gle:hasStartDateTime and gle:hasEndDateTime.

3.3 Vocabulary Ontology

The vocabulary ontology defines categories which are used for the annotation of events and POIs. The categories provided are initially imported from the LinkedGeoData [3] vocabulary and contain concepts such as fair, monument, park, and others. Initial interlinks from the vocabulary ontology to DBpedia are established using simple string matching. The vocabulary ontology is open to direct access and collaborative modification by users who wish to extend the expressiveness of event and POI annotations.

Each category in the vocabulary ontology has a human-readable name, which is of type xsd:string, and uses the rdfs:label property for both event and POI instances. Categories can have subclass or identity relations among each other. Additional owl:sameAs relations are established with resources from LinkedGeoData and DBpedia referring to the same categoric concept.

3.4 User Account Ontology

All user accounts are represented by instances of the class base:User from the base ontology. Each user account has a username, which is expressed using the property foaf:nick and is used for logging in and identifying the user who created, modified, or removed a POI or category. Other user account data is also stated using the FOAF vocabulary: there is the property foaf:name, which stands for the user's full name, and the property foaf:mbox, which holds the user's email address. The password of the user is associated with the user instance using base:password.

In addition to the human users, the user account ontology also includes virtual non-human users, which represent different algorithms contributing data and metadata. All creations and modifications contributed by the system itself, such as the import of LinkedGeoData categories (user: linkedGeoDataImporter) or the detection and merging of duplicate POIs (user:duplicateFinder), are attributed to special users. The duplicate finder of STEVIE is able to detect duplicate user contributions in form of events and POIs and establishes owl:sameAs relations between them [7]. These virtual users have similar properties to normal users, but provide no passwords and generally no email address. Instead of a full name, they provide a short description of their functionality.

3.5 Collaboration Ontology

The collaboration ontology encodes all collaborative activities performed using the STEVIE application. It is the core part of our work. The ontology comprises contributions by users as well as the results of data and metadata revisions of the virtual users and is therefore modifiable indirectly both by the server and by users via a client. Hence, the collaboration ontology as a whole acts as an implicit version history of all events and POIs and their categories.

The collaboration ontology provides different concepts specialized from the base ontology class base:Contribution. These serve different purposes along the following aspects: The resource affected by the contribution is either an instance of an event or a POI or a category. There are different kinds of contributions for creating, removing, and modifying the two kinds of resources, namely instances and categories. There is a difference between the ways in which definite and contingent properties may be modified. The former can only

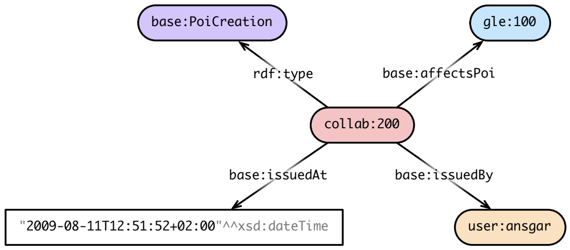


Figure 2: Collaboration Ontology Showing the Collaboration Activity collab:200 of an Event Creation.

change their value while the latter can be deleted or added and can therefore appear multiple times or not at all.

A single contribution is represented in the collaboration ontology by an instance of any of the base:Contribution subclasses. These exist for each resource type (event, POI, or category) and type of operation (creation, modification, or removal) and use the `collab:` namespace. All contribution instances use at least two common properties: base:issuedAt states the date and time the contribution was issued. The user of type base:User who issued the contribution is related to the instance through its base:issuedBy property. The various possible combinations of contribution types, affected resources, and modifiable properties are illustrated with the following two examples:

The first example is shown in Figure 2 and demonstrates how the creation of a new event is expressed in the collaboration ontology using the contribution instance collab:200. The target of this contribution is the event instance gle:100 from the event and POI ontology, which is expressed using the property base:affectsPoi. The instantiation of base:PoiCreation denotes that collab:200 represents the creation of a new event. A literal of type xsd:dateTime states that this event was created on August 11, 2009 at 12:51:52 in a time zone two hours ahead of UTC. The example also uses base:issuedBy to state that the user, denoted by user:ansgar, issued this contribution and therefore created the event.

Figure 3 shows an example where a property of an event is modified. The contribution instance representing this modification is collab:201 from Figure 2. It instantiates base:PoiModification and the event instance which is modified is gle:100, the same as in the previous examples. Analogously to the event creation example, the modification is expressed using the property base:affectsPoi. Unlike event creation, event modification requires the property base:changesPoiProperty to express which property of the affected resource is modified. In this case, the value for the property rdfs:label of the event, i.e., its name, is changed. The property base:poiPropertyValue is used to express the new value of the modified property of the affected instance. The event modification uses the same properties as an event creation to express which user issued this modification and when the contribution was issued.

In the case of contributions affecting events or POIs, there is also the removal of an event or POI. These require two additional types of event or POI modifications, which represent the addition and deletion of properties. For categories, there are additional contribution types for creation and removal, analogous to the corresponding contributions affecting events or POIs.

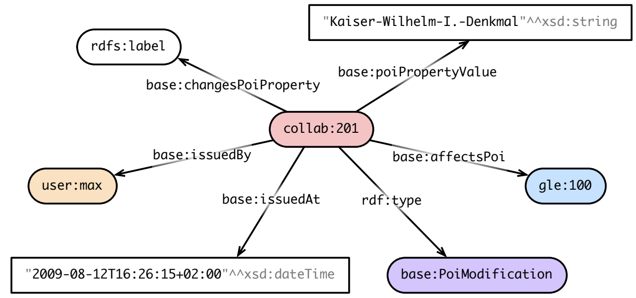


Figure 3: Collaboration Ontology Showing the Modification of the Event gle:100.

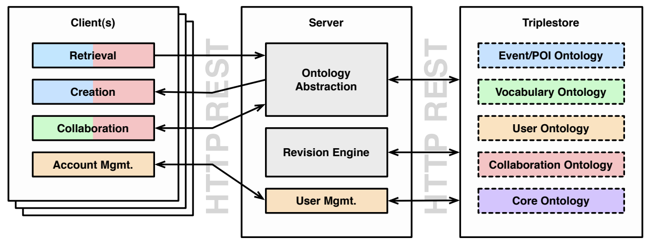


Figure 4: Overview of the Architecture of STEVIE.

None of the events, POIs, or categories affected by a contribution are changed directly. Instead, their original instances are unaffected and only when the server receives a request for an event, a POI, or a category, it retrieves all relevant contributions and applies the modifications they encode before returning the data to the client and shows it to the user.

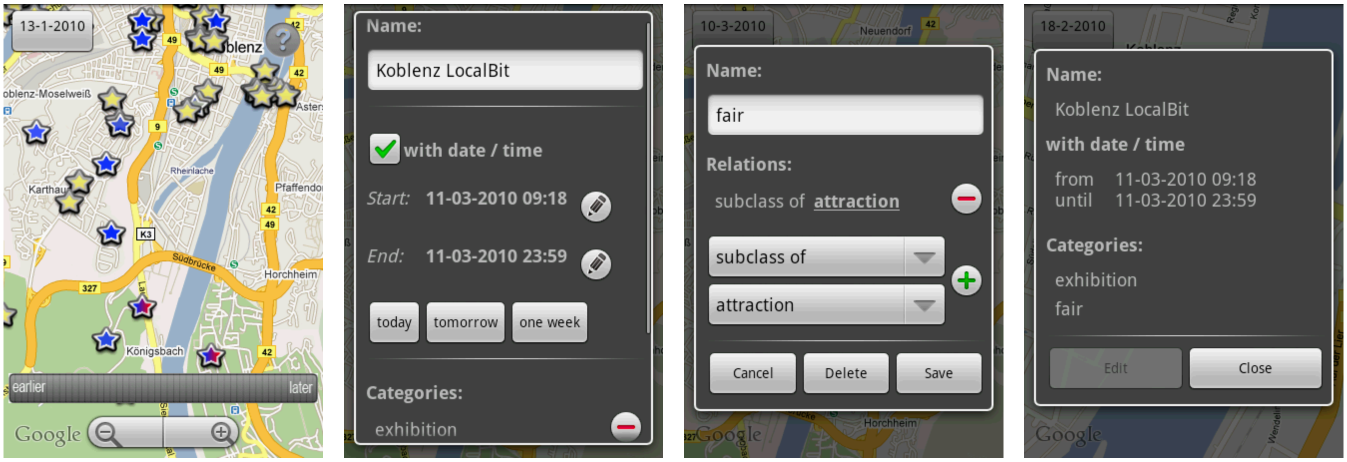
3.6 Summary

So far, we have seen the model that allows us to represent collaboratively created events and POIs. In the following sections, we present the architecture of STEVIE and the implementation of our mobile client that supports this collaborative creation and sharing of event and POI information.

4. ARCHITECTURE

The architecture follows a client-server design and consists of three tiers. The first tier constitutes the client application, which communicates with the second tier, the STEVIE server. Finally, the server itself communicates with a triplestore for storing and retrieving semantic data. Both, the client-server and the server-triplestore communications are designed as RESTful web services [12]. Thus, it is possible to have multiple clients in operation with the STEVIE server such a mobile client, a desktop PC client, and an iPad client.

The client tier is designed for a thin client since little processing is done locally and most tasks are delegated to the server. The main functions offered by a client to the users are shown on the left hand side of Figure 4. Users can search for events and POIs (*Retrieval*) or create new ones (*Creation*). *Collaboration* refers to the collaborative editing of event and POI metadata as well as of the vocabulary ontol-



(a) Map with search results (b) Creation of an event (c) Ontology editor (d) Properties of an event

Figure 5: Screenshots of the STEVIE application

ogy. *Account management* comprises the login and logout processes and the editing of user details.

The server provides clients with data they requested and receives data from them for further processing or permanent storage. To retrieve or store data, the server then communicates with the triplestore and exchanges RDF [9] triples. The higher-level functions of the server are illustrated in the center of Figure 4. The *ontology abstraction* converts client HTTP `GET` requests into the appropriate SPARQL query for the triplestore and returns the filtered result as XML back to the requesting client. It also converts HTTP `POST`, `PUT`, and `DELETE` requests to the corresponding RDF triples and stores them on the triplestore. The *revision engine* employs data mining algorithms on the POI data and metadata from the ontology to improve their quality such as detecting and merging duplicate POIs. *User management* comprises both the authentication of users, for which login information from the client is compared to user account data from the triplestore, and the handling of users editing their account data.

The triplestore shown on the right hand side of Figure 4 is the central database of the STEVIE application. It permanently stores data as RDF triples, which can be queried and updated from various parts of the system. There is no direct connection between clients and the triplestore, which is instead abstracted by the server.

5. PROTOTYPE

The STEVIE application and architecture has been implemented using Google's Android platform[2] for the client tier. The server tier is implemented as Apache Tomcat[3] servlets and the storage tier is provided by the triple-store Sesame[4].

Once the user has started the mobile client on his Android-enabled phone, a map view is shown, where he or she can query for events and POIs. The results of an example query showing several events and POIs is depicted in Figure 5(a). The POIs are visualized as yellow stars whereas the events are depicted as blue stars. In addition, the events are visualized according to their temporal distance to the currently selected date. The events that are close to the currently selected date are shown in dark blue whereas events farer away become increasingly transparent. If an event happens at the currently selected date, it is shown shaded half blue and half red. Events that occurred before the currently selected date or after 30 days beginning from this date are not shown to the users at all. The current date is shown in the top left of the screen and can be modified by clicking on it and selecting a date from an appropriate widget that shows up. Alternatively, the time can be selected by spinning the timeslider widget at the bottom of the screen.

In order to provide contributions to the STEVIE application, the users have to log in first. Subsequently, they can create, modify, and delete events and POIs. As described in Section 3, the collaboration ontology is used to track all contributions made by the users. A screenshot of STEVIE showing the creation of an event is shown in Figure 5(b). The user has entered the event name *Koblenz LocalBit*, a local IT fair in Koblenz, Germany. By checking the box next to "with date/time", the user can specify the POI has temporal information. The start date and end date of the event is specified by pressing the edit buttons at the right and selecting a date and time from a corresponding widget. In addition, the user can choose from three predefined time periods, namely today, tomorrow, or one week. The events and POIs are annotated with semantic concepts such as *exhibition* in the example shown in Figure 5(b). The concepts are taken from the collaboratively created and modifiable vocabulary ontology as introduced in Section 3.3.

An example of adding a new concept to the vocabulary ontology is shown in Figure 5(c). It shows adding the concept of a *fair* to the vocabulary and specifying it as "sublcass" of attraction. Other relations that can be specified are "superclass of" and "equivalent to". All events and POIs on the map can be touched to display their properties, namely name and the categories as well as start date and end date in the case of an event. An example screenshot of this inspection view is depicted in Figure 5(d). In this example, the user is not logged into the system. Thus, he or she cannot edit the event properties.

[2]`http://www.android.com/`
[3]`http://tomcat.apache.org/`
[4]`http://www.openrdf.org/`

6. RELATED WORK

One of today's most sophisticated mobile, context-aware applications is IYOUIT [6]. It supports social relationships, location records, and weather conditions. DBpedia [2] is an effort to extract structured information from Wikipedia and publish it as Linked Data [5]. With DBpedia Mobile, users can explore the structured information extracted from Wikipedia [4]. In addition, there are several other commercial applications for mobile creation and sharing of POIs such as Layar[5], Wikitude[6], Foursquare[7], Gowalla[8], and Brightkite[9]. However, these applications do not allow for the collaborative creation and sharing of such POIs. Thus, a POI published by a user cannot be modified by another user. In addition, the existing applications do not explicitly support the creation and sharing of events. Finally, the events and POIs in our STEVIE application are annotated with semantic concepts of a collaboratively modifiable vocabulary. This allows for searching within the hierarchy of the semantic concepts.

For collaborative ontology creation, we find approaches like Holsapple et al. [13] and the DILIGENT framework [16]. They describe a collaborative ontology design process that is carefully organized and requires a certain level of expertise and explicit communication between the participants. These innate requirements do not hold in a volunteer collaboration context such as considered for the STEVIE application. Balance is needed between the limited commitment of volunteer non-expert user contributions [11] (resulting in quality deficiencies of the data) and the need to retain a necessary degree of organization of the collaboration. Fortunately, even scarcely structured data from folksonomies or websites already carries automatically extractable information about lightweight ontologies [15, 14]. The collaborative ontology of the STEVIE application has a level of organization ranking between these informal structures and the introduced professional process models.

7. CONCLUSIONS

We have presented with STEVIE a model and application for the mobile collaborative creation and sharing of events and POIs. Unlike existing applications, STEVIE provides explicit support for modeling events. The semantic model underlying our STEVIE application follows the principles of the wiki approach, i.e. each user can modify any event, POI, or category stored in the system. All user contributions such as creation, modification, and deletion of events and POIs are stored along with their history. Thus, we can do targeted rollbacks of the database content if necessary. Annotation of the events and POIs with a collaboratively modifiable vocabulary of semantic concepts allows for search within the hierarchy of these concepts. Further information and videos demonstrating STEVIE mcan be found at: `http://west.uni-koblenz.de/Research/systeme/csxPOI`.

In the future, the event and POI ontology will also permit to connect to media elements like photos. The user will be able to click on a certain event or POI, take a photo and upload it to Flickr[10]. By this, a connection between the event or POI instance and the Flickr photo is established. In addition, we plan to publish events and POIs directly from STEVIE to LinkedOpenData services such as LinkedGeoData. Finally, structural relations between events and POIs shall be supported.

Acknowledgments.

This research has been co-funded by the EU in FP7 in the WeKnowIt project (215453).

8. REFERENCES

[1] V. Arnold. *Mathematical Methods of Classical Mechanics.* Springer, 1997.

[2] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. DBpedia: A Nucleus for a Web of Open Data. *LNCS*, 4825:722–735, 2007.

[3] S. Auer, J. Lehmann, and S. Hellmann. LinkedGeoData – Adding a Spatial Dimension to the Web of Data. In *Int. Semantic Web Conference*, 2009.

[4] C. Becker and C. Bizer. DBpedia Mobile: A Location-Enabled Linked Data Browser. In *1st Int. Workshop on Linked Data on the Web*, 2008.

[5] T. Berners-Lee. Linked Data, 2006. `http://www.w3.org/DesignIssues/LinkedData.html`.

[6] S. Boehm, J. Koolwaaij, M. Luther, B. Souville, M. Wagner, and M. Wibbels. Introducing IYOUIT. In *7th Int. Semantic Web Conference*, 2008.

[7] M. Braun, A. Scherp, and S. Staab. Collaborative creation of semantic points of interest as linked data on the mobile phone. In *ESWC*. Springer, 2010.

[8] D. Brickley. W3C Semantic Web Interest Group Basic Geo (WGS84 lat/long) Vocabulary, 2006. `http://www.w3.org/2003/01/geo/`.

[9] D. Brickley, R. V. Guha, and B. McBride. RDF Vocabulary Description Language 1.0: RDF Schema, 2004. `http://www.w3.org/TR/rdf-schema/`.

[10] D. Brickley and L. Miller. FOAF Vocabulary Specification, 2007. `http://xmlns.com/foaf/spec/`.

[11] T. Chklovski and Y. Gil. Towards Managing Knowledge Collection from Volunteer Contributors. In *Knowledge Collection from Volunteer Contributors*, 2005.

[12] R. T. Fielding. *Architectural Styles and the Design of Network-based Software Architectures.* PhD thesis, University of Californa, Irvine, USA, 2000. `http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm`.

[13] C. W. Holsapple and K. D. Joshi. A Collaborative Approach to Ontology Design. *Communications of the ACM*, 2002.

[14] H. Lin, J. Davis, and Y. Zhou. An Integrated Approach to Extracting Ontological Structures from Folksonomies. In *6th Annual European Semantic Web Conference*, 2009.

[15] P. Mika. Ontologies are us: A unified model of social networks and semantics. *Web Semantics*, 5(1), 2007.

[16] H. S. Pinto, C. Tempich, and S. Staab. Ontology Engineering and Evolution in a Distributed World Using DILIGENT. *Handbook on Ontologies*, 2009.

[5] `http://www.layar.com/`

[6] `http://www.wikitude.org/`

[7] `http://www.foursquare.com/`

[8] `http://www.gowalla.com/`

[9] `http://www.brightkite.com/`

[10] `http://www.flickr.com/`