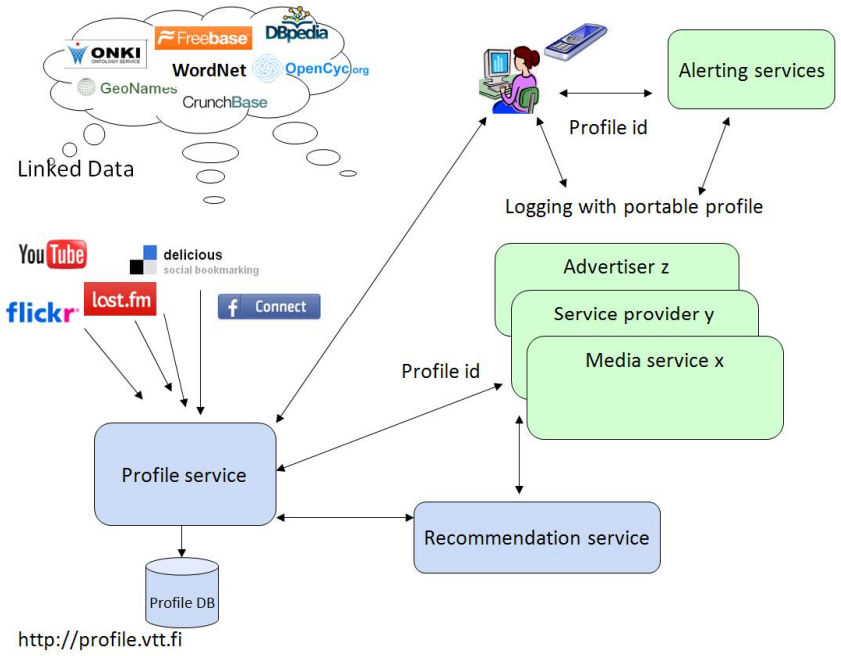

D3.0.1.2

Context awareness – portable profiles, HTML5 and advertiser's metadata

Author(s): Sari Vainikainen, Timo Laakko, Raphael Giesecke, Petri Vesikivi

Confidentiality: Public

Date and status: 13.1.2011 Version 1.0

This work was supported by TEKES as part of the next Media programme of TIVIT (Finnish Strategic Centre for Science, Technology and Innovation in the field of ICT)

Version history:

Version	Date	State (draft/ /update/ final)	Author(s) OR Editor/Contributors	Remarks
0.1	12.11.2010	Draft for table of contents	Sari Vainikainen	
0.2	15.12.2010	Portable profiles	Sari Vainikainen	
0.3	17.12.2010	Advertiser's metadata	Raphael Giesecke	
0.4	21.12	HTML 5	Petri Vesikivi	
0.5	22.12	Profiles and context	Timo Laakko	
0.6	27.12	Executive summary, conclusions	Sari Vainikainen	
1.0	13.1	Updates based on feedback	Sari Vainikainen	

Participants	**Name**	**Organisation**
	Sari Vainikainen Timo Laakko Raphael Giesecke Petri Vesikivi	**VTT** **VTT** **Aalto TKK** **Metropolia**

Next Media
www.nextmedia.fi
www.tivit.fi

Executive Summary

The long term goal of WP 3 in Next Media is to develop tools and methods to be able to reach users and provide them with relevant content in different services based on each user's interests, needs and context.

The vision of semantic portable user profiles is one building block for reaching this goal. The aim is to support the creation and utilisation of service independent portable profiles in a way that the user is in control of the profile data and that the profile can be used in multiple services for getting personalised recommendations.

The report describes the VTT's profile service (http://profile.vtt.fi/), which supports creating and utilising semantic portable profiles. On the demonstration personalised recommendations relating to events (Evenemax/ KSF Media) and magazines articles (Sanoma Magazines) can be explored.

In the Event Management case portable profiles and recommendations are integrated into mobile application showing personalised event recommendations to users. Recommendations based on portable profiles have been developed also for the Palveleva Huittinen portal (http://www.palvelevahuittinen.fi/) that has a supply of services in Huittinen area.

The report describes different steps of integrating portable profiles into services. These steps include defining user authentication methods, defining what kind of personalised services will be offered to users, integrating recommendation methods and defining how the profile will be updated. The profile service has a REST API for creating, reading, updating and deleting a profile as well as for getting recommendations.

The profile service uses Linked Data in creating of semantic user profiles as well for attaching metadata to users' interests or content metadata. Methods for enhancing service providers' vocabularies with semantic metadata in order to make better personalized recommendations have been developed. Semantic enhancement includes different phases such as automatic semantic analysis, checking of the result of the analysis and adding new relations with help of the developed mapping tool, and fetching and storing additional information relating to these linked meanings.

Based on the metadata of content of media services different recommendation methods can be used. Three different recommendation methods have been implemented; semantic recommendation, free-text indexing based recommendation and recommendations that combine both methods. In addition different ranking methods such as place and time for recommendations can be defined based on requirements of cases.

The portable profile has focused so far on user's interest profiles, but need for other type of user information such as social network has been arisen as well. Context information helps the user to get the relevant content in a relevant format for the current situation. Different levels of context (e.g. location, calendar information, time, "at a hobby", "at home") can be described with help of ontology and to be used in different situations of a user. The context information may also be related to content and services. This can be utilized, e.g., for personalized content selection and delivery, and recommendation systems. There is an

increasing need for exchange and trade of contextual information between different providers of context-aware services.

Besides linking metadata of media content to user metadata it is important to be able to link advertiser's metadata to user metadata. Contextual targeting promises higher return on (advertisement) investment than a "one fits all" approach. This is valid for both mass products and more specific (and often more complex and expensive) customised products.

The report defines research questions relating to roles and knowledge processes of content producers and advertisers e.g. how the metadata is defined and exchanged. Interviews have been made with both representatives. The interviews show that even if all players in the ecosystem know their "standard" professional roles and perform them well, and even if metadata is exchanged to some extent, the Finnish media ecosystem is not sufficiently prepared for paradigm changes in advertisement as being practiced and refined by global players such as Facebook.

HTML 5 is shaping the way mobile applications are created. It enables responsive browser based applications through multithreading (web workers). By utilizing application cache and local storage browser based applications will work also while off-line. Canvas, video and audio tags provide for graphically appealing user interface. With HTML5 a browser based application can provide a truly competitive user experience when compared to native applications. However it is unlikely that browser based applications would replace native applications in applications requiring a lot of processing power and direct access to phone capabilities. The report outlines the features and opportunities of HTML5 and experiences of using HTML5 for developing mobile context aware services.

Table of Contents

1 Introduction

This report contains four different parts: creating and utilising portable profiles, relating context information with user profiles, opportunities and experiences of HTML5 for developing mobile context aware services and challenges of context aware advertising.

This report describes the vision and background of portable profiles. In 2010 the focus of the research was on developing methods for utilising the semantic profiles in personalized recommendations. Methods for portable profiles and personalised recommendations have been developed in different cases. The first of these cases was magazine article and event recommendations in the Tivit Flexible Service Crossmedia project. In Next Media, work with different recommendation methods has been continued. The cases of event management and the service catalogue for Huittinen Sanomalehti have been used for gathering general requirements for the profile service and the recommendations.

In the report different steps for integrating the portable profiles into services are described, as well as the current status of implementation. Creation and utilisation of portable profiles are based on semantics. Solutions for using Linked Data for creating profiles and for matching user’s interests with content metadata are described. The report also includes short descriptions of different recommendation methods that are based on semantics, free-text indexing or a combination of both methods.

2 Creating and utilising semantic portable user profiles

By Sari Vainikainen, VTT

2.1 Vision of portable profiles

The aim is to support the creation and utilisation of service independent portable profiles in a way that the user is in control of the profile data and that the profile can be used in multiple services. The goal is to provide users with relevant content in different services based on each user's interests, needs and context. Recommended content can be for example events, articles, advertisement or other people interested about the same topics. The basic idea of the profile service and portable profiles have been visualised in Figure 1.

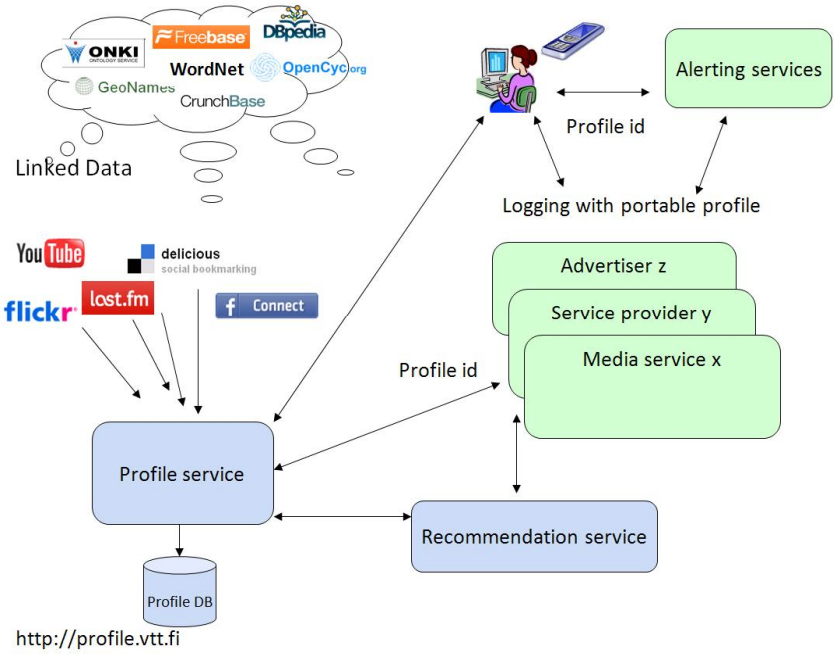

Figure 1 The profile service and portable profiles

2.2 Background of the profile service and example cases

The development of the profile service has been started in earlier projects. Initial development has been made in the EU project Pharos. Development work continued in the Tivit SHOK Flexible service Crossmedia project. More information relating to the implementation of the profile services and creation of semantic profiles can be found at Bäck, A. et al. 2009.

The first experiences of using profile service for recommendations have been gathered in the Tivit SHOK Flexible services Crossmedia project were the profile service was used in two cases: recommending magazine articles and recommending events. In the user tests event recommendations were tested by using event data in KSF Media's Evenemax event database and magazine article recommendations by using a selection of articles published by Sanoma Magazines Finland (SMF). The tests were made to get users' opinions and experiences of the concept of service independent user profile and creating such profile with semantic support, and to test what users thought about recommendation based services as well as to test how well event/ article recommendations could be made based on the profile.

The test persons had positive attitudes towards the concept of the service independent profile and towards exploring event or article recommendations created based on their profiles (See Figure 2).

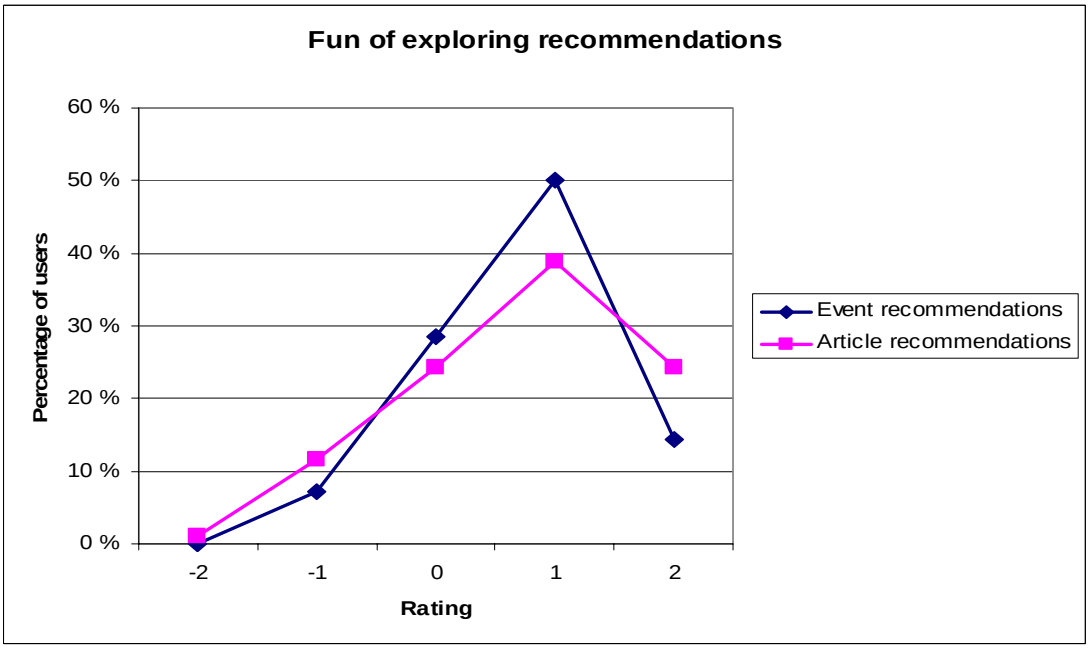

Figure 2 How fun it was to exploring events and articles based on one's personal profile (2 = very fun; -2 = not fun at all).[Bäck, A. (2010)]

One aspect needing attention in further development is making it possible to express more complex issues as interests than only single tags or concepts. Some users also expressed concerns about how much work creating and maintaining the profile would require. Event related services, as well as news and magazine type of services were regarded as potential service area by users for using personal profile for personalisation. The results have been reported in the Crossmedia deliverables [Bäck, A. 2010; Alanko, T et al. 2010; Andelin, J-E, Bäck Asta. 2010].

These cases were used as basis when developing recommendation methods further in the Next Media. In Next Media different methods for utilising a semantic user interest profile in recommendations have been developed. Also methods to

support easy integration of portable profiles with service providers' services have been developed.

The demonstration of the profile service can be found at http://profile.vtt.fi/ . On the demonstration personalised recommendations relating to events (Evenemax/ KSF Media) and magazines articles (Sanoma Magazines) can be explored. The profile service supports Finnish, Swedish and English.

In the Next Media WP3 Event Management case portable profiles and recommendations are integrated into mobile applications showing personalised event recommendations to users. The same features will also be integrated into TV and Facebook applications in 2011. In the Event Management case the profile service is deployed into Amazon EC2 cloud to be used in use scenarios relating to event management.

In Next Media recommendations have been developed also for the Palveleva Huittinen portal (http://www.palvelevahuittinen.fi/) that has a supply of services in Huittinen area. There is a similar portal to Palveleva Huittinen that is a parallel development site. The new site will provide recommendations of the services in the Huittinen area based on the users' preferences. When the user first time enters the site he is asked to create the profile. If the profile is completed the recommendation engine returns the lower service category, as existing in Palveleva Huittinen, that is most relevant to the user. The portal shows random recommendations from this category to the user at the lower part of the frame. The work for more relevant recommendations will continue on 2011.

2.3 Integration to services

Integrating portable profiles into services requires specifying of different steps such as which user authentication methods to use, what kind of personalised services will be offered to users, integrating recommendation methods and defining how the profile will be updated.

The profile service has a REST API for creating, reading, updating and deleting a profile as well as for getting recommendations. A description of API can be found at http://profile.vtt.fi/restapi.html.

2.3.1 User authentication

The profile service supports different user authentication methods. A user is able to log in with his Facebook account or by using OpenID accounts such as Google, I-names and Yahoo. Also traditional registration by user name and password is possible.(See Figure 3)

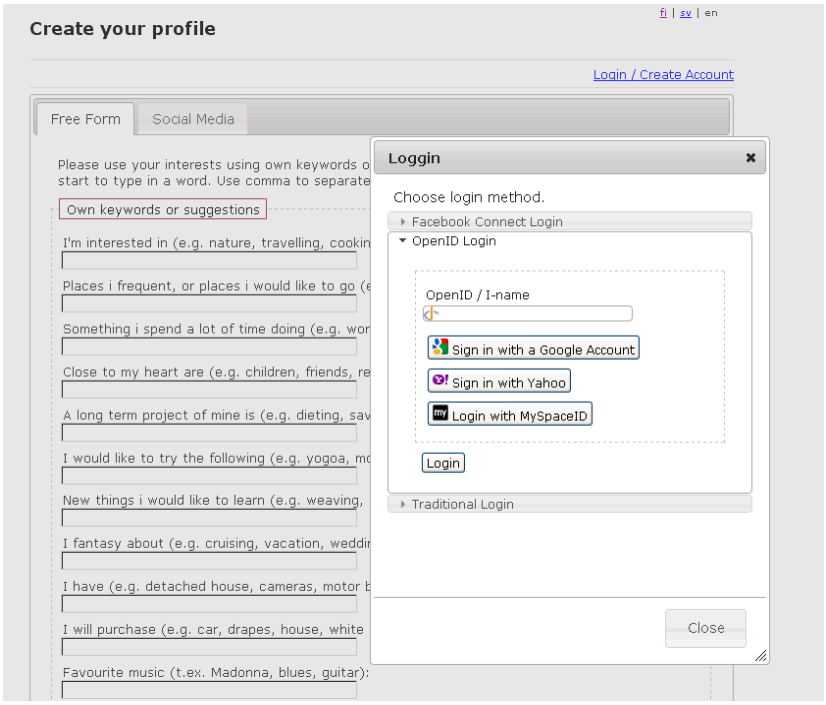

Figure 3 The profile service supports different login methods.

When integrating portable profiles into existing services it needs to be defined which user authentication will be supported and how the profile information is transferred between the profile service and service provider's service.

If a service does not already have user registration, the easiest way is to integrate Facebook Connect or OpenID authentication to the service. Profiles created in the profile service can be attached to these accounts. A link to the profile service for creating and maintaining a user profile can be made in the service. One benefit of a portable profile concept is that a service does not need to maintain their own profile creation pages and they could still offer personalised services. The page for creating a profile can be fine-tuned based on the requirements of a service, for example colours can be changed and the fields for creating the profile can be selected. An altered profile creation page for Evenemax can be seen in Figure 4.

Figure 4 The profile creation page for Evenemax service.

If a service already has user registration, some functionality to connect the existing user id of the media service with the user's portable profile id will be needed. One solution is to offer a user opportunity to link his/her portable profile to a profile created on the service. Information on the user's profile on a service can also be analysed semantically and attached as part of the portable profile of the user.

In cases when a user does not have a portable profile yet and she has logged in to a service by using for example Facebook Connect she can be offered to create the portable profile. The initial profile can be made for example from information on the user's Facebook profile, user's activity e.g. search terms, viewed content on the service can also be used for creating an initial portable profile for the user to accept.

2.3.2 Creating a profile

Both manual and automatic methods are provided for building the user profiles. A user is able to create her profile explicitly by giving her interests with help of different fields which can be seen in Figure 5.

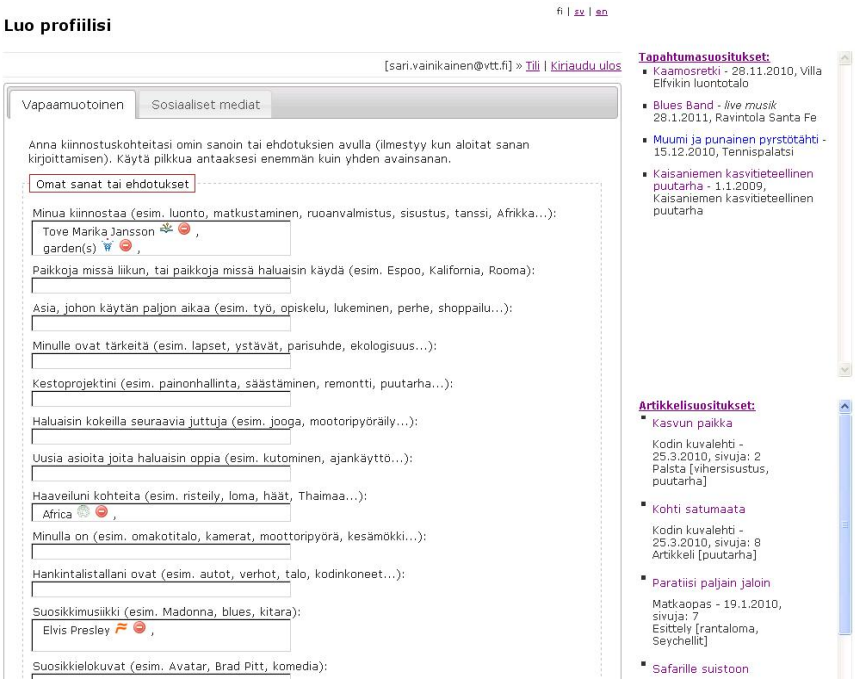

Figure 5 Creating a profile. Personalised recommendations of events and articles can be seen on right. In a similar way the recommendation results can be integrated as a part of a service.

A user is also able to import her tags from existing social media services such as Delicious, Flickr, YouTube and Last.fm. With help of open API's of social media services user's tags are fetched, semantically analysed and the result is shown to the user (See Figure 6). If a user is logged in by using Facebook Connect she is also able to use Facebook account for creating the profile in the same way.

Figure 6 Creating a profile with help of social media accounts.

2.3.3 Personalised services

A service provider needs to define what kind of personalised services will be offered to users and how they will be shown to a user on a user interface.

The profile service includes an embeddable widget that can be used for showing the recommendations on the page of service. Recommendations can be shown also as a part of the profile service so that a user can see recommendation already on her/his profile page. Example of the widgets can be seen in Figure 5 and in Figure 6 showing recommendations of events and articles. One alternative possibility is to build alerting services which send users reminders or alerts when content relating to their interests is available.

Another possibility is to use the profile service for creating a profile not only for a person but also for a site. For example the service would be able to show advertisements on its page based on selected topics.

2.3.4 Integrating content to recommendations

Different recommendation methods can be used depending on what kind of metadata content has.

Recommendations can be made based on full text index, semantic metadata of content or based on both methods. Also different filters relating to places or time can be used. Recommendations are described in more detailed in Chapter 2.5.

In order to be able to match content with a user's profile we add mappings between Linked Data which is used for describing users' interests and metadata describing content. The description of the automatic semantic analysis that is used for analysing a user's interests and for mappings is published in [Nummiaho, A.et. al.2010].

If a service uses some category or vocabulary for classifying content, the profile service includes methods for adding semantics to them to be used in recommendations. Mapping of vocabularies with Linked Data is described in chapter 2.4.

2.3.5 Updating the profile

It is important that keeping the profile updated is easy to the user. Updating the profile based on what the user does in different services where she has logged in with her portable profile will help the user on this.

The profile service includes APIs for creating and updating the profile. Through these APIs a service can update user's profiles based on user's actions on the service. The user is able to modify the profile and decide whether she or he accepts the changes.

Learning profiles will be one research topic in 2011.

2.3.6 Storing a user profile

One important question relating to portable profiles is where it can be stored during its lifecycle in a way that a user is aware and on control of her profile.

So far we have implemented the prototypes were profile information is used directly from the profile service through APIs.

In some cases a service provider might like to store profile information into their database as well. If the profile information will be stored on service provider database the user should be aware of that. It is also important that an up to date version of the profile will be used.

One possibility is that a user would like to store the profile locally on her site or for example on a mobile device.

When making the business model and concept of portable profiles more detailed these different use scenarios needs to be defined as well.

2.4 Matching content and users' interests by using Linked Data

The profile service uses Linked Data in creating of semantic user profiles as well for attaching metadata to users' interests or content metadata. This information is used in personalized recommendations.

2.4.1 Linked Data principles

There are many existing semantic databases that have published their knowledge through open API's and which use the Linked Data principles defined by Tim Berners Lee [Berners-Lee, T.(2006)]. Based on these principles, URIs are used for describing concepts and URIs points to the information describing the concept. Many concepts are represented in multiple databases, but databases include links to other databases describing the same concepts. So Linked Data opens the doors between different databases to be used for finding related things. Figure 7 shows a part of the cloud of such databases. Examples of the databases used by the profile service are Freebase which is American community built database, DBpedia which is based on Wikipedia, Finnish KOKO ontology of ONKI service and Geonames geographical database. The reason to use several databases is that different knowledge bases contain different kinds of knowledge and not all information is available in one database. KOKO is good for general terms giving information about related things and Freebase and DBpedia offer knowledge relating to persons, music, movies etc. Users' interests and freely given keywords can be very heterogeneous depending on the context where the application is used. Therefore, using multiple databases makes it more likely to find the intended meaning.

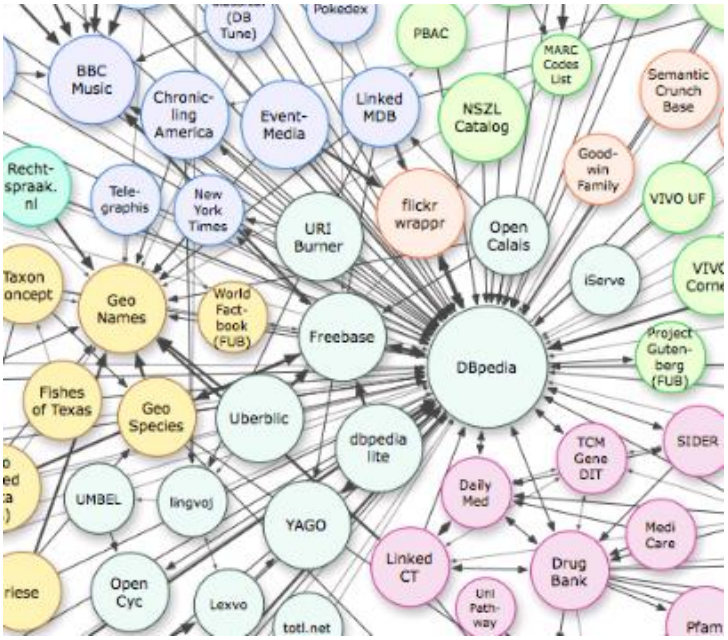

Figure 7 Linking Open Data cloud diagram, by Richard Cyganiak and Anja Jentzsch. http://lod-cloud.net/

2.4.2 Semantic tagging widget

This Linked Data is used in the profile service in different ways. The user is able to define explicitly his/her interests with help of Linked Data. When the user starts to type an interest, suggestions coming from ontology vocabularies are shown below the interest field. Different semantic databases such as KOKO, Freebase and DBpedia are used for suggestions. When user explicitly selects the suggested tag, its meaning is defined as an URI to the database. Based on this more knowledge can be accumulated, for example different language versions, music genre of an artist or geo coordinates or place hierarchy of location tags.

We call this developed functionality the semantic tagging widget (See Figure 8). Based on requirements of each case, the widget can be configured to use only one ontology, a subset of one, several of them or any combination of these. It can also be configured to use new vocabularies and ontology. Semantic tagging widget can also be used for adding metadata to content.

Figure 8 Semantic tagging widget for creating a profile and for annotating content. Automatic suggestions comes from different online databases such as KOKO, Freebase and Geonames.

2.4.3 Semantic tag analysis

We also use Linked Data for analyzing automatically users' interests and linking given plain text keywords to URIs in existing semantic databases. The workflow of the analysis can be seen in Figure 9.

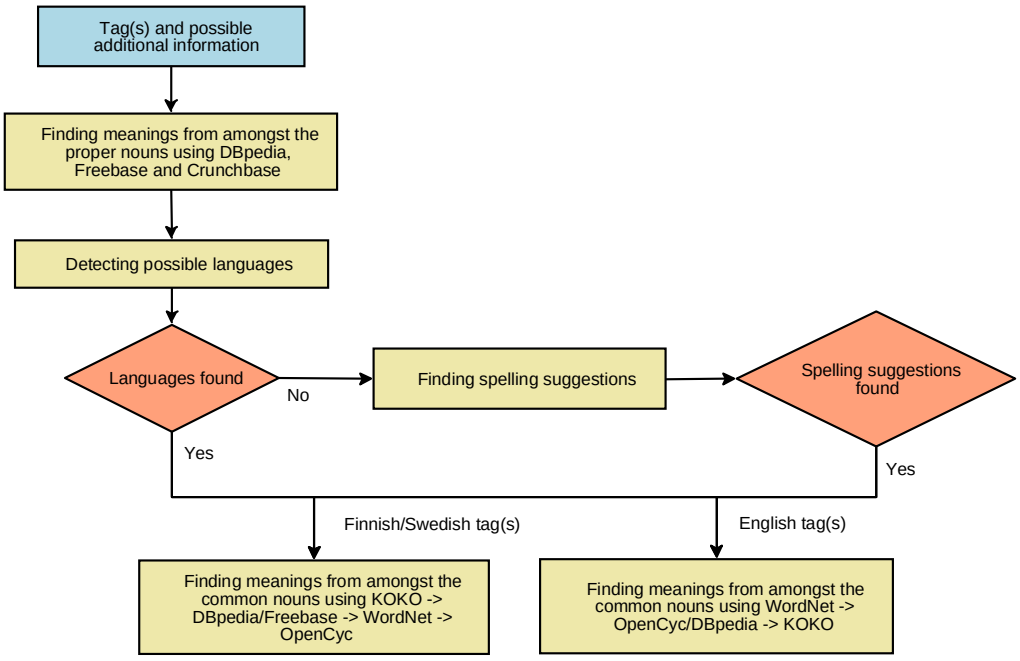

Figure 9 Semantic analysis of tags. A simplified workflow.

As the first step, the analysis tries to recognize the language of the tag. We support lexical analysis and misspelling corrections for Finnish and English tags.

The analysis proceeds differently depending on the language. The analysis for English tags starts with WordNet, then followed by the use of OpenCyc and DBPedia and completed by trying to find meanings from KOKO. For Finnish tags, the analysis starts with KOKO which is designed for Finnish concepts, and is then followed by the use WordNet and finally with OpenCyc and DBpedia. The analysis uses Linked Data to get relations to different databases. More detailed description of the analysis can be found in [Nummiaho, A. et. al. 2010].

2.4.4 Enhancing content with semantic metadata

We have developed methods for enhancing service providers' vocabularies with semantic metadata in order to make better personalized recommendations. This information is basis for semantic recommendations described in the chapter 2.5. Semantic enhancement includes different phases such as semantic analysis, checking of result and adding new relations with help of mapping tool, and fetching and storing additional information relating to these linked meanings.

Semantic analysis

Our semantic tag analysis and Linked Data is used for adding semantic meanings and related things to a service provider's ontology or content categorization. This information is used for matching content and users' interests.

If a service already has some ontology for annotating its content, like in case of magazine articles, the concepts of ontology will be run through our semantic tag analysis for creating links to semantic meanings by utilizing Linked Data.

If a service has some kind of categories for annotating content these categories can also be mapped to semantic data in similar way. In our case studies events were annotated based on Evenemax's event categories. The service of Huittinen includes categories for services. Categories were transformed into the SKOS (Simple Knowledge Organization System)-ontology which was imported to the profile service and the concepts were run through the semantic analysis and links to semantic data were done. The concepts were linked to semantic data with property skos:closeMatch.

Mapping tool

Although the result of the automatic semantic analysis is a good start for creating semantic meanings for concepts, it also produces faulty meanings. Therefore the result needs to be checked manually. For this purpose we have developed an initial version of a mapping tool. First the graph where the results of the automatic mappings of an ontology or a category have been stored will be selected. The mapping tool shows all the created mappings to the concepts. With help of the tool, false mappings can be removed and missing ones added (See Figure 10). Semantic analysis also produces alternative meanings for a same term, which helps defining correct meanings for the false mappings. In addition with help of the tool new meanings for the selected term can be added. This functionality uses the automatic tag analysis that is described earlier. With help of the mapping tool also skos:related type of relations can be added manually.

Figure 10 Mapping tool; Modifying and adding the meanings for the concepts of the selected ontology or category. The example is from the mappings of the categories of the Huittinen case.

Fetching and storing additional information relating to semantic concepts

Once links to the semantic data is created and checked additional information relating to those concepts will be fetched from external databases and stored to the profile service databases. For example if meaning is expressed as a KOKO concept by utilizing APIs of ONKI service broader, narrower and related concepts will be fetched and stored to the profile database. For example the KOKO concept dances (http://www.yso.fi/onto/koko/p48421) will be expanded to its narrower concepts including different types of dances such as jive, foxtrot and country dances. When for example Elvis Presley (http://rdf.freebase.com/ns/en.elvis_presley) is selected as a favorite singer the related terms to music genre such as rock music, pop music, gospel music will be fetched and stored. This information is used in recommendations.

Similarly, when a user selects her or his interest in the profile service, additional information relating to selected meanings will be automatically fetched and stored to the profile database. For every meaning different language versions of the concept, type of concept and links to similar concepts will be fetched. Depending on what type of concept is additional information can be fetched. For example if a user selects a music artist information relating to music genre, bands and songs will be gathered. For KOKO concepts the hierarchy of the concept and for Geonames location tags geo coordinates and place hierarchy will be fetched. From Freebase and DBpedia additional information relating to music, movies and books is used. Information is stored by using SKOS-ontology relations. In the future intelligence relating to other topics such as sports will be added.

2.5 Recommendation methods

Based on the metadata of content of media services different recommendation methods can be used. Three different recommendation methods have been implemented; semantic recommendation, free-text indexing based recommendation and recommendations that combine both methods. In addition different ranking methods such as place and time for recommendations can be defined based on requirements of cases. The workflow of recommendations can be seen in Figure 11.

The goal is to develop general methods for recommendation which can be expanded by the specific needs of a service.

2.5.1 Semantic recommendations

Semantic recommendations are based on semantic metadata of content and on the semantic metadata enhancement process that is described on the chapter 2.4.4. Semantic recommendations can be based on the given ontology of content providers like in case Sanoma Magazine or on the category of service providers that was enhanced with semantic relations like in case Huittinen. The workflow of semantic recommendations can be seen in Figure 11.

In semantic recommendations both semantics of users' interests profiles as well as semantic metadata of content is utilized. The meanings of user's interests are matched to the content metadata that is enhanced with mappings to Linked Data. The match between user's interest and content can be found as a direct match or based on ontological relations. Depending on the distance of the found match different weights are defined and this information is used for ranking the recommendations.

User's freely given tags that does not have semantic meanings are matched to the labels of the content annotations as well as to the labels of extended meanings and weights are defined in similar way than when analyzing semantic meanings.

When a direct match between ontology used for annotating content and a user's interest is found, relations of the ontology is used for searching additional recommended content.

Weights and relevance of the user's interest, which can be altered on the user interface is used for calculating rank value for recommendations. Same content can be recommended based on different tags of a user and this increases the calculated rank value of the recommendation. Also other criteria can be used for ranking such as publishing date or location.

Metadata such as titles, description and link of content are fetched based on the ranked list and the result of recommendation is shown on a user interface.

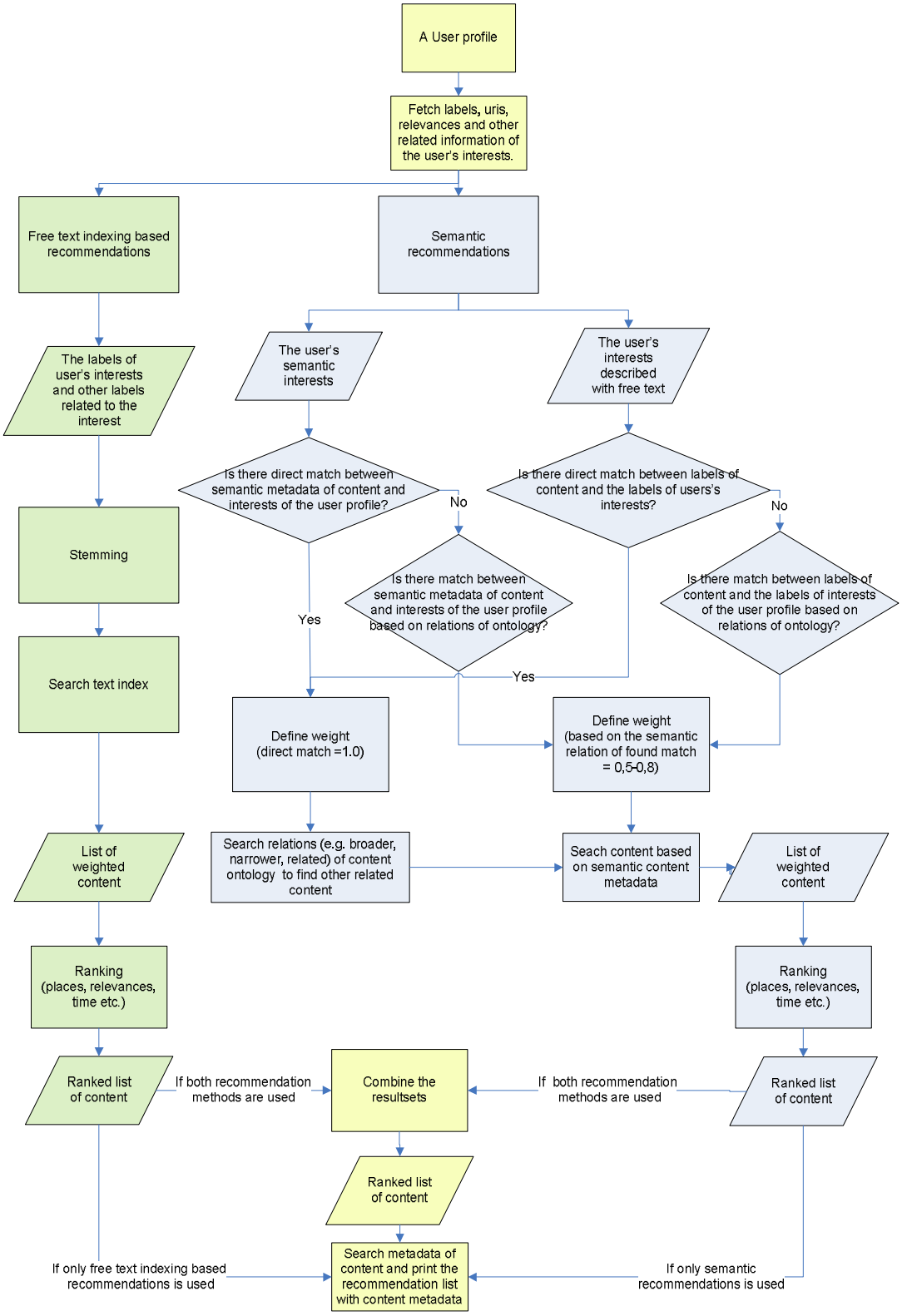

Figure 11 The workflow of recommendations

2.5.2 Free text indexing

Free text indexing based recommendation is based on the Lucene text indexing of the content or content metadata and the Solr-search engine. Terms indicating user's interests on a user's profile are semantically expanded and utilized in free text searching. This means support for multiple languages as well as extending searches with related subjects.

In indexing stemming is used for terms. Relevancies that are obtained from a free text search to the Solr search engine are normalized between 0.0-1.0 and used for ranking recommendation results.

2.5.3 Free text indexing and semantic recommendations

It is possible to combine both semantic and free text indexing based on recommendations (See Figure 11), and benefiting from the strengths of both methods. When comparing the different methods we found that semantic recommendations are more reliable but the challenge is to make an ontology which is used for annotating content extensive enough to respond the different search and usage scenarios of content. The question is on which level content should be annotated. With help of free text indexing different terms of user's interests can be found also in the text, but this causes also false recommendations. By combining both methods findability of content can be increased.

Semantic recommendations and text index based recommendations are combined by calculating normalized relevancies of both recommendation lists together. If recommended content was found only from one list it was added to the recommendations as such. If a recommendation was found as a result of both recommendation methods, rank value was calculated based on the formula: r = max(r1, r2) + min(r1, r2)^2, where r1 is relevance of semantic recommendation, r2 relevance of text based recommendations and r final relevance of recommendation.

Combination of different recommendation methods can also be controlled by changing the linear weight of different recommendation methods.

Calculation methods described above can be altered based on the specific features of different services. The weights between different recommendation methods can be changed on the fly and the algorithm to calculate the rank value can be changed too. Other means to calculate combined rank that were tested are linear (r = r1 + r2) and square-rooted (r = max(r1, r2) + min(r1, r2)^1/2). Both of these boost the lower relevance value more in the range from 0.0 to 1.0 than the squared approach that was used. However, while the squared approach seemed to be the best for these cases there might be situations where alternative methods are suitable.

2.5.4 Ranking

Different filters can be used for ranking the recommendations. Recommendations are based on user's interests, but other type of information can be used for ranking the final result list too. Ranking can be based on different things such as time, place, type or ratings of content or a user's network. What is relevant depends on the service and the metadata of content that is available. In this chapter we share our experiences and observations relating to using location information and portable profiles for recommendations.

Location information in recommendations can be used different ways. When portable profiles are used in a mobile service the current location of a user is important information when ranking recommendations. A user might have indicated in his or her profile also other places that interest them. These places might be often visited places such as home or summer cottage or places where the user has been traveled or is otherwise interested in. So for example a user can be interested things relating to Africa. In recommendations these different use scenarios needs to be taken account. For analyzing and utilizing location information location coordinates as well as hierarchy of places can be used.

When creating a profile the user is able to indicate places of interests with help of the Geonames geographical database and locations in Freebase. Geo coordinates of places are fetched from these external databases and stored into the profile service. Also the place hierarchy of Geonames places are fetched and information is stored into skos:broader, skos:broaderTransitive, skos:narrower and skos:narrowerTransitive relations and this information is used for recommendations. For example if a user is interested in Africa, the system is able to recommend her articles relating to different countries of Africa although the Africa would not be mentioned on the articles themselves.

When using Geonames, there are some restrictions relating to information of Finnish places. Since the beginning of 2010, the concept of province (lääni in Finnish) in Finland don't exist and information in Geonames is not up to date at the moment. Also for Finnish location information Geonames supports quering to which municipality a place belongs to, but does not support querying all the places in one municipality. On the other hand Geonames interface supports findNearby queries which can be used for getting the places of nearby. This information can be used also in recommendations.

Based on the requirements and content of a service different methods can be used for utilizing location information. When thinking about for example event related services developed in the Event management case a user's coordinates will be used for getting recommendations based on the current location of the user. Recommendations can be extended to nearby places within certain radius or based on some polygon [1]. Also the need for determining some impact area of events or news has been detected. This type of information would be useful for recommendations.

Relating to locations both coordinate and hierarchy based recommendations can be used at the same time. Events nearby current location that also indicate the

user's interests can be recommended. The interest can also be a place and referring to earlier examples to be used for example recommending art exhibitions relating to Ethiopia, which is a country in Africa.

2.5.5 Other requirements

A user profile may consist of different parts such as user's interests, social network of friends or contacts, demographic information (such as age, gender, family status), plans and intentions, preferences and behavioral patterns, which have different role and purpose for personalization. The portable profile has focused so far on user's interest profiles, but need for other type of user information has been arisen as well.

The use of intermediary profiles is foreseen potential in different cases. For recommending magazine articles intermediary profiles is foreseen as a potential opportunity to capture and describe users' situations in life and conditions. In the Huittinen case definition of needs of users relating to a certain type of situation would be useful. Examples of such needs are buying a car, moving to a new apartment or planning a vacation.

The profile service also needs features to better support social networks of a user, both for creating them and for using information in recommendations. Features for linking and sharing profiles with other users need to be taken into consideration. Also basic demographic information is important as well. Opportunities of relating different levels of context information with user profiles are described in following chapters.

2.6 Profiles and context

By Timo Laakko, VTT

2.6.1 Different levels of contexts

An efficient model for handling, sharing and storing context information is essential for a context-aware system. The semantic of context information can be used as a basis for adaption and personalization and helps the user to get the relevant content in a relevant format for the current situation.

Further, there is an increasing need for exchange and trade of contextual information between different providers of context-aware services. Context information can be systematically structured by the creation of an ontology that can facilitate the reuse, sharing and communication of context information between different information providers, applications and services.

However, contexts are typically very application-specific. Thus, different applications may require own context types and hierarchies.

In the following figure, an example context hierarchy is shown.

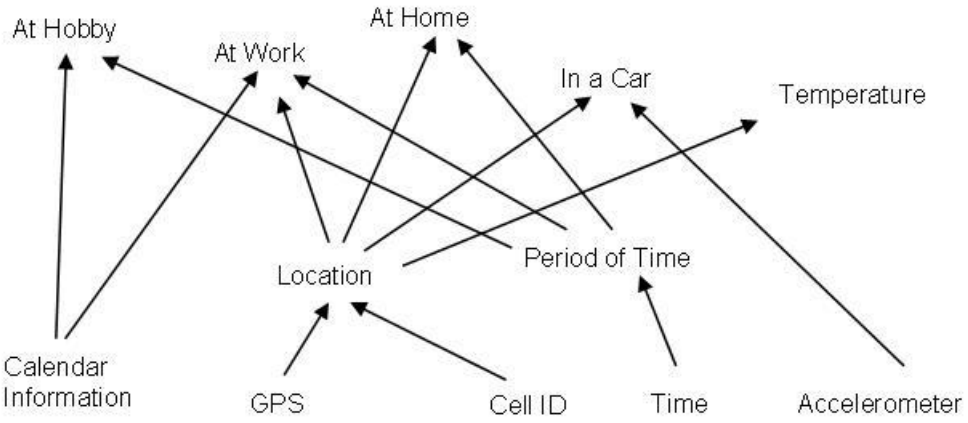

Figure 12 An example context hierarchy.

The above simple context hierarchy builds on lower level contexts that can be automatically recognized in the mobile device (calendar information, GPS, Cell ID, accelerometer data), and lower level contexts that can be automatically recognized in the server (time, weather). These lower level contexts are used in deducing higher level contexts (location, period of time, temperature) and more abstract application level contexts (at a hobby, at home, at work, in a car). Higher level contexts may also be defined manually by the user.

The different levels of contexts are illustrated in the following Figure 13.

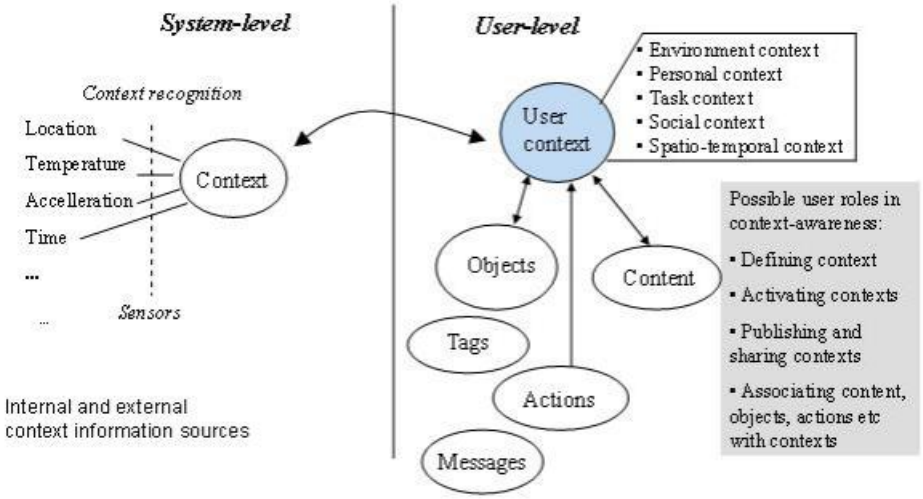

Figure 13 Different levels of contexts.

In general, a low-level context is composed from different sources (sensors, network connection, user preferences, user agent profile etc), for example, measuring location, ambient lighting, temperature, background audio spectrum, color temperature, 3D acceleration, vibration, time, etc. The context information can also be given explicitly.

However, a research challenge is how to derive user-level context information from the low-level context information and sources. A user context may contain parts such as: spatio-temporal (place,time), environment, personal, task, social.

2.6.2 Relating context information with user profiles

Personalization can be based on the stored and semantically refined context information of the user. Therefore, there is first need to sense the user context as an input, and then react to it.

In addition to user profile information, the context information may also be related to content and services (see Figure 14). This can be utilized, e.g., for personalized content selection and delivery, and recommendation systems. The key issues are then how to gather relevant context information and to utilize semantically refined context information of the user. Further, some user preferences may be context-specific. For example, a user may have different interests ”at home” than ”at work”.

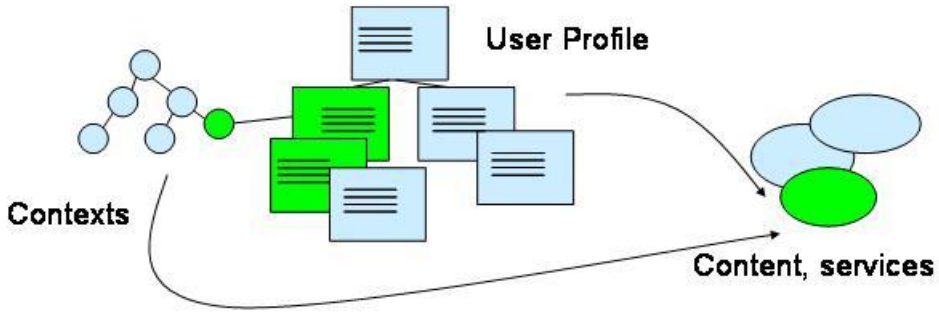

Figure 14 Relating context and user profile information.

Different ontology can be designed to be modular so that one ontology encapsulates one functional element or function. For example, there can be ontology for: devices, users, context data, and services. When generating profiles, information from several ontology can be mapped together as one whole. This is depicted in Figure 15. For example, the user can select a device she wants to use from the device ontology, a set of services she subscribes to from the service ontology and information about time and place from the context ontology.

The user can then incorporate the information together as her contextual profile. Depending on the implementation-specific details, these ontology can be grouped together or stored separately. Functionally they are nevertheless separate units.

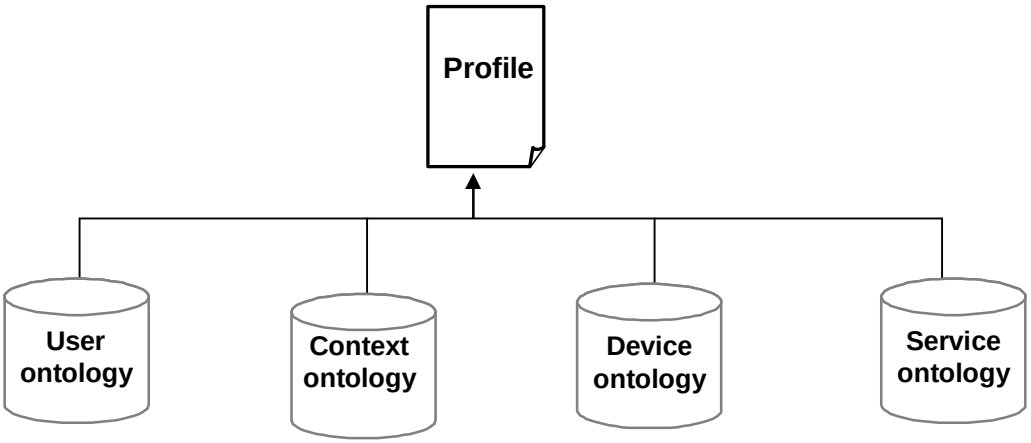

Figure 15 Creating a contextual profile – Context, user, device and service ontology.

Profiles may also form hierarchies just as the contexts. In Figure 16, a hierarchy of contextual profiles are shown. A general profile includes all the common parameters (e.g. interests etc), and it is used as a default profile for the user. The other profiles are context-specific.

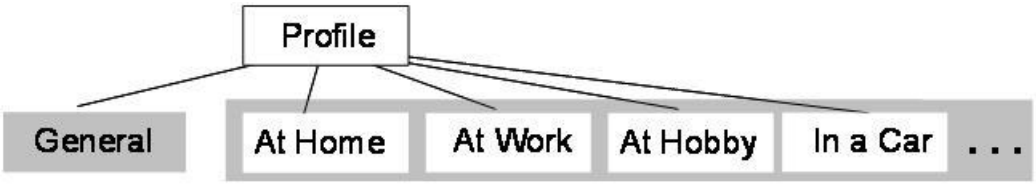

Figure 16 A hierarchy of contextual profiles.

A profile template provides a set of user preferences specific to certain contextual situation, for instance, at home (HomeProfile), at work (WorkProfile), at car (CarProfile). Other parameter values are inherited from the GeneralProfile. The set of templates can be extended. It is a future research challenge to provide mappings of context taxonomies to profile hierarchies; ontology-based approach can also be applied.

There may also be a custom profile, which allows a user to specify new parameters. So, it enables freeform, application specific parameters.

It is a future work item to specify an ontology/semantics so that different application domains can utilize customized profile information.

2.7 Conclusions

In Next Media different methods for utilising a semantic user interest profile in recommendations have been developed. Recommendations can be based on semantics, free-text indexing or a combination of both methods. Different ranking methods based on place, time and other information such as publishing date can be used.

Also methods to support easy integration of portable profiles with service providers' services such as API and semantic mapping methods have been developed. Linked Data is used for creating semantic meanings for the profiles and content. Information available in these external databases is used for producing better recommendations.

The portable profiles have been utilized in different cases such as recommending events in different applications, recommending magazine articles and recommending services.

In 2011 the main focus of the generic research will be on improving the user experience of creating and utilizing the profile. The research question is how users are supported in creating a semantic profile so that it is fun and easy to do and provides detailed information about their interests. User tests in the Event management case will also give valuable feedback for the development of the profile service.

Another important aspect will be how the utilization and updating of the profile can be integrated seamlessly into media services both from the end users' and service providers' point of view. This includes developing methods that help users to keep their profile updated. The aim is to update a user's profile based on a user's behavior in different services where a user has logged in with her profile. Event management and Hyperlocal (Huittinen) cases will also be used as example services when developing and testing the features. The use scenarios will be analysed also from newspapers' and their web services' point of view (hyperlocal case).

Support for intermediate profiles will be developed further in the cases (Huittinen and Sanoma Magazine in WP4). Also recommendation methods will be updated based on the requirements of the cases.

3 Professional Advertisers' Metadata

By Raphael Giesecke, Aalto University School of Science and Technology

3.1 Introduction

Professional advertisers and media agencies both share the same interest: classifying their target audiences. Contextual targeting promises higher return on (advertisement) investment than a "one fits all" approach. This is valid for both mass products and more specific (and often more complex and expensive) customised products.

While demographics and audience behaviour are investigated on a daily (TV), monthly (selected magazines) or six-monthly (news papers) basis, the customisation of advertisements on an online (real time) basis is a new issue for advertisers, still. Thus, while efforts to accompany advertisements with metadata, matching both user metadata and content metadata are common place in, e.g.,

Facebook, Yahoo and Google, Finnish media companies do not actively ask for such comprehensive advertisement metadata.

Instead, media companies have concentrated on metadata linking content (e.g. articles, visuals, audio) to user metadata. See Gylfe 2009 (I + II).

This research should be ideally expanded towards the advertisers and the media agencies, allowing all professional media market participants to share the same type of metadata for a given application (e.g. TV show, magazine article, news article).

In the following, research outcomes are presented based on four interviews, which should allow to scope and define the necessary research better.

3.2 Original Research Questions

The following questions where originally formulated from the media content producer point of view, based on Kivinen et al.'s (2010) Knowledge Architecture.

Topic 1: The role of the advertiser

- How are advertisers integrated into the publishing processes?
- Do they have a clear role description?
- Do they know their tasks?
- Does the media company know their tasks?

Topic 2: The involvement of knowledge mechanisms

- When interacting with advertisers, are the interfaces clearly described – both technically and human to human?
- How do the advertisers describe their target context?
- Which metadata is being exchanged?
- How contexts are linked – users to advertisers?

3.3 Theoretical Foundation

The original research questions are a logical extension of the media publishing processes and their Knowledge Architecture presented by Kivinen et al. (2010) and applied in a case study by KSF Media (Andelin 2010).

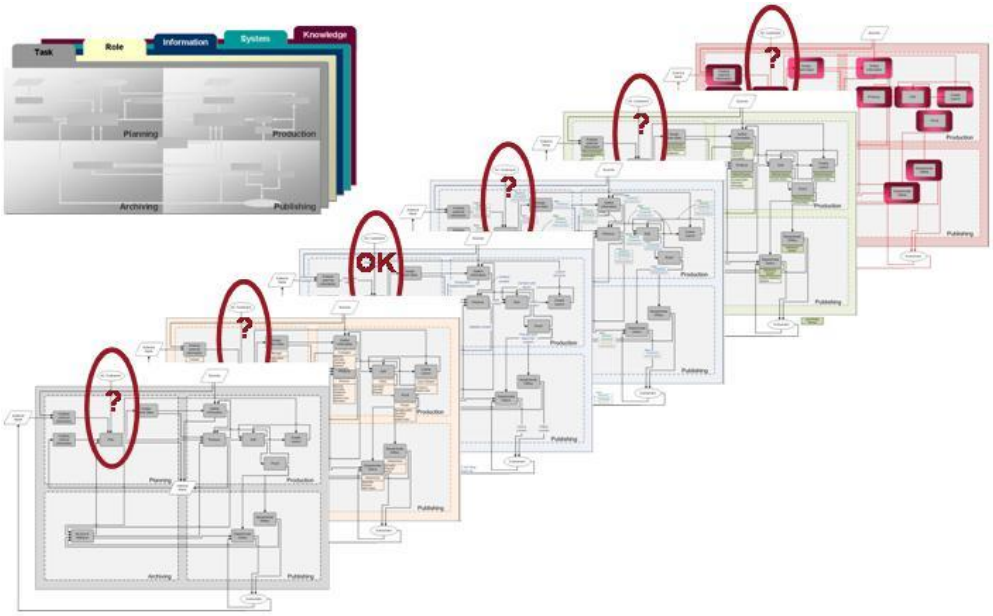

Figure 17 Open questions about advertisers in Knowledge Architecture layers.

The purpose of this work was to find out the complementary view of advertisers, and to contrast their views with the original research questions.

For reasons of simplification, the “dilemma” discussed in the next section was established as a starting point for interview discussion.

3.4 Metadata Dilemma

Especially Facebook has demonstrated its capability to link content metadata, user metadata and advertisement metadata. While the interface to advertisers looks awquard and somewhat naïve, Facebook has managed to create Billions of Euros in advertisement revenue. However, Facebook does not yet provide concrete, detailed feedback to advertisers on their specific audiences.

For a typical media producer (TV, magazines, news papers, radio) the following dilemma develops (see Figure 18): What metadata is available for advertisements? How does it match the media company’s own metadata on content? How can the (two types) of metadata matched to end user metadata (online and offline)? When (at what point in time) shall content be customised (including advertisements)? In how many steps (e.g. first offline, then online)?

This metadata dilemma is made worse by the production ethics in news rooms, which do not allow product placement close to related content. E.g. an article about Baltic sea ferries must not be accompanied by advertisements of Tallink-Silja and Viking line. However, this problem applies less in magazines and much less in (commercial) TV.

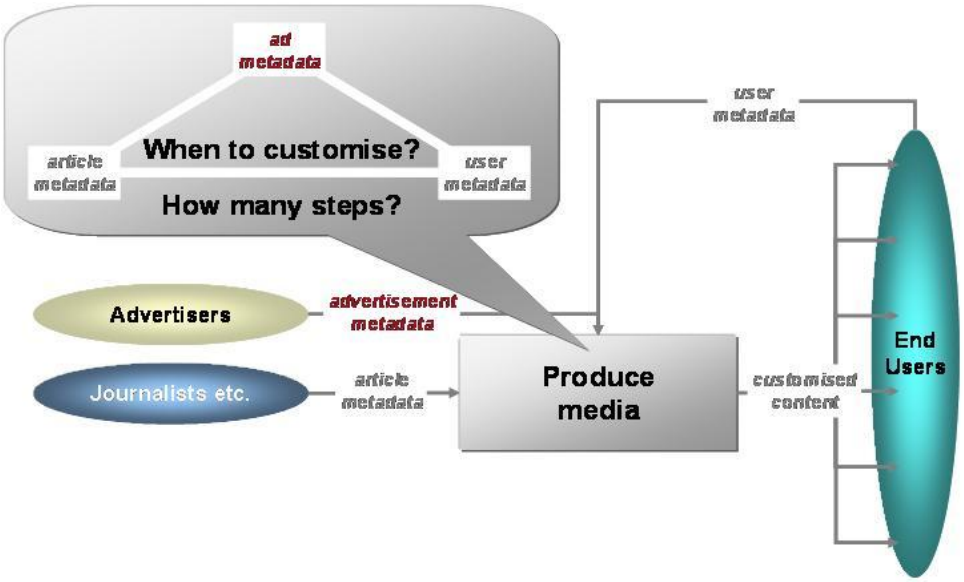

Figure 18 The media producers' metadata dilemma.

3.5 Interview Partners

During the fall 2010 four interviews have been conducted, one within a media house and three with persons of a advertisement/media agency background.

The interview partners where:

- The EvenemaX manager and the development manager at KSF Media
- The leader of the project "The Cooperation with Advertising Agencies" at Aalto University Media Factory
- A *creative* at the most successful advertising agency in German-speaking Europe, measured by creativity and effectiveness
- The Buying Director at Dagmar Oy

The interviews were structured along the original research questions, with a focus on the metadata dilemma as presented above.

3.6 Interview Results

3.6.1 The role of the advertiser

Media and advertisement agencies are gradually changing their role. Digital media have opened some threats but even more opportunities for them. While some even recommend some (test) advertisements in Facebook, others do not. All, however, engage in viral marketing processes.

Still, the standard role, advertisement/product placement is the main stream with most revenues. Basic laws of campaign planning still apply. For long term campaigns, which are prepared months in advance, the respective media agency

will be in contact with publishers, and sometimes even with editors. Note these contacts are not standardised and involve also exchange of tacit knowledge.

Concluding, advertisers (through media agencies) are partially integrated into the publishing processes, but neither in a (semi-) automated way, nor in a way comprehensively addressing the metadata dilemma.

Also the idea of role descriptions was not explicitly mentioned. The tasks of the actors seem to be clear to the actors (who does what, when and why) – however both sides – media houses and agencies, do not collaborate on the changing definition of their roles and tasks. And while the metadata dilemma was immediately understood by all actors, they all agreed that more research on this topic is necessary.

3.6.2 The involvement of knowledge mechanisms

Once the roles are clearly described and understood, concrete interfaces between actors should be described and implemented – both technically and human to human.

The interviewees were generally hesitative when speaking about these interfaces, for various reasons which are mostly related to business intelligence (i.e. confidentiality protection).

Regarding TV, the following “metadata” is being exchanged between advertiser (agency) and (commercial) TV channel:

- programme category
- advertiser
- product category
- product
- audience target group

This metadata is too coarse to be used for digital, online user customisation. Also, TV stations do not usually match product metadata to TV content data – they only match audience target groups (of product and content).

Concluding, metadata interfaces are neither described in a (semi-) automated way, nor in a way comprehensively addressing the metadata dilemma. The contexts between users and advertisers are only linked by the audience target group, but not by joint metadata of both product and (media) content.

3.7 Conclusion

The interviews show that even if all players in the ecosystem know their “standard” professional roles and perform them well, and even if metadata is exchanged to some extent, the Finnish media ecosystem is not sufficiently

prepared for paradigm changes in advertisement as being practiced and refined by global players such as Facebook.

While the offline market will provide very significant main streams of revenues, already today there is a “land slide” towards digital, online and real time advertisement, based on (hyper) local customisation per user (group). The problem here is that media houses on the one side and advertisement agencies on the other address this “land slide” in rather different ways (and also in different work packages in the Next Media programme) and thus seemingly rather diverge than converge concerning shared metadata. Thus the metadata dilemma needs to be researched and solved urgently, preferably within the Next Media programme.

4 HTML 5 in mobile applications

By Petri Vesikivi, Metropolia

4.1 Benefits of browser based applications

Mobile application development was introduced at the same time the first actual smart phones hit the market in early 2000. For first few years the market was dominated by Symbian OS. Java micro edition has been for years the most spread mobile application development environment. Introduction of Black Berry, iPhone and Android have completely changed the landscape and currently there is no lack of choice when it comes to choosing application development environment for mobile.

While mobile application development environments have become more disperse, on the desktops rich internet applications providing user experience equal to standalone application with a browser based application has become more and more common. Many PC applications have a “sister” application that runs on the browser. Even Microsoft has had to launch web versions of their office suite. Gradual introduction of so called HTML5 has enabled a whole new set of applications ranging from standalone games into multithreaded applications. Browsers for mobile devices are based on the same kits as the desktop browsers. Therefore all the features available on desktop have become available also on mobiles. HTML 5 is a loose umbrella term that usually is thought to include also CSS3 and a set of new javascript APIs.

Even if the different browser implementations vary, it is potentially a great platform for developing applications that would run on a wide range of devices. One important benefit of a browser based application is that it doesn’t have to installed, which also means that there is never need to update the application. User has always the latest version at disposal. One of the drawbacks has been that browser based application requires always a network connection, but with introduction of local storage and application cache, this no longer the case: One can develop a browser application, that doesn’t require a constant network access. Local storage with the possibility of multithreading and new user interface

opportunities with Canvas, make HTML5 a very interesting mobile application development environment. It for sure doesn't replace the native application development environments, which can provide tighter integration and effective usage of processing power and other resources.

4.2 HTML5 for mobile devices

4.2.1 Database API

Database API as such is a loose term and may not have a generally accepted definition. In this document it refers to three features that are part of the equally loose term HTML5 namely Application caching, web storage (sometimes also called local storage or DOM storage) and database storage.

Off-line application caching enables the developer to decide which parts of the application are cached at the client and which not. This can be done by including a file with the extension .manifest and defining in the file the cached items. In order for caching to work, the MIME type of files with the .manifest must be set to text/manifest. Once this has been set the cached files are only re-fetched from the server only if the .manifest file has been updated. When testing the off-line application cache we found out that, the manifest file should not have any non-existent elements: if there is even one, it will make the off-line application cache non-functional. Also one should list all scripts including Ajax calls under the NETWORK –heading in the manifest file. Offline application cache enables web applications that work without a network connection however in modern applications the actual data is not with the page, but is dynamically generated using a template engine of some sort. Therefore off-line application requires also the facility to store the actual data also on the device and this is provided by local storage and database storage.

Web storage allows the application to store data locally into the phone with key-value pair structure. On mobile devices the space for local storage is fairly limited a few megabytes, but is likely to become bigger as the devices evolve. Before local storage is used, most browser implementations will ask to user for approval. After the permission is given the data can be stored. Web storage is especially important for mobile devices as it saves the network requests which always take seconds in normal conditions. Writing and reading the local storage is done in a few milliseconds compared to seconds that it would take the transfer the same amount of data over the network. Unlike cookies, the storage data is not sent to server in the headers of the requests and web storage doesn't provide means of traversing the structure without knowing the keys like cookies provide. Furthermore the web storage doesn't have any set expiration time as the cookies have.

Where web storage stores simple key-value pairs, the database storage (called also web SQL database) provides facility to store more complex data structures than key-value pairs. WebKit and Mozilla are using an embedded SQLite engine providing a much more robust way of storing complex data structures than web

storage. The Specification however is not equally robust and implementations in the browsers vary. Furthermore W3C Web Applications Working Group has no intentions to maintain the SQLite specification further. As the event data doesn't have a complex structure, it is merely a list of events with name location and so, use of database storage was not tested as part of this project.

4.2.2 Canvas API

Canvas is a HTML5 element, which is used to create a bitmap graphics element into a web document. The same way as database storage is controversial specification also Canvas is as part of the web developer community believes it doesn't bring any substantial benefits compared to SVG (scalable vector graphics) that can be used to create graphical elements as well. One use case for Canvas would be games requiring bit map graphics. As no immediate benefits were apparent for application like event management, we did not test the feasibility of it.

4.2.3 Geo location API

Geo location API provides the coordinates of the device into javascript. API is agnostic of the underlying means of getting the coordinates even if with mobile phones the most usual source is the GPS embedded into the phone. According to the specification, the browser must ask the user prior to getting the coordinates. All new smart phones include GPS functionality and at least android, iPhone and Symbian browsers support geo location API.

4.2.4 Worker threads

Worker threads provide means for running java scripts in the background. When browser is running on a computer with multicore processor, it is possible to gain performance improvement by splitting processor intensive tasks into several threads. First mobile phone with dual core processor is expected to be available during first half of 2011. Currently web workers, when used on mobile phones, don't really speed up processing, but they can help to keep the user interface responsive.

One possible use case of web worker for mobile device would be implementing the network connectivity into the background thread. By doing this a big chunk of for instance events could be fetched in the background while user is browsing through the already fetched events. Worker thread cannot access the document object model meaning it can't for instance update anything that is shown on the screen for the user. Instead it must send a message to the main thread containing the fetched information and upon receiving the message the main thread will update the document object model accordingly. By implementing the network actions in worker thread the user interface does not freeze i.e. user can continue using it regardless of the network actions. This is of key importance especially in the mobile environment which inevitably introduces a delay with every request and also the transmission of data may take tens of seconds due to retransmissions that might be required before the whole data set is transferred over to the mobile.

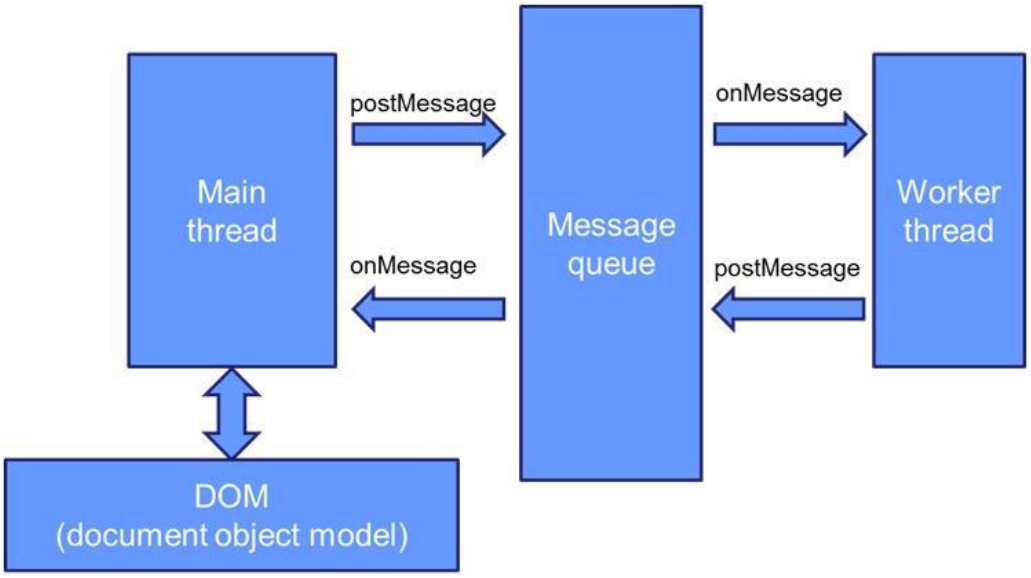

Figure 19 Principles of worker threads.

Unfortunately worker threads are not yet supported on the Safari for iPhone (iOS 3.x) which was the target platform for this project. iOS 4.2 is the first iPhone operating system version, which supports web workers. It also supports server events, which is a mechanism to deliver unsolicited events to the client application.

4.3 Testing local cache vs. network access to event data

4.3.1 Test setup

Tests with 2G and 3G data access were run using iPhone 3Gs with Sonera SIM and WLAN access was tested through Metropolia University of Applied Science's WLAN. Test setup included a proxy server, which converted the data from Evenemax system into JSON format data that was fetched by the browser based application.

Tests were done with three different samples of event data: Vantaa events (around 50 events), Helsinki events (around 1200 events) and all events (around 6300 events). Target was to see if there would be an optimal size chunk for the transfer. In mobile networks there is a considerable delay for each HTPP request going to the network, therefore sending small chunks instead of a bigger chunk introduces more overhead cumulatively. Event data included event titles, event types, link to the event info, location, start time&date and end time&date. One event record was around 270 bytes long (depending on the actual content like length of event name etc.).

4.3.2 Testing event storage and fetch from local storage

4.3.2.1 Local storage tests

Storing events locally were tested with HTC Hero (Android version 2.1) and iPhone 3Gs (iOS version 4.1) phones. iPod Touch/iPhone 3GS (iOS 3.1). Storing and retrieving the test data took the same time with all test devices. For instance loading all events (6300) from the local storage took in average .5 s.

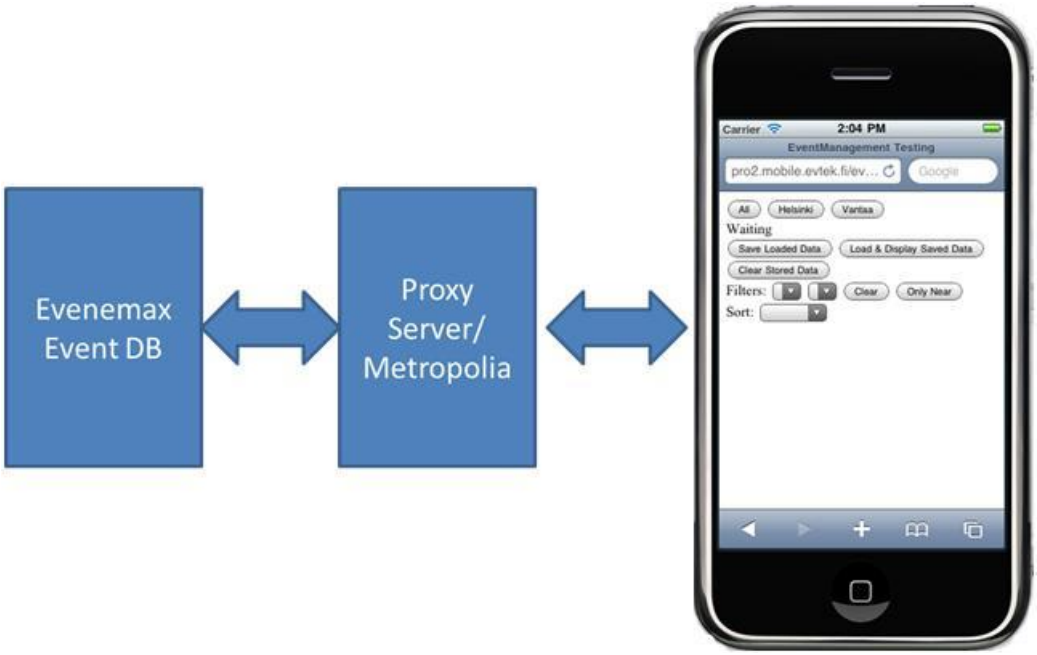

Figure 20 Experiment with local storage.

Space available for storing data varies from 1.7 MB (iPod, iPhone 3GS iOS 3.x) to 8.1 MB (HTC Hero). iPhone 3GS with iOS4 could store up to 2.2 MB data locally. All these devices could store a cache of 1500 events. In near future mobile phones and other networked devices will have more memory available for the browser based applications, which would enable application that cache for instance all events in Finland locally on the device.

4.3.2.2 Event down load tests

Tests were run 50 times for each of the three event chunk sizes. Unfortunately the standard deviation of the measured down load times is extremely large almost equal to the average. This is true for mobile networks, for the tests with PC+LAN the standard deviation was less than 10% of the average. High deviation would actually require a sample size of 1000+ events, which was impractical for the study at hand. Therefore we decided to leave the five longest times out of the results when analysing them.

Time of fetching the event from the proxy server was measured and by dividing the time by the number of events, the down load speed events/s was calculated. By no surprise the lowest speed was with 3G, highest through LAN and WLAN was somewhere between these two.

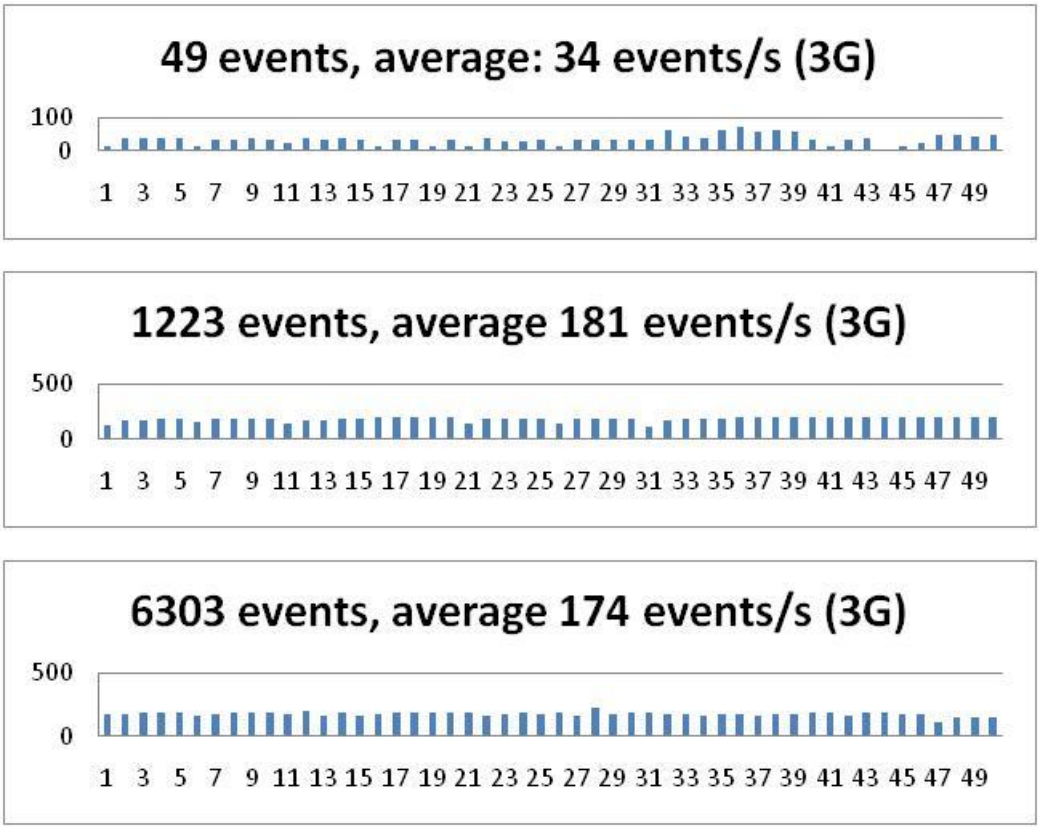

Figure 21 Samples over 3G network.

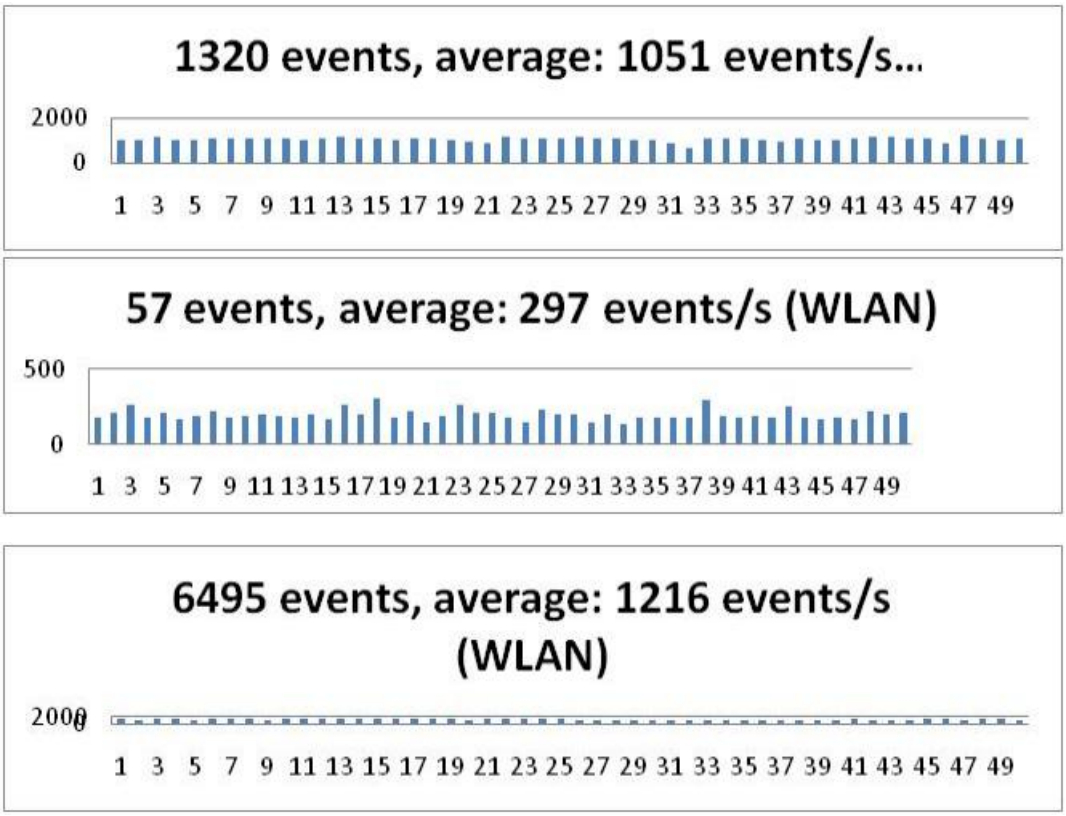

Figure 22 Samples over WLAN.

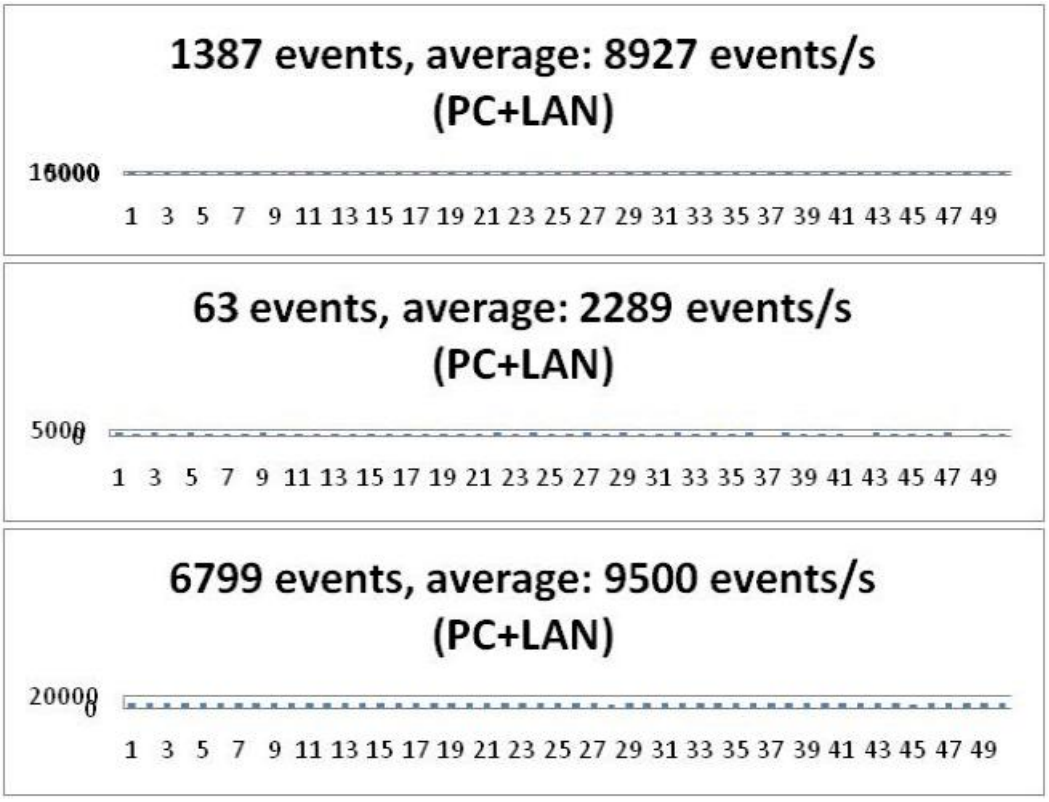

Figure 23 Samples with PC over LAN.

More interesting thing is that there seems to be a certain size of the event chunk after which increasing the size of the event chunk doesn't increase the average down load speed (events/s). Partly this could be because of the inherent overhead in the TCP/IP protocol stack related to setting up the TCP session, but most likely it is related to a transmit buffer size either in Evenemax server or the proxy server used.

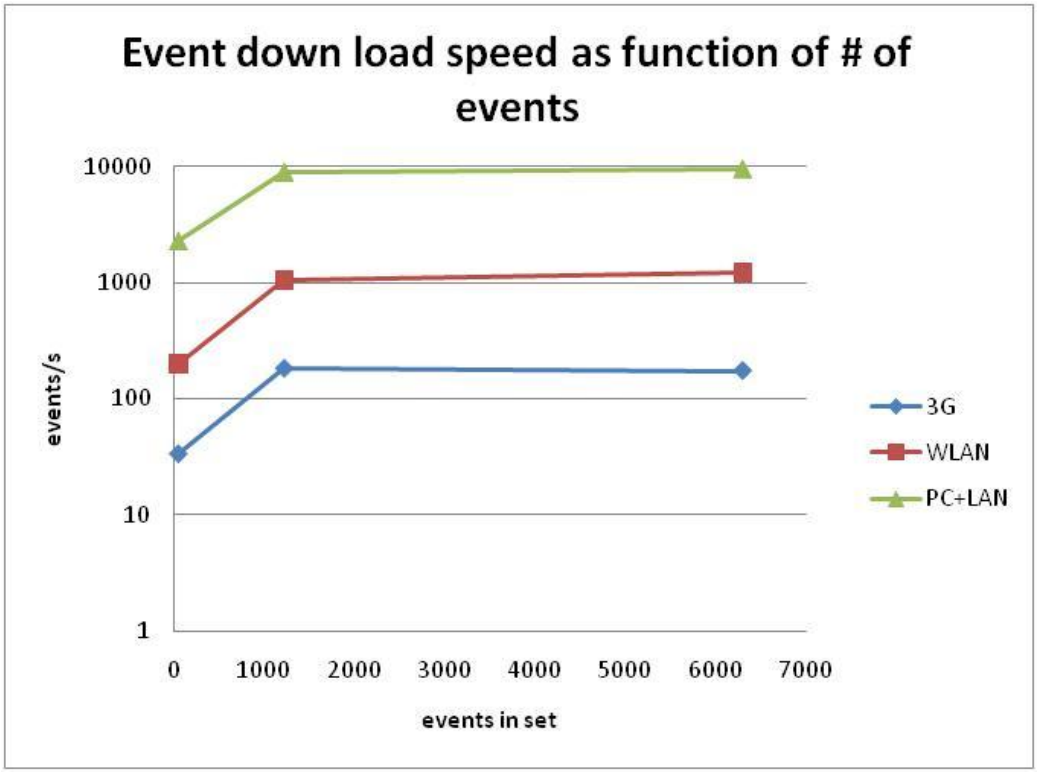

Figure 24 Event down load speed.

4.3.3 Test results

Based on the test, it seems that caching event data locally on the device could improve the user experience considerably as fetch form local storage is immediate (a few milliseconds) with very minimal spread of results. This is truer for mobile environments than the desktop as the download speed for the desktop is in the order of 9000 events/second meaning that all events in the Evenemax database could be downloaded in less than second.

Let's assume that a person would like to search for events in Helsinki area using a mobile device and in order to find the right event would do three queries returning each 50 events. Down loading the three sets would take something like 3*1.5 seconds, but if one is unlucky one download can take 20 seconds. So total time spent on waiting would be in best case 4.5 s and in worst case it can be easily more than 30s. Whole set of the Helsinki events could be down loaded in average of 8.2 seconds. (All figures are assuming a 3G network). Down loading the whole set would enable any number of local searches done in few milliseconds. More over if a caching mechanism managing the local cache would be introduced and

the web workers utilized for updating the event data user would get instant access to all events as for most of the time they would come from device cache instead of remote server.

4.4 Using HTML5 for event data

4.4.1 Off-line access to my events

Off line access was tested in the event management prototype developed in the case. Idea with the application was that information of events the person is participating or plans to participate would be stored locally on the phone and thus provide access to the information anytime regardless of the presence of a cellular data service. Off-line use requires both application cache (which will cache the pages and elements used to compose the pages) and off-line storage (key-value pair store for the event data). Mobile client connects with Conmio's event management proxy that provides the interface towards Profium's event database and VTT's profile and recommendation service. User authentication utilizes openID.

Other views of the client application are generated by Grails, a Groovy framework which is used in this project. As the data for my events comes at least in off-line mode from the local cache the page must be created locally on the device. For this purpose we evaluated java script templating engines and selected Pure, which met our key selection criteria by being nonintrusive and compatible with jquery.

Whenever user selects my events, instead of sending an HTTP get for my events page, the page will be generated through the JavaScript templating engine. This will make the application more responsive, than it the case where another http-request would be generated.

Figure 25 Views of the client application of mobile event application.

Off-line access also requires setting up the application caching so, that all components of my events view are cached locally. For this purpose we created the corresponding .manifest file and changed the MIME –type definitions on the webserver accordingly. Application was also tested with Android 2.1 phone. When application goes from off-line to on-line a document.body.ononline –event will be generated. On the iPhone the event is generated, but on the android phone not. This event is prerequisite for creating a solid application utilizing application caching and on-line storage as it is needed for synchronizing. iPhone iOS 4.2 browser does this, but the Android phone does not.

We also noticed that the Ajax calls when done from application using application caching are problematic. First of all the destinations of Ajax calls must be put under the NETWORK – heading in the manifest meaning that they are always fetched from the network. Secondly if application cache is used, for some odd reason both the success and error call-backs will be called, regardless of the status of the Ajax request. If one wants to store changes into local storage when off-line, it needs to be done by checking the property window.navigator.onLine and depending on the value either write to local storage or create the Ajax request.

4.4.2 Enhanced user experience with canvas

Canvas was not tested, but could be used to make use of the limited screen space available on the device. Canvas provides screen resolution dependent bit maps, which means that different screen resolutions should be catered for in the application. It could provide especially together with the touch screen smoother applications, however as we are not really developing a game, it is unlikely to be worth the effort and therefore was not actually tried out in the first phase of this project.

4.4.3 Location aware event search

Event search was made to location aware by utilising the geo location API. Instead of using the API directly we decided to use geolocation.js, which provides a device independent layer on top of the geo location API. When user navigates to the event management page geo location API is used to get the coordinates. Coordinates are passed to the proxy server using an Ajax request. One challenge with this approach is that some browser implementations like Safari are every time asking the user if getting location is allowed. This happens if user navigates to another view and back which gets irritating after a while as it is unlikely that the location would have actually changed.

4.4.4 Improved responsiveness of user interface

Event viewer application would benefit from web workers as fetching events from back end could be implemented in a back ground thread. Desktop browser support web workers, but currently the mobile browsers don't support it.

4.4.5 Dynamic filters for events

One feature of the event viewer application is to provide multiple filters for the user. We studied the possibility to make the filter generation completely dynamic, so that the application would automatically create a drop down list for any tag that appears in the data set.

Figure 26 Examples of filters of event viewer application.

In figure above (contains screen shots of the research prototype application) on the left hand side the user has pressed the filters drop down and the list shown is automatically generated from the tags the html page has. When user selects a

value, in this case it is the area, all values of the area that exist in JSON structure are used to generate a dynamic selection list.

If we would assume the event data and the parameters of it would be constantly changing, such dynamic creation of filters and possible values would make sense. The drawback is that the data includes also tags, which are not useful as filters from the end user point of view. This actually rules out automatic filter creation: if one anyway needs to write code to remove some filters with almost the same effort one can hard code the filters as well. Even if the filter fields would be hard coded, fetching the possible values would still make sense and is a useful feature. In the current implementation of the actual event viewer prototype events are filtered on the server side, but in the future it could make sense to consider the option of getting a bigger chunk of events, that than would be filtered locally. In this way all searches and filtering would be done locally on the device and the delay would be in order of milliseconds as compared with seconds as it would be if searches would be done over the network.

4.5 HTML5 and local profile

One opportunity of using HTML5 features would be to store the profile of the person on the device instead of the network. This was not tested as it would actually require the architecture of the event management system to be developed in a way that would enable distributed storage of profiles and distribution of event information as well.

Use flow:

- user creates an interest profile either on the mobile or in the network
- profile is stored in the device
- When user fetches events the stored profile will be used.
    - Two options: 1) profile is sent to the server, that returns events or 2) application fetches a set of events, which are filtered on the mobile device based on the interest profile.

Figure below describes the option where event basic data (only text info) is stored on the mobile device (the big blue box). This solution would minimize the network requests between event database and the client. When user registers to the service, he or she would get the complete list of events for Finland. This operation would happen on the background. Once the list of events have been downloaded only new events would be down loaded once they appear in event database. Feasibility of this depends on the memory required to store the complete list of events and the local storage available on the browser. The current browser implementations can store around 7000 events, if the data includes the very basic information like name, venue, coordinates, date and time. One other possible caveat is that the mobile phone possibly wouldn’t have the computing resource that the recommendation engine requires.

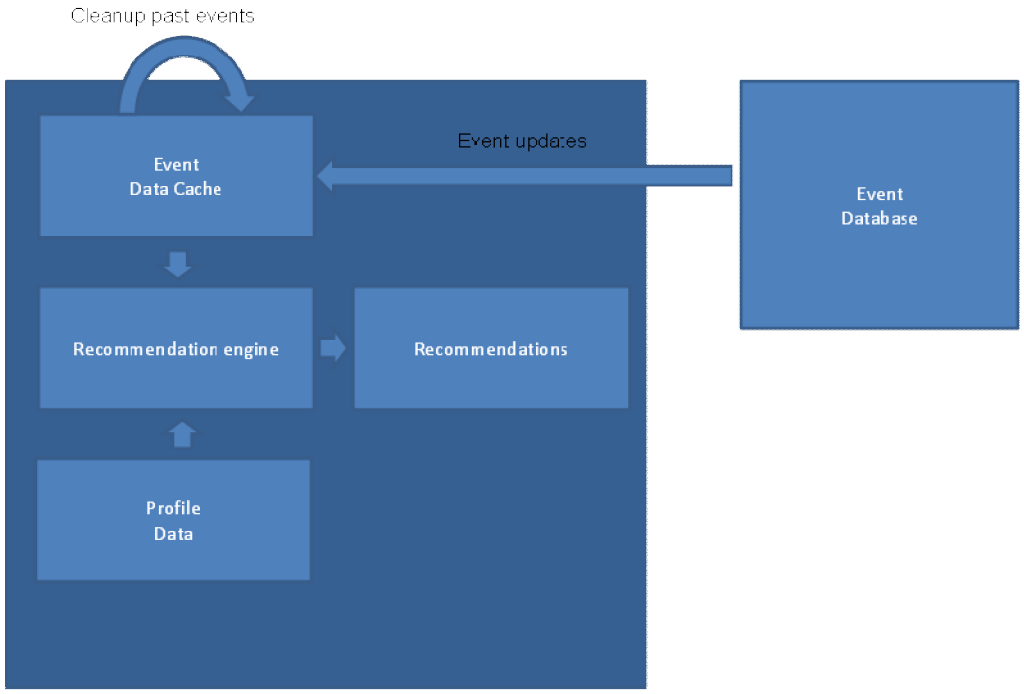

Figure 27 The option where event basic data is stored on the mobile device (the big blue box).

The other optional way would be to store locally only the profile data and my events cache. Picture below describes this option. In this option only the profile would be stored locally, but all requests for events would be sent to recommendation engine in the network cloud.

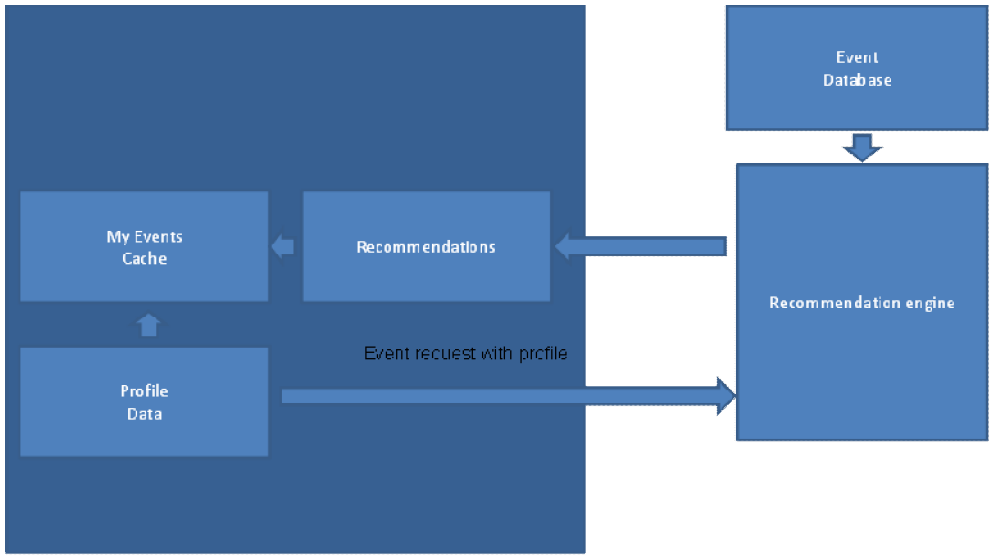

Figure 28 The option where only the profile data and my events cache are stored locally.

In both cases the profile would be stored only on the device, which would considerably improve the privacy of the profile as it would be stored on a device that is completely controlled by the user. Down side would be that the profile or parts of the profile should always be sent to the recommendation engine on the network. As processing power on mobile devices is constantly increasing – for instance first dual core processor phone will come in first half of 2011 – eventually one option would be to run the recommendation engine locally as well.

One could argue that at the same time processing capabilities of mobile phones improve, also the mobile networks will continue improving and running recommendations locally wouldn't make sense. Even if networks are improving (HSDPA, LTE etc.), the protocols used on network and transport layers are still the same that were used in the 80's meaning that the inherent delay in setting up the transmission control protocol session and all the overhead in the protocols will remain the same.

Users expect shorter delays for mobile applications than desktop application, because mobile phone has been and still is very responsive: if you press the call button, setting up the call starts immediately without a delay. The same thing is expected from other applications running on the phone. A network request always creates a delay of a random length and hampers the responsiveness of the application. In order to avoid this, the application should be done in a way where the data is cached on the device and all network operations are done with web workers i.e. background threads. This kind of application architecture can give the user immediate responses to all event searches as in the future the network capacity and the capabilities of the mobile device (increasing memory, improving processing power) allow massive amounts of data transferred, processed and stored on the mobile device.

5 Conclusions

In Next Media different methods for utilising a semantic user interest profile in recommendations have been developed. Also methods to support easy integration of portable profiles with service providers' services have been developed. API for creating, reading, updating and deleting a profile as well as for getting recommendations has been implemented. Portable profiles have been used in Event Management case and also for developing recommendations for Huittinen case.

Besides linking metadata of media content to user metadata it is important to be able to link advertiser's metadata to user metadata. Contextual targeting promises higher return on (advertisement) investment than a "one fits all" approach. Important research question is how advertiser's metadata is described and how it can be exchanged between advertisers and content / service producers.

HTML 5 offers new features for creating advanced browser based mobile applications. Local storage with the possibility of multithreading and new user interface opportunities with Canvas, make HTML5 a very interesting mobile application development environment.

References

Alanko, T., Vainikainen, S., Bäck, A. (2010) Service-independent user profiles as the source of personalization: Personalization in an Article Service.
http://www.flexibleservices.fi/files/file/pdf/crossmedia/CrossMedia_D1.7_Service-independent_user_profiles-Personalization_in_an_Article_Service.pdf

Andelin, J-E. (2010) Adaptation of Knowledge Architecture
http://www.flexibleservices.fi/files/file/pdf/crossmedia/CrossMedia_D1.7_Adaptation_of_Knowledge_Architecture.pdf

Andelin, J-E, Bäck Asta. (2010) Service-independent user profiles as the source of personalization: Personalization in event recommendations

Bäck, A. (2010) Service-independent user profiles as the source of personalization: Experiences.
http://www.flexibleservices.fi/files/file/pdf/crossmedia/CrossMedia_D1.7_Service-independent_user_profiles-Experiences.pdf

Bäck, A., Vainikainen, S., Saarela, J., (2009) Knowledge-based services embedded in editorial systems.
http://www.flexibleservices.fi/files/file/Liitteet/FS_CrossMedia_D1.5_v_1.0.pdf

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

Gylfe, Chip et al. (2009) Inventory of the Types of Metadata in Editorial Operations
http://www.flexibleservices.fi/files/file/Liitteet/FS_D1.1_v1.00.pdf

Gylfe C. (2009) Metadata i Cross Media Redaktionsprocesser. TKK/Elektroniikan, tietoliikenteen ja automaation tiedekunta
http://owww.media.hut.fi/julkaisut/diplomityot/4303.html

jQuery documentation
http://docs.jquery.com/

Kivinen, H., Immonen, S., Giesecke, R. (2010) Knowledge Architecture
http://www.flexibleservices.fi/files/file/pdf/crossmedia/CrossMedia_D1.3_V10_Knowledge_Architecture.pdf

Nummiaho, A., Vainikainen S., Melin, M. (2010). Utilizing Linked Open Data Sources for Automatic Generation of Semantic Metadata. Metadata and Semantic Research 4th International Conference, MTSR 2010, Alcalá de Henares, Spain, October 20-22, 2010. Proceedings. Metadata and Semantic Research,Communications in Computer and Information Science, 2010, Volume 108, 78-83, DOI: 10.1007/978-3-642-16552-8_8.
http://www.springerlink.com/content/v43t108647640654/

pure.js documentation
http://beebole.com/pure/documentation/

W3C HTML5 working draft specification
http://www.w3.org/TR/html5/

W3C Working Group Note 18 November 2010
http://www.w3.org/TR/webdatabase/