

Location-Aware Information Services using User Profile Matching

Alisa Devlić

University of Zagreb, FER
Unska 3, Zagreb, Croatia (HR-10000)
Email: alisa.devlic@fer.hr

Gordan Ježić

University of Zagreb, FER
Unska 3, Zagreb, Croatia (HR-10000)
Email: gordan.jezic@fer.hr

Abstract—User profiles are used to integrate contextual information about mobile users and objects/devices in their environment. They represent structured knowledge used by a system to provide relevant information set to the user. Location-aware information services have to adapt their functionality to a constantly changing user location and preferences. The adaptation is based on a content ontology, that contains concepts for describing user personal information, device characteristics, presence information and location-based subscription preferences. User profiles are generated for every user's applied terminal and matched to enable adaptation and dissemination of location-aware content. The proposed architecture is sketched and the initial implementation prototype of the system is described in the article.

I. INTRODUCTION

The idea of context-aware computing came from the growing number of wireless and mobile devices used in the user everyday life and the user need to be served with timely delivered information anyplace, anytime and on any device, while on the move.

Users want to access the same content and services on various devices, e.g. laptops, handheld devices, mobile phones, that have different processing capabilities, memory capacity, screen size and support different media formats. Users may also wish to change the terminal during the session and the service task is to recognize the user and enable him a smooth transition to the other device and the continuity of the service use.

The mobile context of the user should be tracked and the awareness of own context is important for applications to be able to adapt their behavior according to the present situation and the conditions in the environment. This way a new set of information services, called context-aware information services, will become possible on mobile devices.

User location is one of the main drivers for the context-aware information. Location-aware information services are services that provide the user with the information set that is related to his current position. Because user position can change between the requests for information, different information sets can be generated for different user positions. There are many positioning methods that can locate the user terminal, such as GPS, A-GPS, Cell ID+TA, E-OTD. The problem that arises is how to aggregate the various forms of location information that these methods produce and give it the user-understandable

meaning. In this article we define the term landmark to denote a located user position with the given location name and propose a way to integrate different forms of location information to be used in location-aware information services.

In addition we propose a way to integrate the knowledge about the user context and describe the contextual information in the user profile form. The proposed profile consists of attributes concerning characteristics of the user's currently used terminal, his location and presence information, and subscription preferences. We assume that the user may prefer to receive the desired content either on the current location or some other location of interest (defined landmark). For that reason we define two kinds of location-based subscriptions: *landmark-based* and *current location-based subscription*.

The information from the user profile is used to adapt and disseminate the content to the user currently used terminal. For the implementation purposes we have developed an initial prototype built on the proposed architecture consisting of the client application that interacts with distributed server components through the web application, having the following functionalities: profile administration, location management, and content adaptation and dissemination.

II. CONTEXT DEFINITION AND CLASSIFICATION

Mobile user context is defined [1] as "*any information that can be used to characterize the situation of the entity, where an entity can be a person, place, physical or computational object*".

Schilit et al. [2] claim that important aspects of the context answer the following questions *where are you, who are you with, and what resources are near you*. Therefore the context should be classified into three categories describing the information related to:

- computing environment, e.g. available processors, device capabilities, network capacity, connectivity and computing cost
- user environment described with his location information, social situation and activity, willingness to communicate, preferences etc.
- physical environment referring to conditions in the environment, such as lighting, noise level, temperature etc.

The starting point in a definition of context is to identify the purpose and elements of the context we are interested to describe. In our case, the purpose is to provide relevant content to the user based on his currently used terminal capabilities, subscription preferences, his current location and presence information.

A. Motivation for RDF

User can make requests for information that may not be relevant to him at a certain time, but will be when a particular event occurs, such as reaching a location of interest. If the location relevance of the information, together with the location's coordinates and the information validity period were structured in a metadata format, it would be possible to automatically retrieve the information relevant to the current context of the user. To this extent, location-aware information services could be seen as context-aware services, because a location change event could, for example, trigger an action of searching and disseminating the appropriate information set to the user. The triggering could be more personalized to the user's needs if it would depend on several context parameters. The example is the user's availability for communication, as the user may not wish to be disturbed with the receiving of information, as he may set his availability for communication to 'not available'. These dependencies could be applied to all metadata elements, if they were represented in a well structured, common data format, so the server could match them and produce the appropriate information set.

The Resource Description Framework (RDF) [3], developed by World-Wide Web Consortium (W3C), provides a framework for describing and exchanging metadata on the Web. It allows descriptions of Web resources to be made publicly available in a machine understandable form through the Uniform Resource Identifier (URI). With the defined semantics services can develop processing mechanisms to exchange information between applications.

The RDF basic concept is built upon describing resources through a collection of properties, called RDF Description, each of them consisting of property type and value (Fig. 1). RDF uses XML model and syntax to describe resources. It utilizes the XML Namespace facility, which points to an URI, to uniquely identify set of properties, commonly called schema. Multiple namespaces can be used to provide properties in a single rdf document.

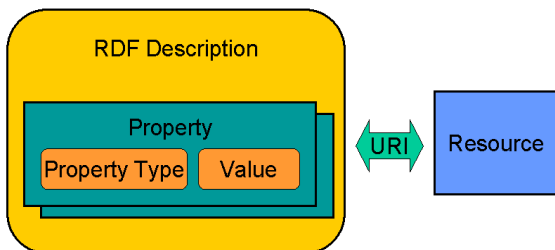


Fig. 1. RDF model

RDF doesn't have a built-in mechanism for defining properties and describing relationships between resources and properties. That is the task of the RDF Schema. Resources are defined as instances of one or more classes that can contain properties and they are described with the hierarchy of classes and subclasses. Properties are defined with domain and range of classes. The domain of the property refers to the class the property belongs to, and it is not restricted to a single class. The range represents the type of values the property can assume, and that can be simple data types or further classes having its own properties.

RDF document consists of statements, combinations of a resource, a property and a value assigned to that property. RDF statements are written in triples: subject, predicate and object. The relationship of RDF Schema with RDF document is as follows: classes are mapped to subjects and predicates to properties. RDF statements present the main idea of creating semantics and describing relationships between objects, while the RDF Schema contains interpretations of information given in the statement.

III. USER PROFILE CREATION

The ontology used to describe profile structure resembles the structure of the CC/PP schema [4].

The user profile schema consists of the following components: User Info, Terminal Capabilities, Presence and Subscription with the properties characterizing them (Fig. 2). Properties can contain static and dynamic values which can be supplied either automatically (by the system) or can be input directly by the user. In our profile ontology we tried to reuse as much ontologies and specifications as we can rather than reinventing them again. At the moment we have utilized some device-related properties from the WAP UAPProf scheme [5] to characterize the device capabilities.

A. User personal information

User personal data represent the static contextual information in the user profile structure. They describe the aspects of the system that usually do not change with time, for example: the username, user's e-mail address, mobile phone number, etc. The user can also specify the preferred contact means with priority numbers to emphasize the

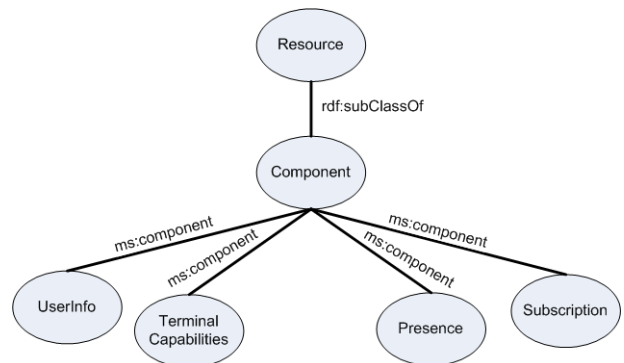


Fig. 2. RDF graph

way he would like to receive information on the applied device (the delivery method). Our ontology introduces a component called *UserInfo* consisting of the simple properties such as *username*, *password*, *email*, *msisdn*, as well as complex ones, such as *contacts*, pointing to the class *PreferredContacts*, which in turn has properties for describing contact, such as *contactName* and *contactPriority*. These examples are serialized as follows:

```
<rdf:Description rdf:ID="UserInfo">
  <rdf:type rdf:resource="&rdfs;Class"/>
  <rdfs:subClassOf rdf:resource="&rdfs;Resource"/>
</rdf:Description>

<rdf:Description rdf:about="username">
  <rdf:type rdf:resource="&rdfs;Property"/>
  <rdfs:domain rdf:resource="#UserInfo"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</rdf:Description>

<rdf:Description rdf:about="password">
  <rdf:type rdf:resource="&rdfs;Property"/>
  <rdfs:domain rdf:resource="#UserInfo"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</rdf:Description>

<rdf:Description rdf:about="e_mail">
  <rdf:type rdf:resource="&rdfs;Property"/>
  <rdfs:domain rdf:resource="#UserInfo"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</rdf:Description>

<rdf:Description rdf:about="msisdn">
  <rdf:type rdf:resource="&rdfs;Property"/>
  <rdfs:domain rdf:resource="#UserInfo"/>
  <rdfs:range rdf:resource="&xsd:positiveInteger"/>
</rdf:Description>

...

```

B. Device characteristics

Device capabilities data introduce the detailed information about user terminal hardware, software and network characteristics when accessing services or some other content. Our ontology defines component called *TerminalCapabilities* with the simple property *clientType* and reuses properties defined in WAP UAProf scheme [5], such as: *Vendor* and *Model* for describing hardware, and *javaPlatform* for describing software characteristics. Client type describes the type of the terminal, e.g. mobile phone, PDA or PC.

```
<rdf:Description rdf:about=
"http://www.example.org/UserProfile#TerminalCapabilities">
  <ms:clientType>mobile_phone</ms:clientType>
  <uaprof:vendor>Nokia</uaprof:vendor>
  <uaprof:model>6600</uaprof:model>
  <ms:javaPlatform>
    <rdf:Bag>
      <rdf:li>CLDC/1.0</rdf:li>
      <rdf:li>MIDP/2.0</rdf:li>
    </rdf:Bag>
  </ms:javaPlatform>
</rdf:Description>

```

The declaration of namespaces used in the user profile document is following:

```
<rdf:RDF xmlns:rdf=
"http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:uaprof=
"http://www.wapforum.org/profiles/UAPROF/"
ccpsschema-20000405#"
xmlns:ms=
"http://stella.zavod.tel.fer.hr:8080/RDF/myScheme.rdf#">

```

This component can be further extended with other properties specified in WAP UAProf ontology to express

information about displaying images, audio or video content, browser user agent or wap capabilities. Our intention was to provide minimum system information capable to discover user currently used device in order to ensure the user the continuity of the service.

C. Location-based subscription

Information services require some user-defined preferences for the type of information that users want to receive. In the context of location-aware information services we define the location-based subscription as a set of topics used to publish and deliver content to interested parties, together with the list of locations where the user would prefer to receive the content. Here we use landmark-based subscription to denote that the user can choose to receive his favorite content only when he is located on the selected landmark. A landmark consists of the geographic coordinate (longitude, latitude) and the name of the location. The following RDF serialization contains definition of location-based subscription component, as well as its properties.

```
<rdf:Description rdf:about="Subscription">
  <rdf:type rdf:resource="&rdfs;Class"/>
  <rdfs:subClassOf rdf:resource="&rdfs;Resource"/>
</rdf:Description>

<rdf:Description rdf:about="topic">
  <rdf:type rdf:resource="&rdfs;Property"/>
  <rdfs:domain rdf:resource="#Subscription"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</rdf:Description>

<rdf:Description rdf:about="landmark">
  <rdf:type rdf:resource="&rdfs;Property"/>
  <rdfs:domain rdf:resource="#Subscription"/>
  <rdfs:domain rdf:resource="#Presence"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</rdf:Description>

```

D. Presence information

Presence information is introduced to describe user current state and activity in order to assist the application in providing relevant content to the user and presenting it on the target terminal according to his current communication status. For example, if the user is currently in a meeting, and his presence status is "busy", the content delivery service will not send him the time-sensitive information. The presence attributes are reused from the Wireless Village Initiative [6] and contain the following information: *the user terminal status (busy / idle / detached)*, his current *landmark (Home / Work / Shopping / Recreation)* and information about his *availability for communication (available / not available / discreet)*. The presence component use is demonstrated on the following example:

```
<rdf:Description rdf:about=
"http://www.example.org/UserProfile#Presence">
  <ms:landmark>Work</ms:landmark>
  <ms:onlineStatus>busy</ms:onlineStatus>
  <ms:userAvailability>available</ms:userAvailability>
</rdf:Description>

```

The presence properties are dynamic, meaning that their values change in time. They are collected from different parts of the system, such as location and status server, as well as the user device, and their change will

trigger the update of the presence information in the user profile.

IV. DEFINITION OF LOCATION SEMANTICS

There are many positioning methods that can locate the user terminal and determine user position. It is expected that in the near future the most devices will be equipped with position sensing technology, e.g. GPS, GSM, Bluetooth, WLAN, RFID. Therefore the adaptivity - the ability of various resources to interoperate under changeable connection conditions becomes an important issue in providing location based services in mobile environments [7].

There are multiple resources of location information that can produce various forms of user location information, such as:

- geographic coordinates in the form of longitude, latitude pair using positioning system;
- WLAN/Bluetooth access point address in the indoor environment;
- MSISDN as the user terminal identification in mobile/fixed telecommunication network;
- IP address as the user trace in the local area network;
- URL as the homepage of the Internet site where the user was registered.

In order to manipulate the given location information, the appropriate user-understandable semantics has to be assigned. The semantic interpretation of location should integrate all these forms in the generic presentation format and store it in the location repository database, from which it could be used as the foundation for building more comprehensive location-aware services.

Our proposed definition of location semantics gathers the specified interpretations of user location information, so when the user would for example access Internet site of the Faculty of Electrical Engineering and Computing, its geographic location would automatically be retrieved and he would be offered with content and services available on that location:

```
<rdf:Description rdf:about=
"http://www.example.org/UserProfile#Location">
  <ms:landmark>FER</ms:landmark>
  <ms:homepage>http://www.fer.hr</ms:homepage>
  <ms:IPaddress>
    <rdf:Bag>
      <rdf:li>161.53.17.*</rdf:li>
      <rdf:li>161.53.19.*</rdf:li>
    </rdf:Bag>
  </ms:IPaddress>
  <ms:coordinates rdf:resource=
"http://www.example.org/UserProfile#Coordinates"/>
  <ms:msisdn>+38516129999</ms:msisdn>
  <ms:wlanAP>192.168.2.19</ms:wlanAP>
  <ms:bluetoothAP>192.168.1.240</ms:bluetoothAP>
</rdf:Description>

<rdf:Description rdf:about=
"http://www.example.org/UserProfile#Coordinates">
  <ms:longitude>N451083</ms:longitude>
  <ms:latitude>E153005</ms:latitude>
</rdf:Description>
```

V. PROPOSED SYSTEM ARCHITECTURE

The following functional requirements have guided the design of the system architecture:

- **User profile generation:** administrator should generate the user profile for every user and his applied device, based on his personal data, device capabilities and preferences to receive location-dependent content based on preferred delivery method (SMS/MMS/e-mail), availability for communication and the name of the topic.
- **Landmark declaration:** user can declare his current position as a landmark giving it a semantic meaning (e.g. Home, Work, Shopping, Recreation).
- **Location-dependent content:** user can create content related to his current location or to one of the specified landmarks, with a possibility to define a validity period for the contained information. The system should provide the user with a mechanism to disseminate this information to other interested users, and to retrieve the location-dependent content (published by other users) on the targeted device, when he is located at the specified location.
- **Preview of most frequently visited landmarks:** user can view his most frequently visited landmarks.
- **View map:** user can retrieve a part of the city map displaying his current location area.
- **User preferences modification:** user can modify his initial settings stored in the profile: the availability for communication, the list of preferred contacts and priorities, and the preferences to receive different information set at the specified location.
- **Terminal heterogeneity:** user can access the application from different terminals, and the system should recognize the same user with different terminal and enable him to continue with the application use.

The proposed system architecture (Fig. 3) is built around two distributed components: profile generation and maintenance subsystem and a middleware.

Profile generation and maintenance subsystem provides the interface to the user equipment and other system components through the middleware, in order to access and modify information stored in the profile and location repository. On the other hand, it maintains up-to-date records with time sensitive information and provides the functionality of creating and administrating profiles and user defined landmarks in the profile and location repository, respectively.

Middleware (Fig. 4) is responsible for receiving HTTP requests from the user equipment, routing them to the appropriate component handler, that will process the request to the target component and return the received result to the user. Middleware implementation is briefly described in the following section.

Two actors that interact with the system are identified: an administrator and a user. An administrator can access the profile generation and maintenance subsystem through the web form. User can access the application using the user equipment (UE), that communicates with other system components through the middleware.

User equipment (UE) can be tracked from various positioning resources, such as the WLAN and Bluetooth

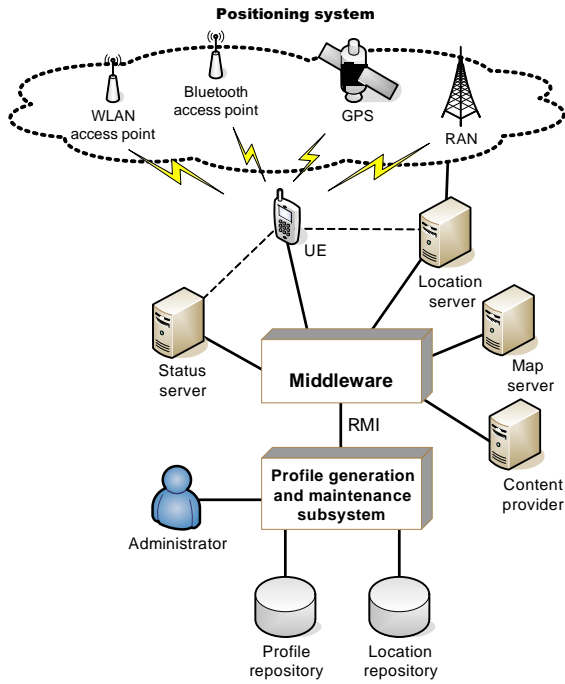


Fig. 3. System architecture

access points in indoor, as well as from GPS and Radio Access Networks (RANs) in outdoor environments (Fig. 3). Depending on the utilized positioning method, the application installed on the UE can be designed to periodically report its position to the location server (if the positioning is terminal-based), or the positioning system can be configured to periodically report the user position to the location server (in the case of network-based positioning). The system should enable the user to find out his own location as well as the location of other entities by initiating the request to the location server. It is assumed that in the worst case scenario at least one system should function properly. Different forms of location information generated from the positioning systems are collected and integrated into the generic presentation format (as described in previous section) and stored with the user-defined landmark name in the location repository.

Status server tracks changes in the user equipment status (busy / idle / detached) and reports this information on the request to the middleware. User can set/modify his availability information anytime using the application.

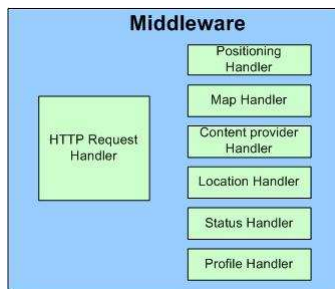


Fig. 4. Middleware

Should any information become available (from the UE or the location server), the system will update the user profile with this information in profile repository.

Map server loads the map of the city and dynamically crops the region centered around the user position with the dimensions of user terminal's screen.

Content provider is presented by a publish/subscribe component that receives the users' subscriptions for content and delivers it to them when the appropriate information set is published. The delivery method is specified for every user's applied terminal in the user profile. Subscriptions can be location-based or non-location-based. If the subscription is location-based, the component will first check if there are users located in the specified area, and if true, deliver the content.

VI. INITIAL IMPLEMENTATION

The initial implementation prototype consists of profile generation and maintenance subsystem, middleware and client application.

Profile generation and maintenance subsystem is implemented as a web application that communicates with the profile and location repository through the JDBC API. Profile repository (table I) contains the records described with the username, terminal and profile column. We assumed here that the user can apply a number of terminals and therefore we have set username and terminal as a primary key to identify a user profile. The profile is serialized in an RDF/XML form and stored as a blob in the repository.

TABLE I
PROFILE REPOSITORY

username	terminal	profile
alice	Sony Ericsson P900	rdf/xml
alice	PC	rdf/xml
bob	Nokia 6600	rdf/xml

Location repository (table II) consists of the records containing the following fields: username, landmark's name, it's longitude and latitude, and a counter. In the initial implementation the system uses only positioning from the network and obtain user location information in the form of geographic coordinate (longitude, latitude). With the definition of landmarks we have envisioned that the user follows the usual routine every day: he goes to *work* in the morning, after which he goes in the *shopping* or to *recreation*, and later on he returns *home*. Therefore we have limited the user landmark declaration to the finite set: {Home, Work, Shopping, Recreation}. Every user will assign geographic coordinate (a pair of longitude, latitude) to the declared landmark name. Due to the fact that user can have more than one defined landmark, we have set the primary key as username and landmark.

The user registration form (Fig. 5) is provided through the Web interface to the administrator, who is authorized (at the beginning of the application) to administrate user profiles. The administration includes the profile's creation, viewing, update and deletion. After entering the data in the form, application generates a user profile document

TABLE II
LOCATION REPOSITORY

username	landmark	longitude	latitude	counter
alice	Home	165248E	452931N	3
bob	Work	162153E	451424N	2

according to the RDF scheme described in the section III. We have used Jena2 [8], RDF API for Java, to implement the generation and querying the user profiles. This profile is, along with the username and the applied terminal, stored in the profile repository database, enabling its later querying and update. Application also provides a remote API needed by other system components to update the user profile information in the profile repository.

Middleware (see Fig. 4) consists of the HTTP client handler that receives the requests from the user equipment and depending on the required action, routes them to one of the following component handlers: positioning handler, map handler, content provider handler, location handler, status handler and profile handler, that communicates with one of the following components: location server, map server, content provider, location repository, status server and profile repository. The selected component handler transforms the request into more appropriate form for the target component, forwards it and waits for the result from the component. When available, it forwards the received result to the user. Communication of the user equipment with the middleware is performed through XML over HTTP.

Considering the user terminals' heterogeneity and looking the diversity of their characteristics, our plan was to create three different client applications to interact with the system: Java 2 Micro Edition (J2ME) CLDC/MIDP application for mobile phones, CDC/Personal Profile application for PDAs, and Java 2 Standard Edition (J2SE) application for personal computers. The initial implementation supports the delivery of location-dependant content to the MIDP compliant user terminal.

At the beginning of the application user has to log in (see Fig. 6), after which he can choose among the following actions: define landmarks, create content, view visited landmarks, subscribe to content, modify user availability and modify user contacts & priorities.

Fig. 5. Profile registration form

Fig. 6. Client application

When the user selects the 'define landmarks' action, middleware sends the request to the positioning system and forwards the positioning result (a longitude, latitude pair) to the user. In the initial implementation user position is determined using the MPC (Mobile Positioning Center) emulator from the Ericsson's MPSSDK (Mobile Positioning System Software Development Kit) [9]. When receives a response from the middleware, client application displays a new screen (Fig. 7) containing user's coordinate, positioning time and choice group for selecting the landmark containing the values: Home, Work, Shopping, Recreation. User has the option either to submit the selected landmark to the middleware, or first to view his current position on the map, before he decides if he wants to declare his location as a landmark.

If he chooses to view the map first, middleware will send the request to map server to retrieve the map of the user's current location area. When receives the request, map server will load the map of Zagreb, crop the region of the located user area using the dimensions of user terminal's screen, and send the URL in the response to the user, where he can download the cropped map. After the user retrieves the map from the server, the client application displays it on the new screen, along with the user position shown as a red dot in the center of the map.

Fig. 7. Declare landmark

After receiving the request to submit landmark, middleware will first check in the location repository if the user has already declared the specified landmark, and if he hasn't, it will store the user record in the location repository and set the counter to start counting user visits to that location. If it determines that the landmark has

already been declared, it will compare the current user location coordinates with coordinates assigned to user defined landmarks in the repository. If the match is found, it will increment the counter value in the record of the matched landmark.

When the user selects the 'create content' action, he gets a new screen (Fig. 8) where he can choose from available topics on which topic he wants to publish his content, whether this published content will be location-based or not, and add the validity period of the information set (expired date and time). He can add text and image, and see the preview of the created content, as shown in Preview screen (see Fig. 8).

User can also subscribe to content by selecting the 'subscribe to content' action. He will select the topic from the list to subscribe to, and choose the type of subscription: (1) landmark-based, (2) current location-based and (3) non-location-based subscription on the displayed screen. These data will be sent to the middleware, that will route them to the content provider component.

A content provider is implemented as a publish/subscribe broker component using Joram JMS API [10]. The component is realized using the publish/subscribe mechanism extended to support location-based subscriptions and publications of location-dependent content. Publishers publish the content related to their current geographical position, while subscribers declare interest in receiving the particular information set at their current location or at the specified landmark. The goal of the publish/subscribe service is to match publications with subscriptions according to the topic and location associated with both events. The matching rule condition is met when the publisher and the subscriber are both located at the same position (in the case of current location-based subscription) or the subscriber has reached his landmark's position (landmark-based subscription), and the publication contains the same topic as specified in the subscription. The former match condition is defined as a *location match*, and the latter as a *topic match*.

The content provider will store the subscription, check if there is a publication to match it, and in the case of successful match, delete the subscription from the storage, check the user availability setting and deliver the published content to the user's terminal using the method

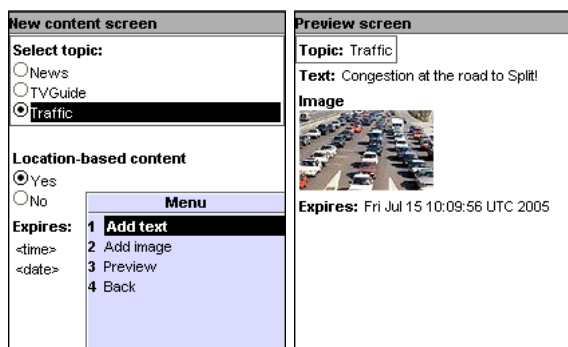


Fig. 8. Create content

specified in the user profile (SMS, MMS or e-mail) if the setting is set to 'available'. If the user changes this setting to 'unavailable' or 'discreet', content will be stored at the content provider until he reverts this setting to 'available' or the content validity period expires. The user availability setting has been by default set to 'available'. If the user changes this setting, it will trigger the update of his profile with this new information. In the initial implementation we have used Sony Ericsson P900 smart phone, and specified MMS as the delivery method in the user profile. In order to see the delivered message, the MMS Player provided by the same manufacturer was used.

The content provider provides a remote Location-Based Publish/Subscribe service interface (LBPS) to the middleware following the distributed component design approach. The LBPS interface, as illustrated in Fig. 9, proposes seven methods that will be described in the following.

```
public interface LBPS extends Remote {

    void publish(
        String topic, Message message
    ) throws RemoteException;
    void publish (
        String topic, Message message,
        String longitude, String latitude
    ) throws RemoteException;
    void publish (
        String topic, Message message,
        long timeToLive
    ) throws RemoteException;
    void publish (
        String topic, Message message,
        String longitude, String latitude,
        long timeToLive) throws RemoteException;
    Subscription subscribe (
        boolean onCurrentLocation, String topic
    ) throws RemoteException;
    Subscription subscribe (
        String topic, String longitude,
        String latitude
    ) throws RemoteException;
    void unsubscribe (
        Subscription sub
    ) throws RemoteException;
}
```

Fig. 9. LBPS interface

A publication represents an event distributed at the publisher's current location. It conveys the following information: name of the topic, serialized object containing content elements (text, image), content validity period and location coordinates (longitude, latitude). Publishers use the *publish()* method to create a publication.

Regarding the type, a publication can be *location-based*, when it is bound to particular location coordinates (longitude, latitude), or *non-location-based*. Considering the validity period of the information set, publication can be *persistent* or *non-persistent*. Non-persistent publications are disseminated *at the time of the publication* using the two *publish()* method variants leaving out *timeToLive* argument. With a persistent publication, an event can be distributed to the interested subscribers located at the determined location *before the event is unpublished*. Persistent publications are performed using other two

publish() variants taking *timeToLive* argument.

A subscription is a request to receive a notification from the publisher when the publication matches the subscription, according to the specified matching rule. It is performed using the *subscribe()* method taking three arguments, i.e., (1) topic, (2) longitude and (3) latitude, if the subscription is landmark-based (where longitude and latitude correspond to subscriber's landmark coordinates). Alternatively, the *subscribe()* variant with two arguments, i.e., (1) onCurrentLocation and (2) topic can be used to denote that the subscription should be based location-based or not. In the case of location-based subscription the subscriber's current location will be used. The *subscribe()* method returns a subscription object, which can be used by the subscriber to cancel the corresponding subscription with the *unsubscribe()* method.

The initial prototype is developed at the Department of Telecommunications, Faculty of Electrical Engineering and Computing, University of Zagreb.

VII. RELATED WORK

Providing contextualized content for mobile users has gained a lot of attention at this time. Not only third party content providers, but individual users can produce and publish the content. This content can be filtered, personalized and adapted to certain context elements. In [11], the authors demonstrated the content adaptation based on combined contexts (location, time and social aspects) on the example of a personalized mobile wap portal. They introduced social contexts consisting of user's state of mind (or mood), mode of spending time and group contexts and provided an RDF serialization of the ontology relevant concepts and their relationships. With using this portal users have to be always online to access the needed information. Our system offers the means of content delivery when the user is offline and is not actively involved in using the application.

The utilization of different position methods in location sensing is described in [12]. The authors discuss a problem of defining semantic location information and its application, but give no solution for it. In our work we tried to define a location semantics that integrate all forms of location information in the generic presentation format to be used by location-aware content delivery services.

The paper that influenced our work [13] presents a GeoNotes system for posting notes to other people in the surroundings, sharing context information. Users can participate as content providers and access other people's notes depending on preferences, situation and information needed. The system lacks the user mobility functionality and terminal heterogeneity. Our solution offers the user the possibility to apply several terminals to use the same service.

Paper [14] proposes a location-based publish/subscribe system intended for use by mobile ad hoc applications to communicate with each others based on location. In our system we have implemented the similar architectural pattern, but for the purpose of location-aware content delivery system. We have personalized the user's subscription

according to his current location, presence information, utilized terminal and preferred delivery method (SMS / MMS / e-mail).

VIII. CONCLUSION AND FUTURE WORK

We have identified the contextual elements needed to describe the user profile and specified their definition in the proposed RDF scheme. The information comprises user personal data, his applied device characteristics, presence information and subscription preferences. Regarding the location where user would prefer to receive the particular information set, we defined landmark-based and current location-based subscription. We proposed an architecture and initial implementation of the system that provides dynamic generation of user profiles, which are used to adapt and disseminate the content to the interested users at their specified locations. Considering the positioning methods that can locate a user and produce various forms of user location information, we proposed a generic presentation format of location information to be used by location-aware information services. In the near future we plan to enrich our system with new client modules written for PDAs to enable the terminal diversity in use of the services build on this system.

REFERENCES

- [1] A. Rakotonirainy, S. W. Loke, and G. Fitzpatrick, "Context-awareness for the mobile environment," in *Proceedings of the Human Factors in Computing Systems Conference*, 2000.
- [2] B. N. Schilit, N. Adams, and R. Want, "Context-aware computing applications," in *Proceedings of the First IEEE workshop on Mobile Computing Systems and Applications*, 1994.
- [3] "RDF Vocabulary Description Language 1.0: RDF Schema," W3C Working Draft available at: <http://www.w3c.org/TR/rdf-schema/>, World Wide Web Consortium, January 2003.
- [4] "Composite Capabilities/preference profiles (CC/PP): Structure and vocabularies," W3C Working Draft available at: <http://www.w3c.org/TR/CCPP-struct-vocab/>, World Wide Web Consortium, March 2003.
- [5] "Wireless Application Group User Agent Profile Specification," <http://www.wapforum.org/what/technical/SPEC-UAPProf-19991110.pdf>, WAP Forum, 1999.
- [6] "Wireless Village Presence Attributes V1.0," <http://www.openmobilealliance.org/tech/affiliates/wv/wv.pa-v1.0.pdf>, Open Mobile Alliance, 2002.
- [7] A. Devlic and I. Podnar, "Location-aware content delivery using publish/subscribe," in *Proceedings of the Telecommunications and Mobile Computing (tcmc2003)*, 2003.
- [8] "Jena 2 - A Semantic Web Framework," <http://www.hpl.hp.com/semweb/jena2.htm>, HP Labs Semantic Web research.
- [9] "Mobile positioning system software development kit," <http://www.ericsson.com/mobilityworld>, Ericsson.
- [10] "Java Open Reliable Asynchronous Messaging (JORAM)," <http://joram.objectweb.org/>, ObjectWeb.
- [11] S. Toivonen, J. Kolari, and T. Laakko, "Facilitating mobile users with contextualized content," in *Proceedings of the Artificial Intelligence in Mobile System 2003 Workshop*, 2003.
- [12] P. Ibach and M. Horbank, "Highly available location-based services in mobile environments," in *Proceedings of the International Service Availability Symposium 2004*, 2004.
- [13] F. Espinoza, P. Persson, and H. Nystrom, "Geonotes: Social and navigational aspects of location-based information systems," in *Proceedings of the Ubiquitous Computing*, 2001.
- [14] P. T. Eugster, B. Garbinato, and A. Holzer, "Location-based Publish/Subscribe," UNIL Technical Report, Sun Microsystems and Université de Lausanne, 2005.