A Guide to Urban Mediator

- what it is, what it does, and how you can use it

Joanna Saad-Sulonen and Andrea Botero (eds.)

Contact person: Joanna Saad-Sulonen, joanna.saad-sulonen@taik.fi

Helsinki 2009

Table of Content

Acknowledgements

What is Urban Mediator?

Urban Mediator in brief

Urban Mediator (UM) is a server-based software that provides a way for communities to mediate local, location-based discussions, activities, and information. Its goal is to provide users (citizens as well as city administrations) with the possibility to create, obtain, and share location-based information that is organized according to topics of interests set up and maintained by the users themselves.

Urban Mediator uses available map-portrayal service as means for representing location-based information and complements it with a set of tools for users to process, share and organize this information. The Urban Mediator software, once installed on an appropriate server, provides a customizable instance that is accessible and usable online, through the web, using a normal PC or any browser-enabled mobile device (including mobile phones, PDS, tablets, which include a pointing device). Print Tags (2D barcode matrices) can also be used as access facilitators to the Urban Mediator mobile interface, making it possible for users to directly reach a particular UM page.
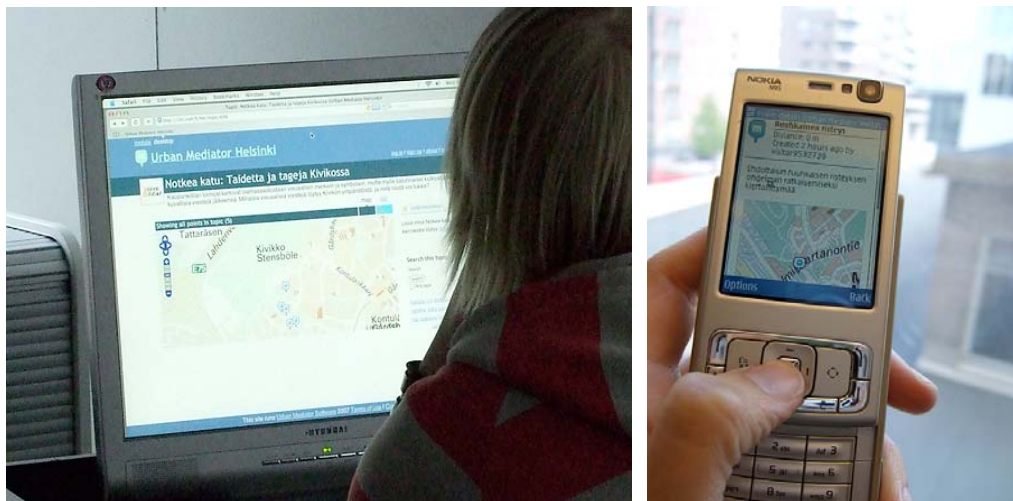
Figure 1. Urban Mediator is accessible on desktops and mobile phones

Main concepts, features and functionalities

Concepts and wording used in and about Urban Mediator have evolved throughout its design and development. This reflects the iterative and collaborative path followed and the need for developing appropriate language and terms along the way in order to address the involvement of the various stakeholders and their different backgrounds.

The current version of the software (UM v2.0) includes several core concepts: UM (the whole), topic, point, tag, user, comment, widgets and tools. These concepts are explained in detail below.

Figure 2. A diagramatic overview of the Urban Mediator main functionalities and components

Topics are used to gather collections of Points related to an issue of interest. Anyone logged in to Urban Mediator can set up a Topic by giving it a name, a description, a color scheme, an icon, and select a map area that shows as the default. Contrary to the use of the term topic for forums and BBS software, UM does not, in its present form, encourage discussions: it is rather a place to link many discussions together.

Points are used to mark locations that hold information, such as location (Where), creator (Who), time of creation (When), explanation (What), and purpose (Why). Anyone logged into Urban Mediator can create a Point, within the context of a Topic, by marking it on the map or giving a street address, and by giving it a title and a set of keywords. It is possible to add comments to points.

Tags are free-form keywords, associated to Points and Topics. The tags subsystem makes discovery of the points or referring to the specific set of points more effective and can allow the exploration of folksonomies.

Widgets in Urban Mediator terminology are web widgets that offer the possibility to extend and embed the functionality and content of UM Topics into any other website. They invite users to do specific action (report something somewhere), allow to monitor certain topics or just provide a linking mechanism between for example similar discussions.

Tools in Urban Mediator are part of the Topics' functionalities and offer possibilities to process, share and organize the location-based information available. Some examples of tools available in UM v2.0 are the CSV tool that makes it possible to export information gathered through topics in a Comma-Separated Values format, and the Newsfeed import tool that makes it possible to import GeoRSS feeds, which will automatically create points out of the feed entries, for the associated topic. New tools can be developed in time, in response to user needs.

Feeds are collections of Points, Topics, Comments in the Urban Mediator, available in ATOM/GeoRSS, KML or LMX web feed formats. UM can also use incoming ATOM/RSS with GeoRSS from other websites or web services to massively add Points to its database, providing data to UM topics.

Urban Mediator Software

Architecture

UM software has client-server architecture. The server side is powered by an application of web.py framework, written in Python, which uses MySQL database to store data; this provides a platform with the possibility to create, obtain, and share location-based information that is organized according to topics of interests set up and maintained by the users themselves. UM extends it with a set of tools for users to process, share and organize this information. Some extensions to Urban Mediator have also been developed to connect to other systems such as city customer services or community portals.

The map portrayal, powered by Javascript library OpenLayers (OL), is used for representing location-based information. The actual map chosen to be used is not dependent of UM and is to be accessed by UM via a Web Map Service (WMS).

ÚM is web based and can be accessed using a normal PC or any browser-enabled mobile device (including mobile phones, PDS, tablets, that include a pointing device). The interface optimized for mobile use offers a selected set of features that facilitate the use of UM by mobile users, focusing on the functionalities required while "on the go". UM has also some more experimental features which are described in the Experimental features section of this document.
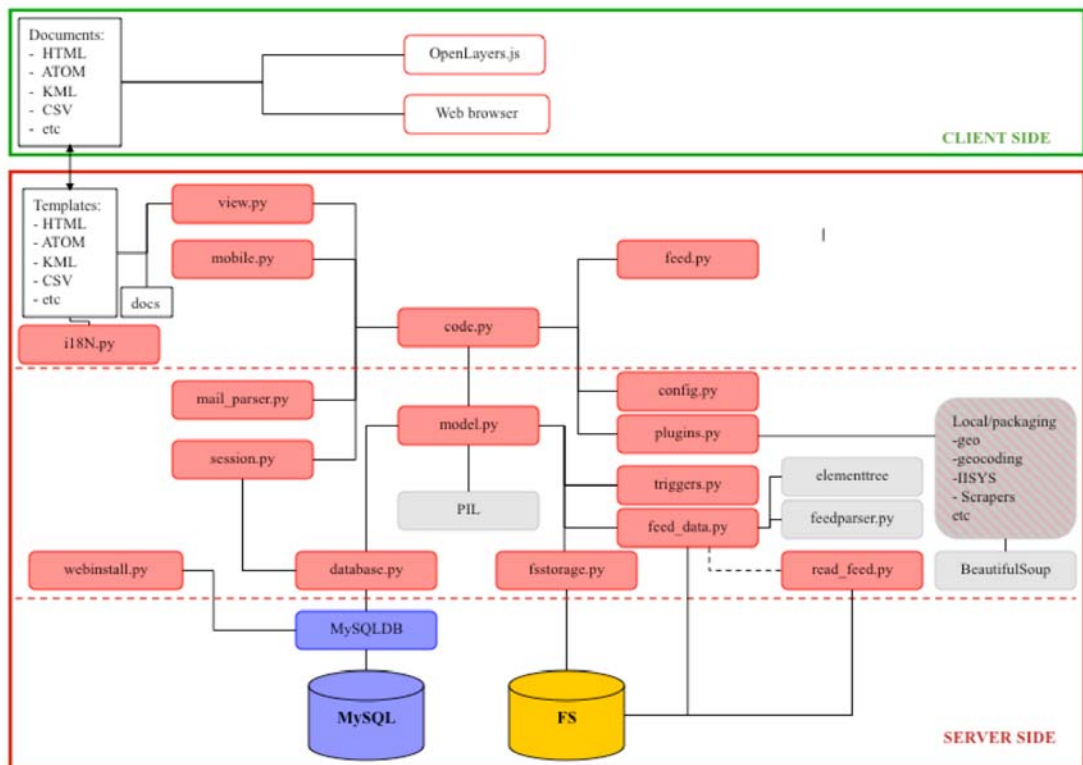
Figure 3. The Urban Mediator architecture is a typical HTTP client-server one

UM v2.0 software release information

At the time of writing the latest software release is UM 2.0.0. The UM v2.0.0 package is available for download as a Zip file, from http://um.uiah.fi/.

Package name: UrbanMediator v2.0.0

Version: v2.0.0

Date: 12.06.2008

Platform information: Unix/Linux, Mac OS X, Windows (not tested)

UM v2.0.0 package includes:

- Software source code and executable files

- README.txt

- LICENSE.txt

The user interface is currently available in the following languages: English, Finnish, Spanish, Catalan, Dutch and Russian. Offers for translations to other languages are welcome.

In order to manage the future development and guarantee sustainability of possible future development efforts, an Urban Mediator project has been

created on the Open Source Software development and distribution site Source Forge: http://sourceforge.net/projects/urbanmediator/. Coordination with interested future new developers and contributors can be achieved through this infrastructure.

Key requirements for installing and running the software

The key requirements and installation guidelines are available in the installation documentation available in the README.txt file in the UM v2.0 package as well the README_MAP.txt file for requirements related to setting up a chosen map available via WMS. This documentation is also available in Appendices A and B of this document.

License
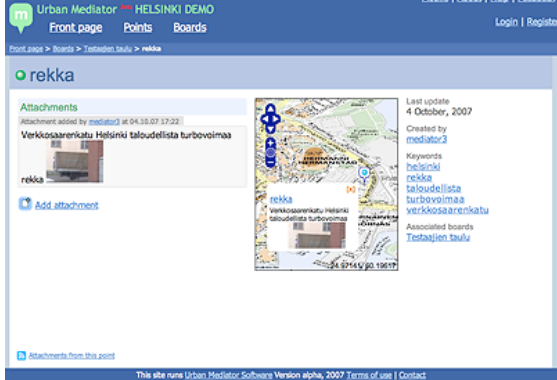
The software is released as Open Source, under the New BSD license. Information about the license is available in the LICENSE.txt file in the package as well as in Appendix C of this document.

Use cases

Since Urban Mediator is available both with a desktop and a mobile interface it is important to notice that a slightly different set of use cases is available in both. The Desktop use cases cover more social processing and administrations tasks, whilst the mobile ones targeted access and contribution to information in-situ.

UM Desktop use cases

1. Create and view points	
	a) User creates and tags Points The user indicates the location of the Point on the map, and provides a title, tags, and description for it.
	b) User views Point information Each point has its own page, with indication of its location, creator, date of creation, tags used, associated topics, and attachments.

	c) User contributes to existing Point with text, picture or URL Text comments, URLs and pictures can be attached to any existing Point.
	d) User views Point(s) on map All Topic points can be viewed on the Topic page map and can be filtered by tags. The screenshot shows an example of the Urban Mediator Topic set up for the Forest Research Institute in Helsinki to ask people to propose natural elements and locations in Helsinki as nature landmarks.
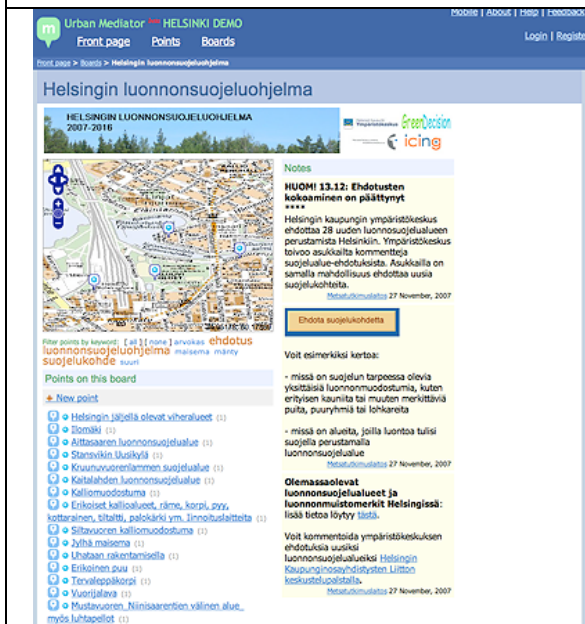
2. - Create topics of interest in order to gather location-based information
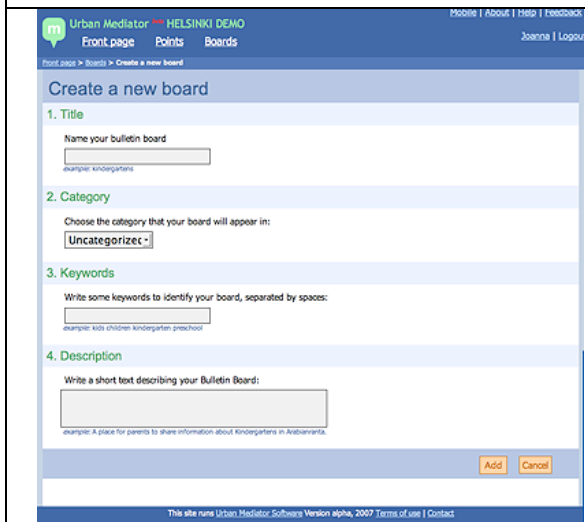
	Any registered user can create a Topic of interest for a UM instance.

3. Search all the information collected though all Urban Mediator topics
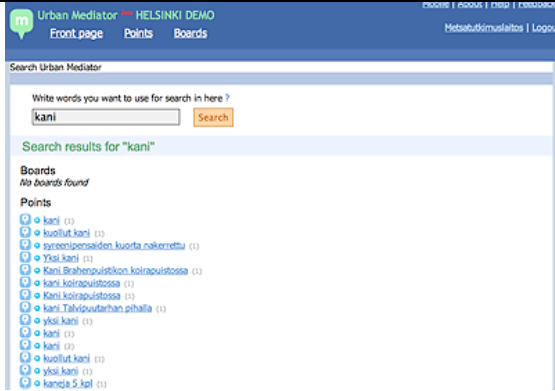
User can search by keyword across topics.

4. Import and syndicate content to and from UM by using standard feed formats (ATOM/GeoRSS, KML)

Subscribe to the newsfeed

A User can subscribe to different newsfeeds of a UM instance. These include: feeds for a topic or for comments, a feed list for a specific point. This is done by clicking on the Subscribe to the newsfeed link available at the bottom of each page.

A User can import content into UM from an external feed by using the Import newsfeed tools available at every topic.

5. Create UM widgets to embed UM functionality and information in other web resources (websites, blogs, etc)
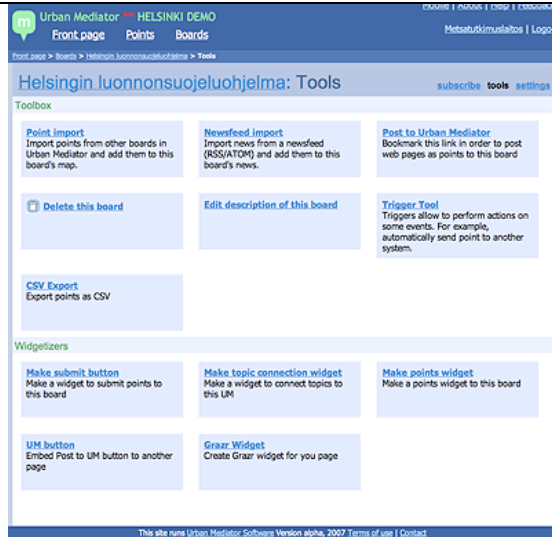
	a) User creates widgets for own Topic

Widgetizers are provided for registered users and Topic owners in order to create various web widgets easily. These web widgets extend Urban Mediator services into external websites. |
| | b) User embeds UM widget on a web page he/she administers
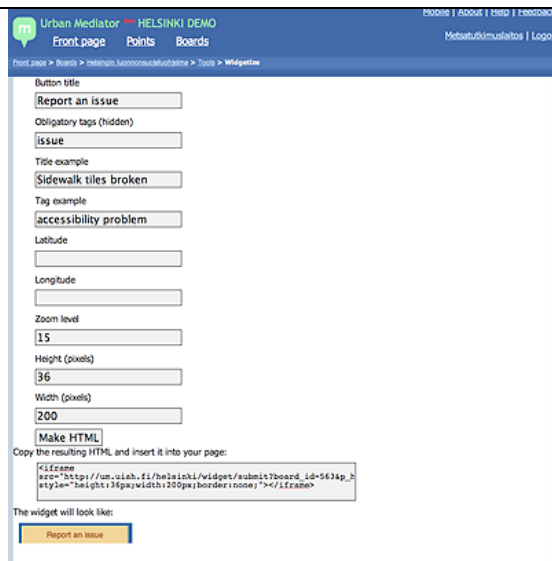
Each widgetizer creates the needed widget according to users' imputs. The widget appears as a preview and its code imbedded in an <iframe> can be cut and pasted into the HTML code of any other web page. |
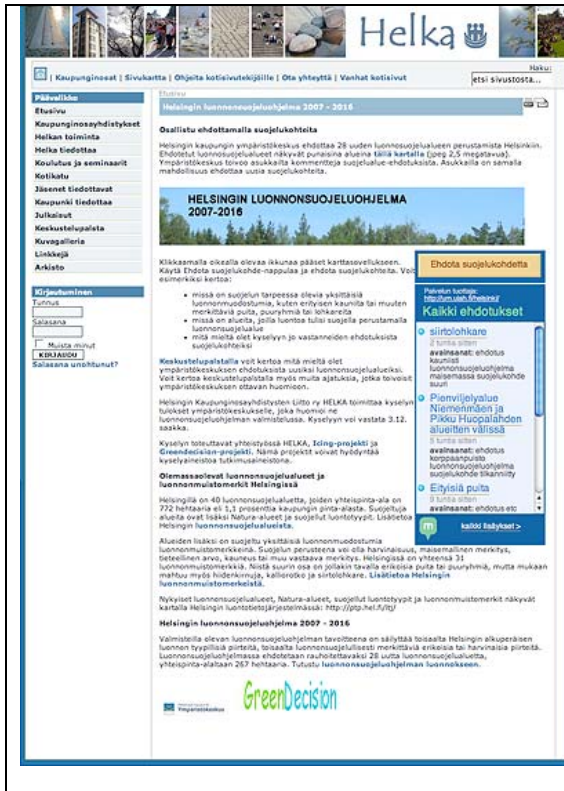
c) User uses UM widget on an (external) web page
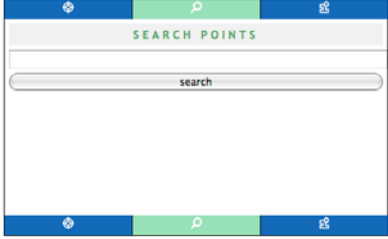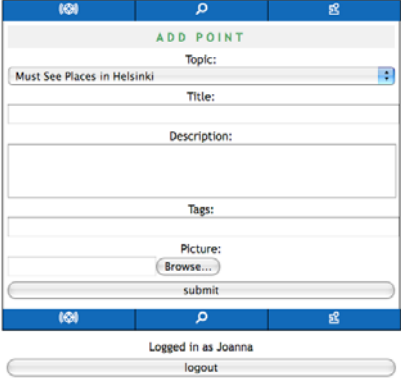
Once UM topic widgets are embedded in another webpage, users can use them to interact with UM.

This example shows two UM widgets (Add a Point widget and Topic Points List widget) imbedded into the Helsinki Neighbourhoods Association website (http://www.helka.net) during a UM trial in Autumn 2007. In this case, citizens were asked to report nature elements or locations in Helsinki, which in their opinion, should be labeled as nature landmarks and thus be protected

UM mobile web use cases

1. Search for information available on UM, on the spot

a) User searches for information related to a particular location

	b) User searches for information using keywords
2. Report/ contribute to UM, on the spot	
	The user creates a point related to a particular topic to which he /she is subscribed to (subscription to a topic happens via the desktop version). This is possible either by navigating UM and choosing the appropriate Topic to add a Point to, or by using external access possibilities like Print tags, which guide the user directly to the possibility to add a point to a particular topic. The user can add a picture to contribution. In this case, the user first has to take a picture, then sends it to the relevant Topic on UM.
3. Receive subscribed information	
	User subscribes to the UM feeds (instance feeds, topics feeds, points' comments feeds) and access them using the mobile phone browsers' feed reader.

Existing Urban Mediator instances

If you want to learn in practice what can be possible with Urban Mediator we suggest you look at some of the available services running with it. There are currently several Urban Mediator instances that reside on the University of Art and Design servers. The following is a list of these instances as per the 5th of December 2008:

- **Urban Mediator Barcelona**: http://um.uiah.fi/bcn

Urban Mediator Barcelona has been set up during the ICING project. Geocoding and map portrayal for UM Barcelona is provided by Institut Municipal d'Informàtica (IMI) and eSpatial. The map used is Copyright of the City of Barcelona.
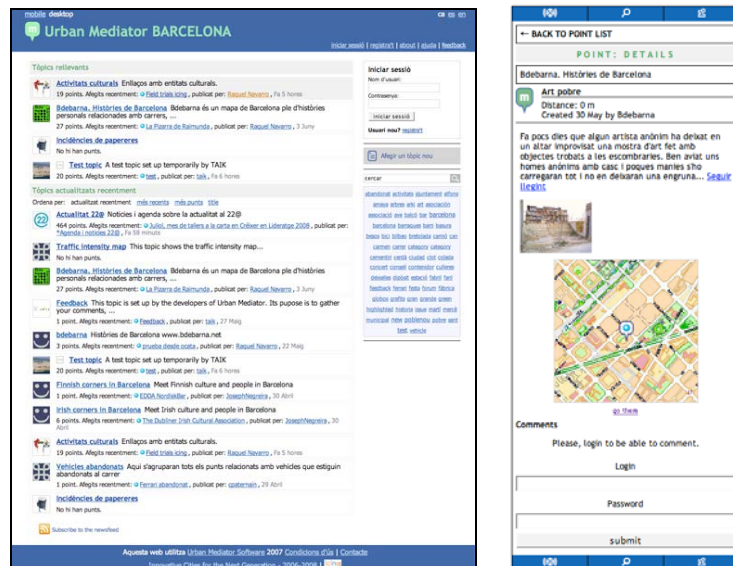
Figure 4. Screenshot of the Urban Mediator Barcelona front page showing the list of existing topics. Next to it the UM mobile interface for UM Barcelona, showing a point on Carrer de Bilbao.

- **Urban Mediator Helsinki**: http://um.uiah.fi/hel

Geocoding and map portrayal for UM Helsinki is provided by the City of Helsinki Survey Department. The map used is Copyright of City of Helsinki Survey Department.
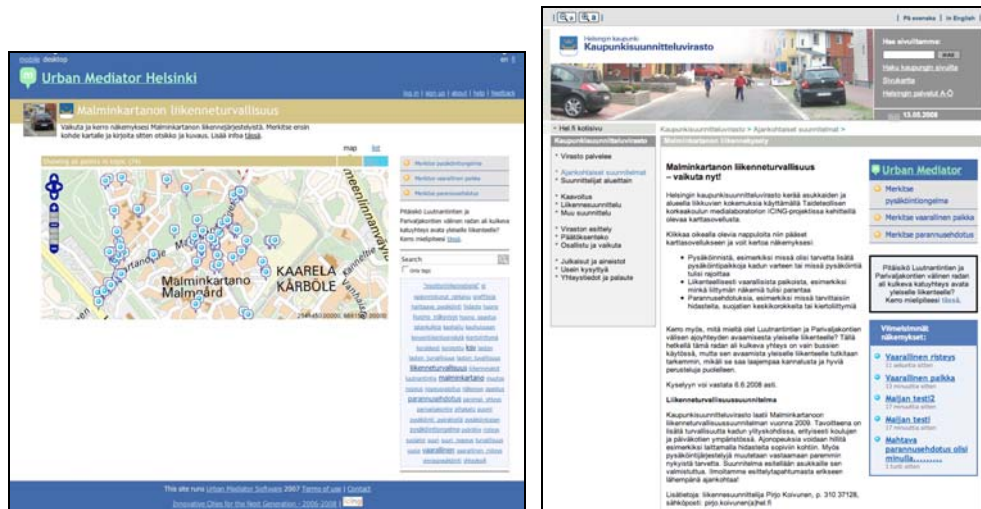
Figure 5. Screenshot of the Urban Mediator Helsinki: the Malminkartano traffic safety. Next to it a screenshot of the City of Helsinki Planning Department website, as it appeared on 13.05.2008, with the UM widgets for this topic embedded in it.

- **Urban Mediator Dublin**: http://um.uiah.fi/dub

Geocoding and map portrayal for UM Dublin is provided by eSpatial. The map used is Copyright of NAVTEQ.

Figure 6. Screenshot of the Urban Mediator Dublin showing a point on the Dublin Accessibility Audit topic. Issues can be added through other components of the ICING system, or through UM web interface. The Topic is connected to the Complete ICiNG system using a set of UM tools.

- **Urban Mediator EU**: http://um.uiah.fi/eu

The map used is OpenStreetMap, a free editable map of the whole world.

Figure 7. Screenshot of the Urban Mediator EU: the "Issue reporting in European cities: trash" topic.

- **Urban Mediator Demo**: http://um.uiah.fi/demo

The map used is OpenStreetMap, a free editable map of the whole world.
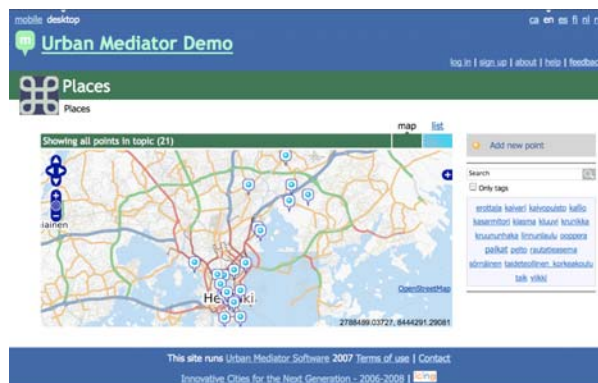
Figure 8. Screenshot of a test topic on UM Demo

Please note; as the Urban Mediator software is openly available for download and use, there might be instances available elsewhere.

Urban Mediator: co-design and use

UM development has, since the start of the project, followed an iterative, co-design approach, where various stakeholders (e.g. active citizens, schoolchildren, local developers, city planners) have been involved in the design process by taking part in a variety of co-design activities, such as workshops and use of prototypes.
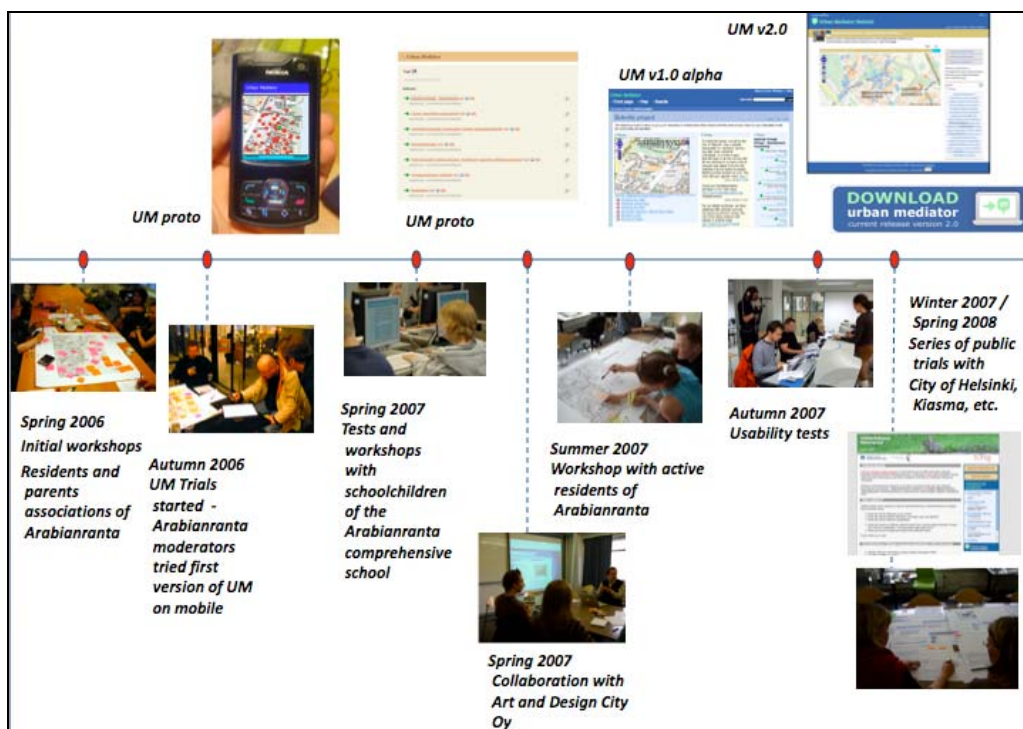
Figure 9. shows an overview of the co-design activities, interweaved with prototype production

The later versions of Urban Mediator (from UMv1.0 beta to UMv2.0) have been used by the general public through a series of participatory projects. The following list is a selection of these projects, including those set up in collaboration with the City of Helsinki. The setting up of these projects and the experiences related to them give a good idea of the types of practices and services we aim to support with UM features.

- **Reporting bunny rabbits**

This first project in collaboration with the City of Helsinki Public Works Department (*Rakennusvirasto*) was about asking citizens to report the locations where they have seen bunny rabbits (a new threat to park vegetation in Helsinki!) and damage done by them. UM version 1.0 was used and a UM topic entitled City rabbit situation in Helsinki (*Villikanitilanne Helsingissä*) was set up. UM "add point" widgets were created and embedded in an information page about the project and the general rabbit situation

(http://icing.hel.fi/villikanitilanne). The widgets used prompt the user to: 1) Report rabbit sightings and 2) Report damage done by rabbits.

The project started the 1st of October 2007, with short announcements provided by the Public Works Department placed in different media, advertising the bunny rabbit info page (e.g in Helsingin Sanomat, the biggest newspaper in Finland, in the City of Helsinki main information portal, in the Arabianranta portal, etc). During this public trial that lasted until the 4th or March of 2008, 450 rabbits sightings were reported as points on the UM map. The information gathered included detailed accounts of the amount of rabbits seen, their eating habits and behavior, coupled with exact location information (easy to be imported to the City's own GIS systems). In parallel, a discussion thread about the topic was set up on the web discussion forum on the Neighborhood Association of Helsinki (Helka), with links to the information web site containing the UM widgets.

- **Traffic safety in Malminkartano**

This project, in collaboration with the City of Helsinki Planning Department (*Kaupunkisuunnitteluvirasto*), asked residents of the Malminkartano neighborhood in northwest Helsinki to map their opinions on traffic issues in their neighborhood, as a way to inform the upcoming traffic safety-planning project.

A topic was created on UM Helsinki (now using UM v 2.0), entitled Traffic safety in Malminkartano (*Malminkartanon liikenneturvallisuus*). UM widgets were created for this topic and were embedded in the Planning departments' website, within a page explaining the project. There were three "Add point" widgets (Mark a parking space, Mark a dangerous space, Mark suggestions for improvement) and one "Points list" widget showing the latest contributions. Moreover, one question that the planners wanted to ask residents about opening an underpass to car circulation was included as a "Note" widget that lead to a Point page indicating the location of the underpass. By opening this point, citizens could respond using the commenting feature.

The project lasted from the 13th of May 2008 to the 6th of June 2008. 74 points were created and there were 27 comments as replies to the underpass question. The information gathered has been transferred to the department's GIS system MapInfo and will be used by the traffic planner during Spring 2009 in designing the new traffic safety plans for the area.

- **Proposed locations for skateboard park**

The third project was a collaboration with the City of Helsinki Youth Center (*Nuorisoasiainkeskus*). The Youth Center is a department at the City of Helsinki, addressing youth issues in the city. The Youth Center was interested in using Urban Mediator in June 2008, to ask young people to propose locations for a new skateboard park. We provided them with some instructions on how to create a topic and widgets, and their web editor took it from there. They themselves created a topic on UM, worked with the Urban Mediator tools and inserted widgets into their own website. The flexibility offered by the use of both free tags and pre-defined tags was also highlighted in this case: the

Youth center wanted users to indicate the age group to which they belong. This was easily achieved by using the pre-defined tags feature, offering the possibility for users to choose between the following categories: 7-12 years, 13-17 years, 18-22 years and Above 22 years.

The project lasted from the 27th of June 2008 to the 31st of August 2008. 39 points were created. At the time of writing, it was not yet clear how the information gathered will be processed by the Youth Center.

- **Other**

Various other experiments have been done in collaboration with local groups (for example with the Bdebarna active citizens group in Barcelona), local developers (for example with Art and Design City Oy in the area of Arabianranta in Helsinki), Kiasma (the museum of contemporary art of Helsinki), as well as with the Dublin City Council.

Instructions

The following instructions are target to users of Urban Mediator. The first set provides general instructions for use for casual users. The second set of instructions is geared for more advanced users who are interested in using the full potential of Urban Mediator as a public participation tool.

General instructions for UM use

The bulk of the following user documentation automatically appears as Online Help, under the Help section of any Urban Mediator instance using UM v2.0. It is currently available in English and awaiting translations in other languages.

Desktop version

Urban Mediator's full set of functionalities is available in the desktop version. A series of explanatory screencasts that complement the following instructions will be available at http://um.uiah.fi (What is UM).

- Login

In order to be able to add information to Urban Mediator (for example: add points for a Topic, or create a Topic), you will have to first Log in.

You can register and create a username and password for yourself and use it to login anytime.

You can also log in as an anonymous guest. For that, simply copy the characters you see in the picture in the Login page. This login allows you to submit information and edit it for the time you keep your browser window open or until you log out.

- Points: features and operations

Points are used to mark locations that hold information such as location (Where), creator (Who), time of creation (When), explanation (What), purpose (Why). Anyone logged in to *Urban Mediator* can create a Point, within the context of a Topic, by marking it on the map or giving a street address, and by giving it a name and a set of keywords.

View Points

You can view Points within the context of a Topic in the following fashion:

- From a Point list: click on the List link and icon that is available for each Topic. You can choose to order the list by: recently updated, newest, oldest, or by the point titles alphabetical order.

- From the map view: click on the Map link and icon that is available for each Topic. On the map, you can click on the points to open up their info box.

You can filter your view by using the Topic's Tag cloud and clicking on the desired tag. You can also search the topic by using the Search topic functionality.

Create Points

You can add a point only within the context of an existing Topic.

1. Go to the desired Topic page.

2. Click on the Add point buttons that are available to the right hand side of the page. These buttons can either show the default text "Add new point" or show a text generated by the creator of the Topic. There can also be one or more buttons

3. Select the desired spot on the map. You can zoom in and out using the bar at the left of the map. You can move right or left, up or down, by using the arrows on the left of the map. Click on the desired spot, this will mark the selected spot. You can also mark the location by simply typing the street address in the field just under the map and click on Find.

4. Write an appropriate title for your point/contribution

5. Write a short description

6. Attach a picture if you want. The picture will be uploaded from your computer.

7. Write tags that describe your contribution

8. Click Submit

8. If you had not logged in, you will be asked to do it at this point.

Edit and Delete points

Registered users can edit and delete their own points, when logged in.

Visitors can delete and edit their own only during the time they are logged in as visitors. Once a visitor logs out or closes the browser, s/he can't edit nor delete the created points anymore.

- Topics: features and operations

Topics are used to gather collections of Points related to an issue of interest. Anyone logged in to *Urban Mediator* can set up a Topic by giving it a name, a description, a color scheme, an icon, and a map area that shows as the default area.

Urban Mediator offers the possibility to create topics of interest where collection of location-based information (points) can be gathered. Topics can be created to show information coming from external systems and websites, through feeds. Topics can also be created for users to submit information to (add points to a topic).

View Topic

You can view the list of Topics on the Urban Mediator frontpage. This list can be ordered by: recently updated, newest, most points, or by the topic titles' alphabetical order.

There can also be a list of highlighted topics that is set up by the Urban Mediator administrator.

Create Topics

1. In order to create a Topic on Urban Mediator, you have to be a registered user. If you have to yet registered, you can do it by going to the Sign up page and choosing a username and password.

2. Log in using your Username and password

3. Click on the "Add new topic" button on the Urban Mediator frontpage.

4. Fill in the required fields in the form. Remember to keep the description short (maximum 4 lines). You can customize your topic page by choosing a color scheme and an icon. You can upload an image of your own to use an icon. Best results are achieved if the image is 60x60 pixels.

5. Click Submit

6. Continue by choosing the map that will appear on your topic page:

       - choose the desired zoom level by using the + and – buttons on the top left hand side of the map

       - pan the map to the desired area by using the arrows on the top left hand side of the map, or by grabing and moving the map

       - choose the map's center point by clicking on it

       - You can also go directly to a desired location on the map by typing in the exact street address

7. Choose Yes for Automatic zoom if you want your topic map to adjust so that it will show the area including all the points created for that topic. Choose No for Automatic zoom if you want the topic map's default view

to always remain the same as the one chosen in step 6. Automatic zoom is set to No as the default.

For instructions on how to use the Advanced Settings for topic page, please refer to the section on advanced settings.

Topic tools

The set of tools for Topics allows you to:

- Import newsfeeds from external source, like RSS or Atom feeds, to a Board (*Newsfeed import*). Importing a GeoRSS feeds will automatically create points on the board.

- Create a browser button that allows posting a current page to Urban Mediator and creating a Point containing a link to it (*Post to Urban Mediator*).

- Connect an UM topic to another system (*Add trigger*), for example the ICING system.

Describe the trigger: give a short description for the trigger.

Condition (e.g. addpoint): condition which fires the trigger (e.g. when user adds a point.) For example, 'addpoint' means that the trigger will be fired right after the point has been added. It allows the implementation of hooks, such as sending the point as an issue via Issue Reporter.

Action: the action that the specific adapter should perform in the even of

fulfilled condition, e.g. send

Adapter: the name of the adapter used (for example: iisys, ismart, sendmail)

URL (as a parameter for the adapter): additional parameter for the adapter, for example, DataAccessor URI where issues are being stored

- Convert the Topic's point information into CSV (Comma Separated Values). The *CSV export tool* makes it possible to automatically get a CSV file with all the topic's points data.

Widgets

It is possible to easily generate web widgets that can be embedded in any webpage. These widgets can for example be for adding a point to UM, for showing the list of recent points, for creating a map, or for creating a note. It is also possible to generate a combo widget that includes two or

more of the UM widgets (for example, Add point widget + Point list widget). For more detailed instructions refer to the Widget Creation section.

Urban Mediator feeds

You can subscribe to and use Urban Mediator information in other applications by using the Urban Mediator RSS feeds. The Urban Mediator RSS feed icon appears on selected Urban Mediator pages and in your browser's address bar.

Mobile version

A chosen set of functionalities is available for the Urban Mediator mobile version.

Note: one needs to be registered to UM to access some of the functionalities and have list of subscribed topics (to be done using desktop). It is not possible to create topics on the mobile version.

The main functionalities are:

- Login

- Search by location

 --> search for location

--> search for existing points around location

- Search by keyword

- Create point

--> Either browse map using up/down/left/right arrows or type location and click Lookup location button. Note: The location can set set automatically if using phone with inbuilt GPS receiver and if have UM location app (see Experimental features)

--> Continue

--> add point

--> type login and password

--> select topic from list of subscribed topics

--> type:

    Title

Description

Tags

Picture (Note: picture has to be taken before hand with the phone's camera + saved in the phone or card memory)

--> submit

- View topics list, or Topic's points list or Points' comment

--> view RSS feeds for topics or points --> handy with mobile

Advanced settings

The advanced settings are geared for more advanced users who are interested in using the full set of Urban Mediator tools and functionalities in order to set up participatory projects.

Advanced topic settings

Once you have created a basic topic page, you can customize it further by editing the advanced settings (available under Settings):

1. Topic page elements

You can choose which elements appear on the topic page's right hand side column, and in which order they appear (the order of appearance on the topic right hand side column is from up to down and corresponds to the order in the settings field, from left to right):

new point button: the default "Add new point" button that automatically appears when a topic is created (see Figure 1). You might consider removing this default button if you create customized buttons using the Widgets options (see Figure 2).

widgets: any widget created for a particular topic using the widgetizers functionalities of Urban Mediator. The widgetizers are available under the Widget section. Widgets can be imbedded in the Topic page and their order of appearance on the right-hand side topic column is set here. The most common widgets to add to the Topic page are the Add Point button ones.

search: the search topic field (for searching the topic only, not the whole Urban Mediator)

tagcloud: a tagcloud showing the most used tags in the points created for this topic

triggers: and advanced features that can connect urban mediator to other systems

hr: the diamond shape graphical elements that acts as a separator between elements of the topic's right-hand side column

2. Point display

You can choose what kind of information is shown for each point, on the topic points list:

title: The title of the point, as provided by the point creator

added: the time when the point was added

url: if the point is linked to an external web page, this page's URL would appear

description-full: the full description for the point, as provided by the point creator

author: the username of the creator of the point, or the visitor number if the point was created anonymously

tags: the keywords attached to the point (both hidden ones set by the topic creator and the ones chosen by the point creator)

3. Default view mode

You can choose whether the default view for your topic page shows the map or the list of points. Both view are of course always available to users by clicking on the map and list link on the topic view.

4. Point view order

You can set the default view for the topic's point list:

recently updated: shows the most recently updated points first

newest: shows the newest created points first

title: shows the list of points ordered according to the alphabetical order of the points titles

5. Who can add points

You can choose who can add points to your topic:

anyone: Anyone can submit anonymously, without even typing the security number verification. This makes it possible for anyone to quickly submit points anonymously, but this might encourage spam.

visitors: Anyone can submit, using their own username and password, or anonymously. Anonymous users have to however type a security number.

registered: Only users who have registered to UM can submit points.

topic administrators: Only the topic administrators can submit points.

nobody: There is no possibility to submit points to a topic. This possibility is handy for topics that receive their information as feeds from other systems of websites, and do not require users to submit information (see Figure 1). Moreover, this possibility can be chosen if the topic would need to stop being active and would no more need to receive information from users.

Figure 10. Example of a Topic using the default Add point button

Figure 11. Example of a topic using two customized Add point buttons, each guiding the users to submit a particular kind of information.

Figure 12. Example of a Topic that shows feeds coming from other websites, with no possibility for users to add their own points manually

Connecting UM to other information sources using feeds

UM can be connected to other systems through feeds (ATOM, RSS) by connecting a topic to the feed.

In order to set up a topic to receive feeds, go to the topic's Tools and choose Newsfeed import. You'll need to fill the following:

Feed address (URL): this is the URL of the feed

Describe the feed: a short description for the feed

Feed for issues? → tick this option if special treatment is needed for issues (like indicating where they are stored or where they come from)

Check interval in seconds: the time lapse for refreshing feeds

It is possible for a topic to receive feeds from any website that provides geoRSS feeds. In the context of ICING for example, Topics can be set to receive feeds from eSpatial's iSmart system, making it possible therefore to view information created using other ICING components, such as the iMessenger application. For more information contact the Icing consortium: http://www.fp6-project-icing.eu/index.html

Connecting a Topic to the Complete Icing system (via IISYS)

These instructions are only for users involved in a project where the ICING system is used. For more information contact the Icing consortium: http://www.fp6-project-icing.eu/index.html

It is possible to connect a UM topic to the Icing system, so that points created on that topic reach the IISYS and become available to users using other components of the system.

Go to the topic's Tools and choose "Add trigger" and fill the following fields:

Describe the trigger: give a short description for the trigger. For Icing, this can be for example: ICING city

Condition (e.g. addpoint): condition which fires the trigger (e.g. when user adds a point.) For example, 'addpoint' means that the trigger will be fired right after the point has been added. It allows the implementation of hooks, such as sending the point as an issue via Issue Reporter.

Action: the action that the specific adapter should perform in the even of

fulfilled condition, e.g. send

Adapter: the name of the adapter used (for example: iisys, ismart, sendmail)

URL (as a parameter for the adapter): additional parameter for the adapter, for example, DataAccessor URI where issues are being stored

Widgets creation

A set of widgetizers are available for advanced users who want to extend Urban Mediator's topic functionalities to their own website or blog.

"Add point" widget

You can create customized "Add point" buttons for both your topic page and for imbedding on any other website. For that, you need to go to Widgets and choose "Add point button".

The critical information to fill is the following:

Widget Basic Settings:

- Text that will show on button

- Text to show when point creation is disabled

Add Point page Basic Settings:

- Point title example

- List of fixed hidden tags → these tags are set by the topic admin and are always attached to any point a user creates, using this widget. These tags especially help topic owners to filter the information collected according to their own needs.

- List of predefined tags (leave empty for none) → this feature is still more experimental

- Free tags (user's own)

- Point Tags example (for free tags)

If you want to embed the widget into an external page, you might consider adding a header to the button itself. This header can show the UM logo as well as a title and subtitle.

An Add Point button without header is suitable for the Topic page, e.g.:
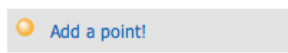
For an external page, it might be interesting to have more information, in the "header", e.g.
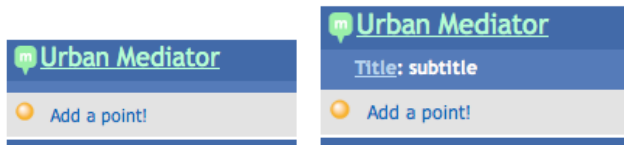
Remember to choose a convenient height:

- for a simple button, the height should be 36 pixels (default)

- with the logo the height should be 72 pixels

- with the logo and a title the height should be 90 pixels

Note: remember to check how the widget looks in different browsers!

Make Note widget

Creating some extra text that shows on the right hand side of the topic page can be done as follows:

Go to Widgets and select Make note. Then fill in the description and click Make html, then Add to topic page. This note can be used for example to insert logos or lists of relevant links.

Topic's point list widget

You can create a web widget that shows the most recent points submitted to the topic. This topic is best used for external websites. Remember to choose the number of points to be display and test to find a convenient height and width for the widget. It is also possible to include a header (similarly as for the Add Point widget).

NOTE: It is possible to use several widgets one on top of the other, on an external page, by either positioning each of their <iframe> codes one after the other, or by creating a Combo widget using the UM widgets.

Recommendations for affiliated sites

This document was originally written for the UM collaborators and possible stakeholders as guidelines.

One of Urban Mediator's goals is to bring together all relevant information related to a particular urban neighborhood. This includes existing online information from various official and non-official websites. Urban Mediator does not aim to replace these existing sources of information, but would rather offer access to them, giving users the possibility of accessing the variety of existing local information in such a way that it is always in the context of all other related information.

In order to include information into its index, Urban Mediator needs machine-readable web material. Each piece of data need to have metadata reflecting: What (usually a title), When (usually publication date), Where (WGS84 coordinates or other coordinates understood by UM), Why (some categorization, etc), Who (author or who is involved).

The following is a list of technical recommendations for websites that wish to be affiliated with Urban Mediator, or indeed other kinds of web applications that use information from external sources, like mashups. Please note that these recommendations should be considered as "giving a direction" rather than being strict requirements as in some cases there are no widely adopted formats and protocols.

1. Webfeeds

"A web feed is a data format used for serving users frequently updated content." A feed is in principle a collection of entries. Each entry has metadata associated with it: link to content, summary, author, datetime, tags (categories). Content is usually ordered by most recent entries first.

Popular webfeed formats: ATOM 1.0 and RSS 1.0, 2.0.

A typical webfeed logo:

More information about feeds can be found here:
http://en.wikipedia.org/wiki/Web_feed

http://en.wikipedia.org/wiki/Web_feedhttp://en.wikipedia.org/wiki/Web_feed
Web feeds works both ways: they can be used to syndicate data from the UM on the website or provide data to the UM from the web-site (or web-service).

2. Webfeeds with GeoRSS

GeoRSS encodes location information in RSS feeds. Webfeed with GeoRSS metadata make it therefore possible to visualize entries on a geographical map.

More information about GeoRss can be found here: http://georss.org/

3. Search API

API means Application Programming Interface.

In case a site has a large enough database, it can provide an API to search it. Even if the site has a search form for users, it doesn't mean that other software can use this search easily. Search API enables software to make queries.

Most often APIs are made with HTTP, RDF, XML-RPC or SOAP.

The simplest form of "API" is a specially designed URL. For example, page behind the URL http://del.icio.us/tag/Dapper shows entries found by keyword (tag) Dapper.

For Urban Mediator to make use of search API, special software adapters may be needed. Right now, we are not using any specific standards but this can be discussed with the interested partners.

More information about APIs in general can be found here: http://en.wikipedia.org/wiki/API

4. Web Widgets

Another way to interact with Urban Mediator is embedding UM widget on one or more pages. UM widgets (when needed) are aware of their host page. Simple use case is "advertisement" of certain UM topic(s) which need user's (citizens) contributions. More complex example is connecting certain discussion on the web forum with locational data (points) and/or other similar discussions).

5. Possible other Internet formats

There are various existing Internet formats can make web site content machine-readable, addressing the potentials of Web2.0. Using existing standards or emerging standards helps make interoperable Web possible.

Some examples include:

- "The Open Content Directory Format is intended to provide a concise, machine readable-listing of a set of syndicated services. The directory format is capable of supporting multiple sites, each with multiple services. Each service can have muliple formats such as RSS (RDF Site Summary), XHTML, Plain Text, Avantgo or WML format as well as separate publishing schedules or languages."

This site has a lot of real examples on how mashups can be done: http://www.programmableweb.com/

- "iCalendar is a standard (RFC 2445) for calendar data exchange."

Or http://internetalchemy.org/ocs/directory/0.5/

Experimental features

Location script

A location script writen in Python has been produced to make use of the internal GPS available in some Nokia S60 3rd ed series smartphone, to make it possible for the user to access Urban Mediator mobile web version in such a way that the map is centered on the user's location.

The instructions for using the location script are the following (they have been written and tested for Nokia's N95 phones only):

1- Download the ZIP file from http://mlab.taik.fi/urbanmediator/?page_id=21 and open it on your computer.

2- Sign PythonScriptShell and locationrequestor on https://www.symbiansigned.com/app/page/public/openSignedOnline.do

Remember to click on Select all for Capability information.

3- Send the following files (including the two SIGNED ones) to the phone (the most reliable way is by using the USB cable or Nokia PC suite rather than Bluetooth or email), and install in the following order:

a) PythonFors60

b) PythonScriptShell

c) locationrequestor

This procedure is standard for the phone, so follow the phone's instructions for installations.

d) Send now the loc_i.py file to the phone. Use the cable connector to connect the N95 to a computer. (For some obscure reason we found out that on 3rd ed. phones pushing a *.py script to your inbox via Bluetooth doesn't work). IMPORTANT: copy the file into a Python folder that you create on the phone's memory card (E: drive). Please make sure that you create the folder with a capital P, so: Python, and save the loc_i.py file into it.

4- Go to Application folder, click on Python application, go to Options, and select Run script, then choose loc_i.py.

5- The first time you run this script you will have to fill in the following info:

 Server URL: http://um.uiah.fi/bcn (or /demo or /hel or /dub ... whatever you're using)

(so replace the default http://um.uiah.fi/final)

Username: < type your UM username here>

(NOTE: if you want testers to use it, you need to FIRST create a particular username and password on the desired UM instance desktop version)

Password: < type your UM username here>

(NOTE: again, please type the password for the chosen username)

6- The phone will now search for GPS. Please note that you need to be near a window or outside, for best GPS reception. The GPS signal is received when the phone screen starts showing numbers or the phone says "GPS ". Please wait for that (the numbers or the voice) before moving to point 7.

7- Click Options, and Go to browser

8- You should now be on your desired UM mobile page that shows you the map centered on your GPS location. You can then click on Continue and then Add a Point.

The next times you want to use the GPS, the phone will no more ask you for step 5.

Using Nokia's N95 map application

A Mail-to-Um service has been created and tested in conjunction with the functionalities offered by the Nokia N95 phone's own map application. This makes it possible for users to create "landmarks" on the N95's map and send them by email to the email addresses that can be assigned to UM topics. The "landmark" created on the phone is translated into a UM point on the chosen UM topic.

Experimental mobile app

By Abhigyan Singh

An experimental Urban Mediator Mobile Application has been built for Nokia's S60 3rd Edition phones using Python S60. This application uses Urban Mediator backend.

The Application serves the purpose of facilitating users to add a point to UM from any location.

The Application's flow has been kept simple. A user starts by either specifying his/her location or by taking a picture. Application makes use of Nokia S60 3rd Edition Phone's inbuilt camera module. The Location Module of the application fetches the city map used in the UM instance, and centers the map on address typed and allows user to navigate, zoom in, zoom out and clicks on the map. (note: this requires that the phone is connected to the Internet).

Once the user has completed both the tasks of Location Specification and clicking picture he/she is presented with a list of his subscribed UM topics. He selects one of the topics, types title and submits the point to Urban Mediator using an Internet connection from the mobiles.

Map search tool

By Tuomo Tarkkiainen

Urban Mediator Map Search Plugin is a tool for exploring UM data. It offers a dynamic way to do searches to UM instance database and to explore search results by dynamic map -interface.
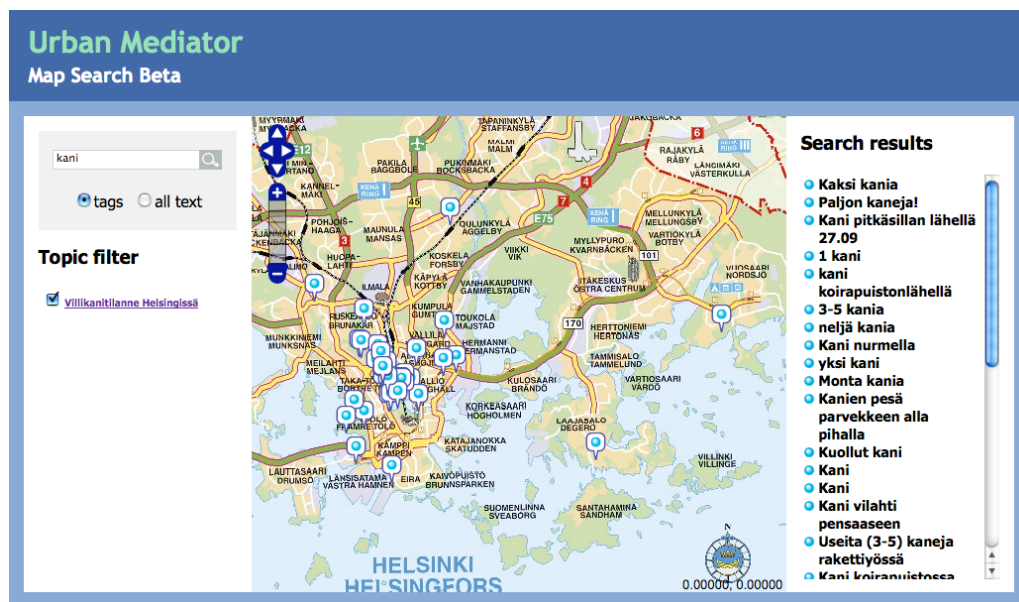

Figure. A screenshot of the Map Search tool (beta) on
http://um.uiah.fi/hel/map_search.html showing search results for the term
"kani" (rabbit in Finnish)

Map Search Tool is an independent alternative for UM search. It is not meant to replace the text search of UM but to enhance the user experience by offering more dynamic way to observe search results on map.

Map Search Tool is built in JavaScript, XHTML and CSS and it uses OpenLayers map -libraries to deal with maps. The prototype is strongly bound to WMS map projections but the final tool should be more abstract, a framework of map search - interface that could be used with any map data and location data.

Map Search Tool communicates with the UM server by JSON. All the queries and results are JSON -formatted but the form of the data is not standardized. There are some common ways to deal with this kind of data and they should be somehow standardized. It is possible that in the future UM development group is able to propose some kind of definition for this kind of map search data.

The tool itself was made for a purpose of finding out if it is useful to explore UM data in dynamic way ,so basically the tool is a research prototype which can be used for testing. Only the most essential functionality was implemented and the code was written in "quick and dirty" way so that in final tool the whole code should be designed again. The idea of the prototype was not to produce the optimal tool but to give insight on what parts of this kind of tool should be abstracted

Parts of the Tool

The tool has 4 main parts:

1. The search -form

The search form is basically one text field and two radio buttons plus a Search –button. The Search -button reloads the page. Even though the page is reloaded, the search query for the server is done by AJAX to make page asynchronous. Asynchronous search is useful approach since the amount of data that the server may return varies a lot. With an an asynchronous system it is also possible to load search result data in many phases but this was not implemented in prototype

 2. The List of Search Result Points

The search result is returned by server as a list of points and this list is printed as one part of UI . List entries have three modes: normal, mouseover and selected. The modes of list entries correspond to the modes of the current point's representation on map. With the map UI the list of points makes a pair of UI elements which interact with each other. From the perspective of usability it would be good if user wouldn't have to think why things are changing same time on the map and in the list of points. The representation of the point information on the map is called the marker of the point.

- In NORMAL MODE only the title of the point is shown

- In MOUSEOVER MODE the list of point entry is highlighted. This mode triggers the mouseover mode of the marker of the point (on the map) the

purpose of this mode is to provide the user a way to quickly browse through the list of points and to see where they are located on the map

- In SELECTED MODE: if a list entry is selected by clicking the title, it expands and shows

3. The Map

The list of search result points is also represented on the map as

Markers. The functionality of markers is bound to functionality of list of points so that also marker has three modes : normal, mouseover and selected. Considering the usability and intuitivity of the tool, markers functionality should resemble as much as possible the list of points entry functionality. The map also raises interesting usability problems since it is related both to topic filter and to list of points. There are multiple situations where user testing is needed to find out what kind practices would be most useful when zooming in and out and moving the map and filtering markers by topics. The prototype "remembers" chosen point after zoom and map movements but not after topic filtering

4. The Topic filter

The topic filter is a UM specific part of the map search –interface. The idea of the topic filter is to enable user to filter markers and list of points entries based on the specific topic they are interested in. The topic filter makes it easier for user to narrow down search results.

NOTES: With the current user interface, it might not be very clear for the user what is the difference between tags -based search and general text search

Related research

- A series of academic papers and studies were written about Urban Mediator:

    Saad-Sulonen J, Botero A (2008) *Setting up a public participation project using the Urban Mediator tool: a case of collaboration between designers and city planners*. In Proceedings of NordiCHI 2008: Using Bridges, 18-22 October, Lund, Sweden.

    Botero A; Saad-Sulonen J (2008*) Co-designing for new city-citizen interaction possibilities: weaving prototypes and interventions in the design and development of Urban Mediator*. In Proceedings of the 10th Participatory Design Conference. University of Indiana Bloomington. USA

Raike, A.; Saad-Sulonen, J.; Scheible, J.; Suzi, R. & Toikkanen, T. (in press) *Visual Tools for Accessible Computer Supported Collaboration*. In: Miesenberger & al. (eds.) Computers Helping People with Special Needs: 11th International Conference, ICCHP 2008, Linz, Austria, July 9–11, 2008. Proceedings. Springer, Lecture Notes in Computer Science.

Saad-Sulonen, J., Susi, R. (2007) *Designing Urban Mediator*. In Proceedings of the Cost 298 conference: participation in the broadband society. 23rd-25th May 2007, Moscow, Russian Federation.

Saad-Sulonen, J. (2007) *Introducing Urban Mediator: concept and work in progress*. In Research and Activism: 4th Urban Studies Days Conference proceedings, pp103-108. Department of Urban Studies, Estonian Academy of Arts, Estonia. ISBN 978-9985-9803-8-5

Saad-Sulonen, J. (2006) *First steps towards designing the "Urban Mediator"*. Presented at Tools'06, Media Lab Research Symposium, 19.05.2006 Helsinki. To be published in Proceedings of the symposium (forthcoming - 2008)

Saad-Sulonen, J. (2006) *Everyday Life in the Interactive City: exploring the potentials of interweaving digital technologies and urban space*. In E-City, Uli Bucher und Marcos Fincas (ed.), Berliner Wissenschaftsverlag. Berlin. 2008.

The list can be accessed online via the University of Art and Design's research database Reseda:
https://reseda.taik.fi/Taik/jsp/taik/Research.jsp?lang_global=en&id=26557

- Joanna Saad-Sulonen's MA thesis at Media Lab / University of Art and Design Helsinki introduced the concept of Urban Mediator:

Saad-Sulonen, J. (2005) *Mediaattori – Urban Mediator: a hybrid infrastructure for neighborhoods*. Master of Arts Thesis in New Media, University of Art and Design Helsinki, May 2005 (http://www2.uiah.fi/~jsaadsu/thesis.html)

- Information about the user interface design of Urban Mediator can be found on Eirik Fatland's website: http://www.efatland.com/portfolio/um/

- A study about the ICiNG platform and Urban Mediator in particular have been chosen by the SimoBIT project, as a case study in the mobile government domain as the ICiNG project has successfully launched a platform and actively markets this solution. SimoBIT is funded by the German Ministry of Economics and Technology. http://www.simobit.de/. The research into ICING and Urban Mediator is conducted by Martin Schmid from Ludwig-Maximilians-Universität München.

Appendix A: UM v2.0 README.TXT

```
Urban Mediator web-server

=========================


Urban Mediator (UM) web-server is a software project of Icing
research

project WP5 by ARKI research group of Media Lab of Univeristy of
Art

and Design Helsinki.


UM utilizes client-server architecture and is a web application
with a

usual web interface and mobile one.


Software in the package is under new BSD-like license (see
LICENSE.txt)

except for the modules mentioned under Dependencies (they were
shipped

with the UM software to make installation of the software easier
for the

end user)


Authors

-------

- Andrea Botero Cabrera, ARKI (project management) (2007)

- Eirik Fatland, ARKI (HCI, web design)

- Joonas Juutilainen, ARKI (some graphic design icons)

- Mika Myller, ARKI (software design) (2007)

- Iina Oilinki, ARKI (project management) (2006)

- Tommi Raivio, ARKI (software design, etc)

- Joanna Saad-Sulonen, ARKI (research, concept and user
interface design, project management)

- Roman Susi, ARKI (software design, etc)

- Tuomo Tarkiainen, ARKI (map search)

- Mark van der Putten, ARKI (HCI, web design, 2008)
```

- Abhigyan Singh, ARKI (usability tests, 2008)

and others.

Acknowledgements

Web.py has been choosen for the framework.

Coordinate conversion given to us by the City Council of Helsinki.

We are in gratitude to the developers of the programs UM depends upon.

Dependencies

Included with the Urban Mediator source code:

feedparser.py (UM uses slightly modified version)

    http://feedparser.org/

    (Open Source license)

BeautifulSoup

    http://www.crummy.com/software/BeautifulSoup/

    PSF license (same license as the Python)

FCKEditor

    http://www.fckeditor.net/

    MPL license (actually, it can be also distributed under GPL and LGPL)

OpenLayers

    http://www.openlayers.org/

    BSD License

freefont.ttf

http://www.tulrich.com/fonts/ (renamed Tuffy font)

Public Domain

webform.py

has been derived from form.py of web.py

Public Domain

Captcha code inspired/based on

http://aspn.activestate.com/ASPN/Cookbook/Python/Recipe/440588

Icons from ExtFile/ExtImage (meant as temporary, to be replaced soon!):

http://www.zope.org/Members/MacGregor/ExtFile

simplejson (encoder)

http://undefined.org/python/#simplejson

MIT license

json parser from json.org with small modification by Wensheng Wang

http://aspn.activestate.com/ASPN/Cookbook/Python/Recipe/440637

License?

Prerequisites

- Python 2.4.4 or 2.5.x

- MySQL v 5.0.38 or later

- MySQLdb v 1.2.2 - Python extension module

- web.py (0.22) - pure Python module

- elementtree - pure Python module (present in Python 2.5 readily)

- PIL (1.1.5 or later) - Python extension module

- pyproj (1.8.4 or later) - cartographic transformations and geodetic computations

Installation

============

Install (or check if they are installed) prerequisites mentioned above

according to their instructions for your operating system.

Important: (for web install) get hold of MySQL root password or password of a user

capable of creating new databases and users. In the example below:

root and SeCrEt respectively.

Hint: To setup root password for first time, use mysqladmin command at shell prompt as follows:

$ mysqladmin -u root password SeCrEt

(you may need to specify the full path to the mysqladmin executable, like

/usr/local/mysql/bin/mysqladmin)

Installing from tar.gz or zip

....

Untar and or unzip the package to the place where it will be run from.

(UM doesn't have ready start/stop scripts.)

Choose the port number (say, 9080) and the domain the UM will run

(e.g. localhost).

Go to the um root directory (of just untarred archive)

and run:

```
python webinstall.py 9080
```

Go to the browser:

    http://localhost:9080

Fill in the form and press "Setup DB". If everything will go smoothly,

MySQL database and user will be created and tables created in the

databse. If not, webinstall can't be made - manual install is required. (Please, check MySQL root password carefully

before doing webinstall).

Visit local/config.py and add your changes. Most likely,

these will be needed:

base_url="http://my.server.org:9080"

schema_base_url = base_url + "/static"

WMS server also needs to be specified: metacarta.com is

just to quickly show UM.

Look into config.py for more configuration parameters:

local/config.py override config.py settings.

Note about manual installation.

Instead of running webinstall:

- create a MySQL database and user with privileges to that DB

- put the latest sql/*-base.sql and sql/*-patch.sql with numbers greater than those of base.sql.

- put connection into local/config.py as follows (example only):

```
   db_parameters = dict(dbn='mysql', db='um_db', user='um',
pw='*secret*')
```

- other configuration parameters

Running: can be done (in screen, for example):

```
  python code.py 9080
```

UM instance variable data is stored in the ~/.urban_mediator

(files as well as cached feeds and map tiles), docs, local

and in the MySQL database. (I.e., to move UM instance,

one should copy them all to the new place.). Logs are writted

to stderr/stdout..

Installing from the egg

....

When UrbanMediator will be available in the PyPI, it

should be as easy as

east_install urban_mediator

Before that, download the egg file (the name is something like

urbanmediator-2.0_6409-py2.5.egg)

To install UM in the user space, setup PYTHONPATH to the

directory of choice (for example, in shell configuration file

like this:

export PYTHONPATH=/my/um/directory;other posible pathes

```
easy_install --install-dir=/my/um/directory urbanmediator-
2.0_6409-py2.5.egg
```

After that,

create a directory

/my/um/directory/local

and put config.py, plugins.py and other possible files.

To start web installer:

/my/um/directory/install_urbanmediator_server 9080

To start the server:

/my/um/directory/start_urbanmediator_server 9080

There is no way to stop the server but killing it.

file called PID with the server's process id

can be found in the /my/um/directory

Known issues

============

UrbanMediator software is designed to be a tool-kit for creation
of specific city-citizens and/or citizens-citizens interactions.
The main value is not software itself but the services it
provides. The software itself is, however, not void of certain
problems, bugs and limitations. Those are listed below in know
specific order. (Issues, marked with "--" were solved)

1. Some glitches are possible with OpenLayers map portrayal
depending on the browser. For example, click point shift may
occur in some relatively rare cases.

2. Starting and stopping software is not automated. At the
moment, on some platforms process should be explicitly killed to
be stopped, especially, if there are active users fetching pages

--3. Search engines do not index UM pages due to some meta tags.

--4. Restarting UM back-end makes users logout

--5. No paging of comments, points, list in 1.0 beta(solved in the new UI)

--6. No way to add tags after point has been created (solved in the new UI)

--7. Sign-up disrupts user contribution (thus, user should sign up with UM beforehand or use Visitor account)

8. Problems with uploading large files on some platforms (limit about 2Mb)

9. Web installer is not user-friendly

10. When switching language, form entries are lost

--11. No way to relocate a point once created

12. Out-of-extent locations are not properly handled

13. Comments HTML is not sanitized enough at the moment (this may be security problem for the users with vulnerable browsers is malicious JavaScript is included by another user)

--14. Deletions from the database are not journalled

--15. Delay in loading Front page (and maybe other pages) caused by MySQL query on Mac OS X 10.4 (works fine on Linux Ubuntu/Debian)

16. New point policy "registered" is not enforced enough – visitors can still add points

Appendix B: UM v2.0 README_MAP.TXT

```
Setting up the map

------------------

WMS

===

Urban Mediator (UM) uses OpenLayers (OL) as its map portrayal
engine.

In order to configure the map, WMS (or other OL compatible map

material) should be available.


Map logic is controlled via the following configuration
parameters

(editable by UM instance administrator):

getmap_layers  - (string) name of the most detailed layer

getmap_layers1 - (string) name of the less detailed layer

getmap_layers2 - (string) name of the least detailed layer


getmap_params - (string) map parameters, JavaScript code, e.g.:


    { maxExtent:

    new OpenLayers.Bounds(2546000, 6665800, 2568009, 6687009),

    maxResolution: 5000, units: 'meters',

    projection: "EPSG:2392", numZoomLevels: 20}


getmap_server - (string) URL of the server which provides WMS,
e.g.

    http://labs.metacarta.com/wms/vmap0?


srsname - (string) name of the SRS (projection), should be the
same as

possibly given in getmap_params.


getmap_zoom1 - (int) threshold between most detailed zoom level
and less detailed
```

getmap_zoom2 - (int) threshold between less detailed zoom level
and least detailed

with these two parameters, right level is choosen for the right
zoom level.

It may require some experimentation to choose zoom levels
correctly.

The best way to experiment is some topic view with the map: it
is possible

to set topic zoom level and then play with getmap_zoom1,2 to
achieve

best match.

getmap_zoomshift - (int) configuration parameter which
translates

UM's logical zoom level to OL's zoomlevel. In some cases it is
ok

to be 0, but sometimes it should be less. OL's zoom levels
possibly

depend on many other parameters (please, refer to OL docs). So,

this parameter may help to adjust zoom levels especially when UM

switch one map for another.

topic_zoom - (int) default topic zoom level. Usually, shoosed to
show

a small distrcit of the city. This zoom level is logical zoom
level.

center_lat, center_lon - (float) coordinates for the city center

(or district center, if UM is for the district). Coordinate
system - WGS84

UM database holds all coordinates in WGS84 internally and
translates

coordinates (if necessary) with pyproj library. UM database

saves only logical zoom levels and translates them to OL zoom
levels

by use of getmap_zoomshift parameter. (it may be done more
complex

than that in the future).

OSM

===

UM can work with the OpenStreetMap (and other epsg:900913 maps).

In order to enable that, be sure to use pyproj library with the following

additions to the /pyproj/data/epsg file (found in the pyproj):

<41001> +proj=merc +lat_ts=0 +lon_0=0 +k=1.000000 +x_0=0 +y_0=0 +ellps=WGS84 +datum=WGS84 +units=m +no_defs   no_defs <>

<900913> +proj=merc +a=6378137 +b=6378137 +lat_ts=0.0 +lon_0=0.0 +x_0=0.0 +y_0=0 +k=1.0 +units=m +nadgrids=@null +no_defs

To use OSM map, UM needs "custom" way of setting up the map. To switch that on,

edit the UM settings (login as admin first):

getmap_custom: 1

getmap_custom_server: http://tah.openstreetmap.org/Tiles/tile.php/

getmap_custom_wms: osm

getmap_zoomshift: 0

srsname: epsg:900913

and (for example, its possible to make other changes)

getmap_custom_init:

```
        var options = {

                projection: "EPSG:900913",

                displayProjection: "EPSG:4326",

                units: "m",

                maxResolution: 156543.0339,

                maxExtent: new OpenLayers.Bounds(-20037508, -
20037508,
```

```
                                                                  20037508,
20037508.34)

            };



    map = new OpenLayers.Map('map', options);


            var options = {numZoomLevels: 24};


            // create OSM layer

            var graphic = new OpenLayers.Layer.TMS(

                "OpenStreetMap (Tiles@Home)",

                "http://tah.openstreetmap.org/Tiles/tile.php/",

                {

                        type: 'png', getURL: osm_getTileURL,

                        displayOutsideMaxExtent: true,

                        attribution: '<a
href="http://www.openstreetmap.org/">OpenStreetMap</a>'

                }

            );



layer = graphic;

layer1 = graphic;

layer2 = graphic;

            map.addLayers([graphic]);

            map.addControl(new
OpenLayers.Control.LayerSwitcher());

            map.addControl(new
OpenLayers.Control.MousePosition());

            // map.zoomToMaxExtent();

}


function osm_getTileURL(bounds) {

            var res = this.map.getResolution();
```

```
            var x = Math.round((bounds.left -
this.maxExtent.left) / (res * this.tileSize.w));

            var y = Math.round((this.maxExtent.top - bounds.top)
/ (res * this.tileSize.h));

            var z = this.map.getZoom();

            var limit = Math.pow(2, z);


            if (y < 0 || y >= limit) {

                return OpenLayers.Util.getImagesLocation() +
"404.png";

            } else {

                x = ((x % limit) + limit) % limit;

                return this.url + z + "/" + x + "/" + y + "." +
this.type;

            }
```

Appendix C: UM v2.0 LICENSE.TXT

Copyright (c) 2008 University of Art and Design Helsinki

EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,

PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR

PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF

LIABILIRY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING

NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS

SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.