MapWiki: A Map-based Content Sharing System for Distributed Location-dependent Information

Yuuichi Teranishi†, Junzo Kamahara‡, Shinji Shimojo†
† Cybermedia Center, Osaka University
Email: {teranisi, shimojo}@cmc.osaka-u.ac.jp
‡ Faculty of Maritime Sciences, Kobe University
Email: kamahara@maritime.kobe-u.ac.jp

Abstract— **In this paper, we propose a new map-based content sharing system 'MapWiki' for ubiquitous content distribution. In MapWiki, users can publish location-dependent information on the map as Wiki contents which contains links or inline images of the objects in the ubiquitous environment by simple formatting rules. The status change of the content are updated on the map in real-time which enables users to communicate between real field and virtual environment. Publishers and receivers are authenticated using social network to ensure the trustfulness of the content. We have implemented MapWiki on Google Maps and evaluated its effectiveness. We also propose a distributed P2P-based content management method to manage location-dependent information of GMapWiki. In this content management method, each user stores his or her own location-dependent content in the local computer. This P2P-based content management method realizes scalability, flexibility and privacy of the GMapWiki system. We have enhanced a P2P networking method called 'LL-Net' to implement a prototype system and confirmed its feasibility.**

Index Terms— **MapWiki, Ubiquitous content distribution, P2P, Agent system, Location-dependent information, LL-Net**

I. Introduction

Recent technological advances in portable computing devices and wireless communications have made it possible to realize the ubiquitous computing environment. In the ubiquitous computing environment, everyone can seamlessly publish and receive content at everywhere. In such environment, the importance of the open, distributed content distribution, so to speak, 'word-of-mouth content sharing' increases rather than existing closed, centered content distributions. Followings are examples of such word-of-mouth content distribution in the ubiquitous computing environment.

- **Provide content to the real field**
  Contents are delivered to the users in the real field according to the user situations. For example, deliver a reputation information about the restaurant located in front of the user. If an ingredient for the supper is missing, a message asking to buy it is delivered to the family member who firstly comes close to the supermarket.
- **Provide content from the real field**
  Contents are created by the users in the real field according to the user's current location or status and send to the other users. For example, if a car accident occurred in front of the user, he or she sends the picture to the related people around there.
- **Collect content from remote user**
  Contents are collected from remote users in the real field or in the Internet. For example, disaster information about the place where user is currently heading for are collected from the user who are living there. Picture image data of a certain location is corrected from the users who have the picture of that location.

As shown above, in the ubiquitous environment, there is a possibility that various kind of new applications can be realized by the word-of-mouth content sharing between real field and virtual environment (e.g. users connected to the Internet).

Many applications have been developed to collect and arrange the information about real field. But it is not clear to realize ubiquitous content sharing mentioned above. Our goal is to realize interactive, transparent and seamless ubiquitous content sharing between real field and virtual environment.

In this study, we propose a new content sharing system called 'MapWiki' to publish and distribute the ubiquitous content on the shared map. Using MapWiki, users can publish the location-dependent information just by clicking on the map and query them using their geographic coordinates. Collaborative work can be realized by notifying the added information to the user around at the position in the real field and vice versa.

So far we have implemented MapWiki system on Google Maps, called 'GMapWiki' on centralized server architecture. Furthermore, to solve the several problems of the centralized server architecture, we are now re-implementing content management subsystem of GMapWiki on a P2P-based architecture. Each user can manages his or her own location-dependent information of MapWiki in the local computer. We have already implemented a prototype and confirmed its feasibility.

In this paper, the next section describes the basic concept of MapWiki we propose here. Section III describes our current MapWiki implementation on GoogleMaps. Section IV describes the P2P-based content management method we propose and its prototype implementation. Section V describes the related works. Section VI discusses the limitation of our framework and some hints on how to solve them. Section VII is a conclusion.

II. MapWiki

A. Requirements

To realize the ubiquitous content sharing mentioned in the previous section, following requirements must be satisfied.

- Easy publications
  Users cannot use fully featured tools in the real field to create contents. To reduce the difficulty of publishing contents, users must be able to create content in simple and easy way.
- Real-time communication
  To communicate between users in the real field and virtual environment, they need to obtain the latest, real-time information.
- Trustfulness of the content
  Users need to judge whether the creator and the receiver of the content are reliable.

We propose a new content sharing system to satisfy the requirements shown in above.

B. Basic concept of MapWiki

Content management systems called WikiWikiWeb (hereafter, Wiki) has been used widely in the recent Internet. Wiki is a content management system to edit, share, publish information on the Web. Wiki has following features.

- The documents consists of two or more pages distinguished by the identifier called WikiName are published.
- The pages are editable from the Web browsers.
- Simple and intuitive mark-up language.

Wiki is suitable for the cooperative works for two or more users. A lot of Wiki engines are developed and distributed on the Internet. Many sites use it for the cooperative works and communications.

In this study, we propose a new content sharing mechanism that realizes the word-of-mouth type content sharing in the ubiquitous computing environment called 'MapWiki', an Wiki system on the shared map. MapWiki has following features (Fig.1).

- Wiki contents are located on map
  Anyone can add and edit Wiki content from anywhere and to any place on the map. A person or a object in the real field is treated as an Wiki content.
- Immediate updates of the content
  Additions, changes and movements of the Wiki contents are reflected immediately on the map.
- Identify of the content sender and receiver
  Users are authenticated to identify content sender and receivers.

As an Wiki system, anyone can add and edit contents on the map. Users can publish content intuitively and easily by Wiki mechanism. Only an Web browser is required for content publications. Using Web browser, portable terminals such as PDAs, cellular phones can publish a content. Here, we assume that users can use 'full-functioned' web browser, which is realistic on current cellular phone technology. Information about person and object in the real field can be viewed as an Wiki content. By reflecting the changes of the Wiki contents immediately on the map, collaborations between the user in the real field and virtual environment are realized. The content publishers, editors and receivers are identified and their access rights can be controlled to achieve the trustfulness of the content.

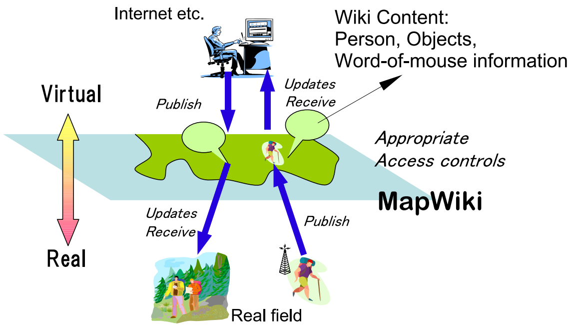

Figure 1. The basic concept of MapWiki

Figure 2. An example of adding content on MapWiki

Fig.2 shows an example of adding new MapWiki content. As shown in this figure, user can add new content on the map at the position where the user clicked. Added content is shared with other users immediately.

In addition, MapWiki treats not only user inputs from a browser but also automatically generated data from the real field such as sensor information. For example, if a weather sensor generates a rainy data, overlay image of rain is displayed on the map. It is possible to display the current position of the user according to the data from GPS which user carries.

C. Location-Dependent WikiName

In MapWiki, people and objects in the real field are treated as Wiki contents. We must consider how to identify to treat them. In Wiki, the identifier called WikiName identifies the content and is used as a anchor description of the link between contents. But in MapWiki, there may be multiple contents that have same name. For example, many buildings may be called 'Library' in one country.

The geometry information can be used as an identifier since buildings or facilities do not change their geometries. But it is difficult to memorize the geometries since they are expressed in long format, for instance, 135.394038209. In this case, copy and paste operations

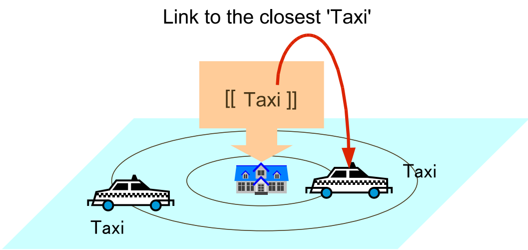

Figure 3. Location-Dependent WikiName

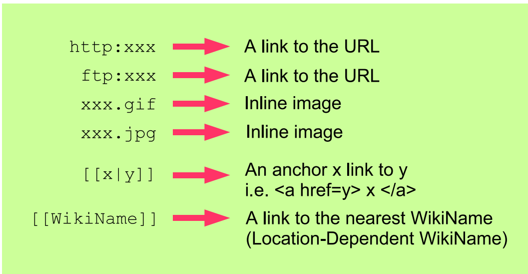

Figure 4. Example text formatting rule of MapWiki

are necessary to describe links. Moreover, the geometries cannot be a link identifier for people since their locations are changed frequently. It is also possible to use a unique ID for each Wiki content. But it is desirable to identify contents in more simple and easy way.

To solve this problem, we defined a special WikiName for MapWiki, called Location-Dependent WikiName (Fig.3). Location-Dependent WikiName indicates the nearest content with specified WikiName. By this identifier, users can easily describe links to the other contents without considering their geometries on the map.

Fig.4 shows an example text formatting rule of MapWiki, including Location-Dependent WikiName. Inline images, hyperlinks are formatted like normal Wiki. Hence user can describe contents simply and intuitively.

III. Current Implementation of MapWiki

In this section, current implementation of MapWiki is described.

A. Closeness-First Update

To realize a system on the ubiquitous computing environment, we have to consider that there are many low-spec terminal computers.

To reduce the load of the terminal computers, we propose Closeness-First Update (CFU) described in the Fig.5. As shown in the figure, CFU issues an query of the Wiki content to the database every time the display area of the map is changed. Retrieved contents are stored as cache data and displayed in order of their closeness from the center position of the map on display. Suppose the display area is inside the rectangle of $(x_1, y_1), (x_2, y_2)$, the query is issued for the contents with geometry x and

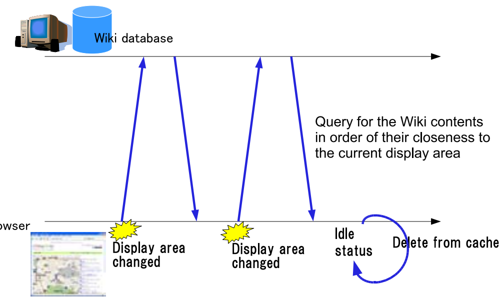

Figure 5. Closeness-First Update

y that satisfies the following condition.

$$x_1 - (x_2 - x_1)t \leq x \leq x_2 + (x_2 - x_1)t \quad \wedge$$
$$y_1 - (y_2 - y_1)t \leq y \leq y_2 + (y_2 - y_1)t$$

In above condition, t is a parameter. The larger the value of t, the wider the range of the updated area.

If the value of t is large, the required processes increase for the contents which are not in the display area but the movement of the display becomes smooth. Tuning of the parameter t required to get the best performance. In the idle status, the Wiki contents on cache are deleted in order of their distance. By these mechanisms, CFU can reduce the processing load and the required storage area for the client machine.

B. Access control according to social network

As described in the former section, MapWiki authenticates user and identify the sender and receiver of content.

To protect user privacy, it is not desirable not to notify user's position to other people without limitation. Besides, there is a demand to limit to spread recommendation information to the acquaintances. But it spends time and effort to specify access limitations for every contents. Alternatively, like UNIX systems, it is possible to specify the group names which are prepared beforehand. But as for assumption here, everyone can be a user of the system and is informal, the united group management is difficult.

Therefore, we decided to utilize social network to form groups by user initiations. The social network we call here is the human network formed by the users acknowledging each other. By social networks, users can form their own groups like families, friends, colleagues, etc. Hence it is suitable for the grass-roots group formation which is required in this system. In the implementation described in the section III-C, there are three access control levels: Private(user only), 1 Hop(directly connected users only), Public (for all users).

C. An Implementation: GMapWiki

We have developed a prototype system of MapWiki using Google Maps API [1], called 'GMapWiki' [1]. Google Maps API provides a smooth interface of the

[1]GMapWiki is running at following URL experimentally: http://gmapwiki.org/

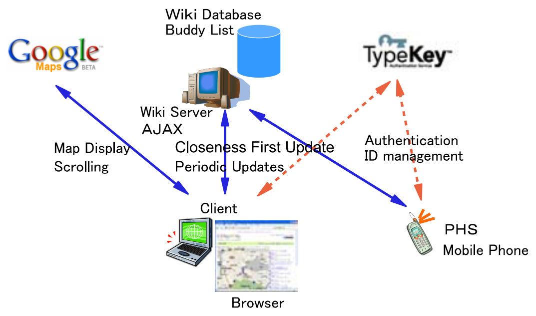

Figure 6. System architecture

```
<?xml version="1.0"?>
<rdf:RDF
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos#"
 xmlns:="http://gmapwiki.org/rdf-mapwiki/"
 xml:lang="ja">
  <rdf:Description
   rdf:about="urn:x-geo:135.524768,34.81518">
    <geo:long>135.524768</geo:long>
    <geo:lat>34.81518</geo:lat>
    <name>Information Science and Technology</name>
    <description>Research on Information
    Science</description>
    <category>info</category>
    <owner>user-1</owner>
    <access>1 Hop</access>
    <last_modified>
    Sat Aug 06 18:48:26 JST 2005
    </last_modified>
  </rdf:Description>
</rdf:RDF>
```

Figure 7. An example of MapWiki RDF

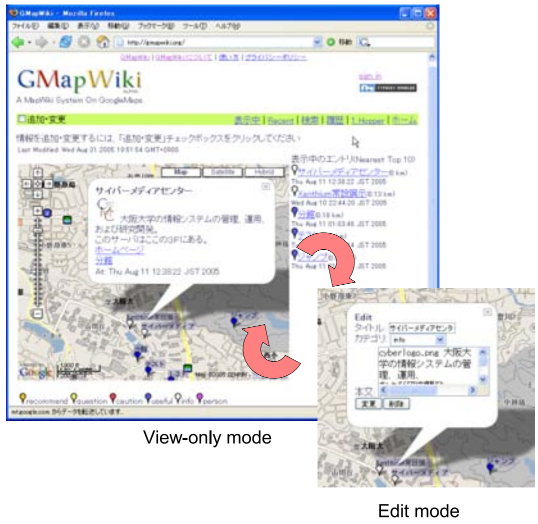

Figure 8. An example of GMapWiki interface

Figure 9. PHS interface example

world map which uses asynchronous processing of the HTTP called 'Ajax' which is now a popular Web site development technology. To follow the Google Maps API, the client program of the GMapWiki is implemented using JavaScript. GMapWiki does not require a special program but a web browser.

As a user authentication mechanism, we utilized a single sign on system called 'TypeKey' [2] which is widely used for Blog authentication.

We have defined a RDF schema for GMapWiki content. Fig.7 shows an example. In this RDF schema, an URN namespace 'x-geo' is used. Hence each Wiki content is expressed as a metadata for a location. Each Wiki content is represented as a 'metadata of the specified location'. There is a standard geographic data representation format called G-XML [3]. It is a future work to take a correspondence between such standard forms and our RDF format.

The 'description' tag contains the Wiki representation itself. If the RDF is retrieved in the 'edit mode', it is row Wiki data. In the 'view-only mode', it is formatted and escaped HTML page data.

Fig.8 is an example of GMapWiki content. User can easily add/edit the content since view-only mode and edit mode of the content can be changed seamlessly. Users can register current position or Wiki content from real field by this interface.

For the real field user, we have implemented an interface to register and retrieval for cellular phone with GPS and PHS. Fig.9 shows the example of the PHS interface. In this interface, user don't have to set the user's current position to publish or query. Thus cellurar phone user can skip the map clicking behavior.

Fig.10 shows an comparison between CFU and naive implementation which retrieves all content to the client at once('Naive' in this figure) on GMapWiki. 'Start up' shows the process time for start up and 'Display' shows the process time for updating displayed map. The value of the parameter t mentioned in III-A is set to 0.2. As shown in this figure, the process time for updating map only takes 0.5 second for 100,000 contents, approximately. On the other hand, 'Naive' method takes more than 2 minutes to start up when there are 1000 contents. The browser hangs up over 10,000 contents. In each evaluation, we used the client computer with Pentium M 1.7GHz CPU and MS Internet Explorer 6. We used server computer with Pentium4 3GHz CPU.

IV. P2P Implementation

A. Advantage of P2P Implementation

As shown in the previous section, current GMapWiki implementation based on the centralized server archi-

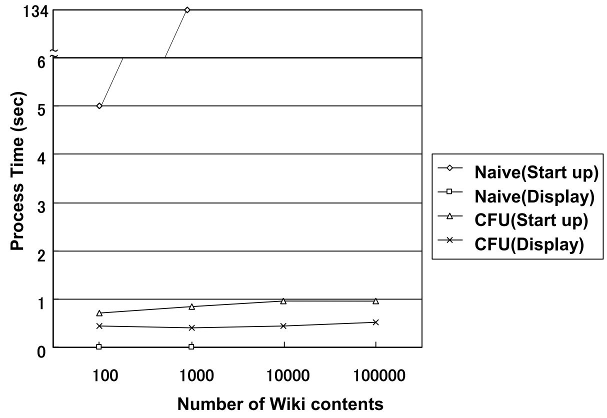

Figure 10. An evaluation of CFU

tecture. In this architecture, it may cause heavy CPU and network traffic load since all client accesses are concentrated to the server. In addition, it requires huge storage size to manage all geographic information that covers entire earth.

Furthermore, in the GMapWiki, explicit publishing behavior by user is required to store the new content. User need to upload the Wiki content to server to share the information. Therefore, newest information may not be reflected to the GMapWiki world even if the information is frequently updated on the user terminal. For example, user profile or sensor information may be updated frequently on the terminal computer but it is hard to upload every time the data is changed. Moreover, private information such as user profile should be under control of the user. Privacy leakage problems may occur if the centralized server is cracked in the client-server architecture.

To solve these problems, we are developing a P2P-based distributed content management subsystem for GMapWiki. The advantage of P2P-based implementation is following:

† **Scalability**
In the P2P architecture, content accesses are not concentrated to one server. Thus heavy load of the server can be avoided. It is easy to follow the increases of the number of the terminal computers. The more the terminal computers participate, the more the CPU power and storage size of the P2P network increases.

† **Flexibility**
The peers/users can join and leave freely. For example, a sensor node can join the GMapWiki P2P network without registering to the server. In addition, implicit information publishing can be achieved. That is, for example, weather sensor can publish their sensor data automatically. Picture images can be put on the map based on the geographic data embedded in the image (e.g. Exif with GPS data) which are stored on user's terminal computer.

† **Privacy**
Users are free from a proprietary, centralized server. Users can keep their private information on the local computer. All private information is under control of the owner. Users need not to worry about server cracking nor illegal use of private information.

B. Basic Design

To execute a content retrieval according to the locationin the distributed environment, we utilize the LL-Net(Location-Based Logical Network) [4], which our research group proposed. LL-Net is a P2P overlay network based on geographic coodrinates of users, nodes, and so on. LL-Net achieves efficient propagation of location-dependent queries. Fig.11 shows the basic notion of the LL-Net. In the LL-Net, the entire world is divided into the areas as square region. Multiple areas are hierarchically grouped into higher-level area and link to the neighbor areas are maintained for each peer. The region query is forwarded using these links. In the smallest area, flooding is used to propagate the query. Refer the paper [4] for more in detail.

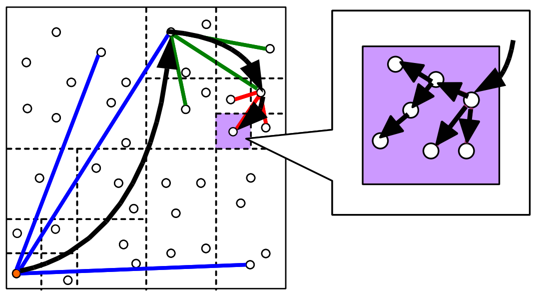

Figure 11. LL-Net: Location-based Logical Network

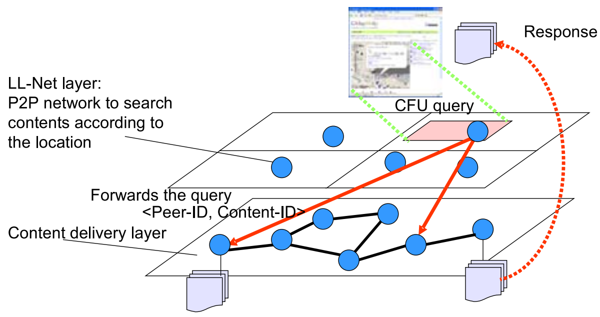

Figure 12. P2P-based GMapWiki Implementation

In our design, as shown in Fig.12, there are two layers of peer networks: one is LL-Net based location index layer, the other is content delivery layer. Each location index holds the Peer-ID of the content holder peer and the Content-ID which is unique in each peer. If CFU query is issued in the GMapWiki, then the query is forwarded to the LL-Net. If a content exists in the specified range, the Peer-ID and Content-ID is forwarded from LL-Net layer to the content delivery layer. In the content delivery layer, the peer responds which have the Peer-ID and searches local content with the specified Content-ID. In this architecture, P2P network only shares the index data of the content. Thus the privacy of the content itself is secured.

In our research group, a P2P-based mobile agent system called 'PIAX' [5] has been developed. PIAX implements LL-Net as a P2P overlay module. We already implemented an early prototype of the P2P-based content management subsystem (but not connected to the GMapWiki GUI yet). In this prototype, index layer and content delivery layer are implemented as agents of the PIAX. The experiment using this prototype shows that it takes approximately 2 seconds to retrieve a content from 1000 contents in the 10 peers (i.e. each peer has 100 contents). In our estimation, it is possible to shorten the retrieval time because the prototype is not optimized yet. But as mentioned above, P2P implementation has an advantage even if it is slower than centralized implementation.

V. Related Works

There are many approaches to arrange information on the Web and map. In the paper [6], a coding method called 'geocoding' is introduced to describe mapping between web information and map information. A system called Blogmapper [7] can generate a map with anchor points by RSS or Blog entries with location information. These can be used as technologies for mapping virtual environment and real fields. But neither can realize ubiquitous content sharing enough.

On the other hand, services such as GeoWiki [8], Kakiko-Map [9] are known as map systems that anyone can edit information. These systems can share the information on the maps but they cannot treat immediate updates of the status of the real field. Furthermore, the content sender and receivers are not connected to the real world.

After Google Maps API is released, many Google Maps based content publishing service have been appeared. For example, mapli [10] and Hatena map [11] can treat content publishing by each user. But these systems does not treat real-time updates to collaborate between users in the real field and virtual environment. Furthermore, these systems assume that all content are stored in the centralized server.

VI. Discussion

GMapWiki is designed mainly focused on the practicality. In the practical point of view, it is effective enough for small scale of users. Since the content retrieval is executed asynchronously, user interactions to the maps are not interrupted by the CFU queries.

As described in the former section, we implemented early P2P-based content management mechanism for GMapWiki. In our architecture, index information is shared by the P2P peers. But there is no guarantee that peer doesn't illegally rewrite the index information. If index information is illegally rewritten by the malicious user, there is a possibility of the denial of service. To keep the entire service securely, there should be an authentication and authorization mechanism of the peers. We are planning to implement PKI-based peer authentication mechanism.

We are still implementing the P2P-based GMapWiki. As described in the former section, GMapWiki uses GoogleMaps API and JavaScript. The browser security model dictates that the target of `XMLHttpRequest`, the core method of asynchronous content retrieval, and the page itself must have the same domain. But CFU content retrieval should be issued to the peer which stores user's own information, that is, user's local computer, not GMapWiki site. This violates the browser's security model. To avoid this problem, currently we are planning to apply JSONP [12] technique to issue cross domain data retrieval message from GMapWiki script to the user's own peer.

To realize the intelligent content delivery to the real field, it is necessary to define semantic relation between Wiki contents. For example, to send a message asking to buy an ingredient for the supper to the family member who firstly comes close to the supermarket, we need to define the relationship between family member and the message content. We are planning to define a Wiki formatting rule to describe such semantic relationship between contents. We need to consider how to describe such relationship. Semantic Web framework might be suitable to process such relationship.

In GMapWiki, social network is mainly used for the purpose of the group formation for the access control. But there is a possibility to utilize social network as a person based filtering mechanism. For example, users evaluate contents (e.g. add score to the each content) on GMapWiki and only the valuable contents (e.g. the content with high score) can be propagated wide range.

Moreover, in GMapWiki, the tendency of the user is reflected to the published content or viewed content. Thus there is a possibility of new applications, for example, automatic community formation of similar content publishers, intelligent content recommendation system based on the published content, and so on.

VII. Conclusion

In this paper, we have proposed a new map-based collaboration environment 'MapWiki' for ubiquitous content distribution. We have implemented a MapWiki system 'GMapWiki' using Google Maps API. Using GMapWiki, users can add or edit location-dependent information on the map as an Wiki contents by simple formatting rules. The status change of the content are updated on the map in real-time which enables users to communicate between real field and virtual environment.

GMapWiki can be used for a user interface of the word-of-mouth content authoring tool, a visualization tool of the sensor information, a browser of a user tracking, and so on.

To solve the problems of current GMapWiki implementation as a centralized server, we also proposed a P2P-based content management subsystem implementation.

We have enhanced a P2P networking method called 'LL-Net' to manage the distributed GMapWiki contents and confirmed its feasibility using the prototype. We are still working on the P2P-based GMapWiki implementation.

ACKNOWLEDGMENT

This work was supported in part by Research and Development Program of Ubiquitous Network Authentication and Agent (2003), The Ministry of a Internal Affairs and Communications (MIC), Japan.

REFERENCES

[1] Google, "Google maps api," http://www.google j.com/apis/maps/.
[2] six apart, "Typekey authentication service," http://www.sixapart.jp/typekey/.
[3] G-XML Project, "G-XML PROTOCOL 3.1," Database Promotion Center, Japan.
[4] Y. Kaneko, K. Harumoto, S. Fukumura, S. Shimojo, and S. Nishio, "A location-based peer-to-peer network for context-aware services in a ubiquitous environment," in *Proceedings of 2005 Symposium on Applications and the Internet Workshops*, Feb. 2005, pp. 208–211.
[5] PIAX Development Team, "PIAX - A P2P Agent Platform," http://piax.org/.
[6] K. Noaki and M. Arikawa, "A geocoding method for natural route descriptions using sidewalk network databases," *W2GIS 2004*, pp. 38–50.
[7] J. Harlan, "Blogmapper - map your blog and blog your map," http:// blogmapper.com/.
[8] R. Fairhurst, "Geowiki - a map which you can annotate," http:// www.geowiki.co.uk/.
[9] R. Manabe, ""kakiko map" - an internet mapped information board system," The 8th International Conference on Computers in Urban Planning and Urban Management, Sendai, Japan (2003).
[10] Cirius Technologies, "mapli.jp - social map service," http://mapli.jp.
[11] Hatena, "Hatena::map," http://map.hatena.ne.jp.
[12] B. Ippolito, "Remote JSON - JSONP," http://bob.pythonmac.org/archives/2005/12/05/remote-json-jsonp.

Yuuichi Teranishi received his ME and PhD degree in Information and Computer Sciences from the Osaka University in 1995 and 2004. He is currently an Associate Professor of Applied Information Systems Laboratory of Cybermedia Center, Osaka University, Japan. He was an employee of NTT, Japan from 1995 to 2004. His current research interests include the availability and security of ubiquitous content distribution systems, especially on the peer-to-peer architecture. He is a member of the Information Processing Society of Japan and an Organizer of the Special Interest Group on Distributed Processing System (SIG-DPS).

Junzo Kamahara received his ME and PhD degree in Information and Computer Sciences from the Osaka University in 1994 and 1999. He is currently an Associate Professor of Faculty of Maritime Sciences, Kobe University, Japan. His current research interests include the location services for mobile devices on the Internet, social filtering and recommendation with multimedia database and so on. He is a member of the Institute of Electronics, Information and Communication Engineers (IEICE) and an Organizer of the Data Engineering Technical Group, He is also a member of the Information Processing Society of Japan, ACM and IEEE.

Shinji Shimojo received the ME and a DE degrees from Osaka University in 1983 and 1986, respectively. From 1986 to 1989 he was an Assistant Professor in the Department of Information and Computer Sciences, Faculty of Engineering Science, Osaka University. Since 1998 he was an Professor in Cybermedia Center, Osaka University. He was engaged in the project of object-oriented multimedia presentation system called Harmony. His current interests cover wide diversity of networked applications such multimedia database, grid computing, networked virtual reality and so on. He is a member of ACM and IEEE.