

An Environment for Collaborative Content Acquisition and Editing by Coordinated Ubiquitous Devices

Yutaka Kidawara
Interactive Media and Content
Group, NICT
3-5 Hikaridai, Seikacho,
Sorakugun
Kyoto, Japan
kidawara@nict.go.jp

Tomoyuki Uchiyama
Graduate School of
Informatics, Kyoto University
Yoshida-Honmachi, Sakyo-ku
Kyoto, Japan
tomoyuki@dl.kuis.kyoto-
u.ac.jp

Katsumi Tanaka^{*}
Interactive Media and Content
Group, NICT
3-5 Hikaridai, Seikacho,
Sorakugun
Kyoto, Japan
tanaka@dl.kuis.kyoto-
u.ac.jp

ABSTRACT

Digital content is not only stored by servers on the Internet, but also on various embedded devices belonging to ubiquitous networks. In this paper, we propose a content processing mechanism for use in an environment enabling collaborative acquisition of embedded digital content in real-world situations. We have developed a network management device that makes it possible to acquire embedded content using coordinated ubiquitous devices. The management device actively configures a network that includes content-providing devices and browsing devices to permit sharing of various items with digital content. We also developed a *Functional web* mechanism for processing embedded web content in the real-world without a keyboard. This mechanism adds various functions to conventional web content. These functions are activated by messages from a *Field* in a content processing device. We construct a practical prototype system, which is simple enough for children to use, that we called the "Virtual Insect Catcher". Through a test with 48 children, we demonstrated that this system can be used to acquire embedded web content, retrieve related content from the Internet, and then create new web content. We will also describe the proposed mechanism and the system testing.

Categories and Subject Descriptors

D.2.11 [Software Engineering]: Software Architectures; H.3.4 [Information Systems Applications]: Communications Applications; H.5.2 [HCI]: User Interfaces; B.4.1 [Hardware]: Data Communications Devices

General Terms

Algorithms, Design, Human Factors, Languages

Keywords

Multiple device operating, embedded content, RFID, Functional web, Ubiquitous Computing, Ubiquitous Network

^{*}Dr. Tanaka also works at Kyoto University
Copyright is held by the International World Wide Web Conference Committee (IW3C2). Distribution of these papers is limited to classroom use, and personal use by others.

WWW 2005, May 10-14, 2005, Chiba, Japan.
ACM 1-59593-051-5/05/0005.

1. INTRODUCTION

Information, communication, and entertainment services are increasingly becoming available indoors and out because various network-accessible devices now provide wide ranging coverage in the real-world. Such devices rely on recently developed ubiquitous computing technologies. The evolution of communications and device technology will soon enable us to achieve a next-generation communications tool that can be thought of as a ubiquitous Internet. Web content will be accessible anywhere at any time through mobile phones or mobile devices based on mobile Internet technologies. While current mobile technology enables us to access Internet servers anywhere at any time, users still cannot directly obtain local information from physical objects even though they might want to seek and store such information in real world situations. We have to consider various keywords so that we obtain this information using the WAP/i-mode browsers of mobile phones and small web browsers of mobile devices in the real-world.

Locally ubiquitous computing technologies are widely used in the distribution of physical objects via information directories on a network. RFID tags and optical ID codes have changed how such real world information is accessed. The tags and codes attached to physical objects enable us to bind actual objects and digital information (e.g., Web content, pictures, movies). Digital content will eventually be ubiquitously embedded throughout the real world through these technologies.

Such technologies, however, are being developed with a focus on the development of an infrastructure and a framework for dealing with several types of devices. Far less attention has been paid to what content should be available for browsing and what information from physical objects should be accessible in the real world.

Even if ubiquitous computing enables us to acquire digital information regarding physical objects anywhere at any time, we might not be satisfied with the procedure to access such contents. In practice, when we try to access information concerning physical objects we often want to compare related information obtained from many sources.

In many cases, we cannot obtain sufficient information because of limitations with the capabilities of the mobile device at the location where the physical objects are dis-

played. Information-rich digital content cannot easily be understood through the limited capabilities of mobile devices. Simple content is most suitable for display on mobile devices, but users often cannot acquire the information they need in this way. We believe that two kinds of content operations are necessary to enable effective access to embedded digital content through ubiquitous networks. First, we need a content acquisition operation. Second, we need a data browsing and comparison operation. A requirement is for each operation to be performed by various devices. For instance, we envisage the following scenarios.

Scenario 1 (data acquisition): A user finds her favorite shoes at a shoe store in a mall. She obtains information by reading the RFID tag attached to the shoes with her personal device. Basic information (e.g., price, some product information, and recommendations) can be stored in the device. In this way, the user can acquire information on all her favorite shoes throughout the mall.

Scenario 2 (data browsing and comparison): The user meets her friends and goes to a cafe. They each transfer acquired information from their personal devices to a large-screen display device on the table via the cafe's network. They can then compare information regarding their favorite shoes. Related content is searched for in the mall databases and on the Internet, and then the information acquired from the various sources is compared.

In practice, such operations may be difficult to perform effectively because it is awkward to manipulate web content using a keyboard and mouse in such situations. A web content processing mechanism that did not require keyboard input was therefore needed. With this in mind, we developed the *Functional web* mechanism, which attaches functions to web content. Functions attached to web content operate effectively and autonomously on ubiquitous devices.

In this paper, we propose the concept and implementation of collaborative content acquisition and an editing method that can be applied through coordinated ubiquitous devices.

Section 2 explains our objectives and assumptions regarding embedded web content and access through ubiquitous devices. Sections 3 and 4 describe the concept of the *Functional web* mechanism and a mechanism to translate conventional web content to *Functional web* content. We also describe implementation and testing in Section 5. Section ?? discusses related work and Section ?? summarizes our findings and future work.

2. OBJECTIVES AND ASSUMPTIONS

A ubiquitous network requires a more complex network configuration. It is not only global and connected with the Internet, but is also a local network. Various devices have network connection capabilities and distribute various kinds of content. We expect that information in embedded devices will generally be distributed to other devices through peer-to-peer networks, and this information will be moved to other locations by users as needed. In this section, we describe our assumptions and objectives with respect to embedded web content operations.

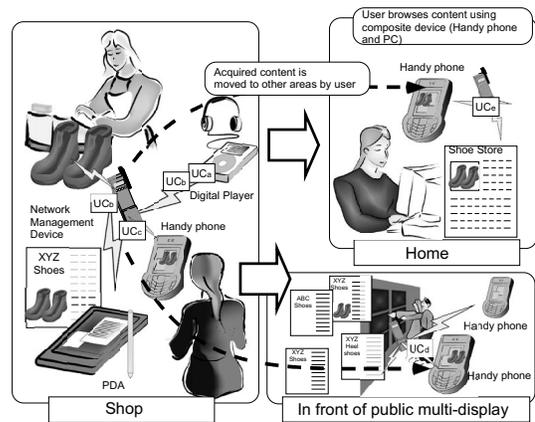


Figure 1: Real-world content acquisition and related content retrieval on the Internet

2.1 Assumptions

In this research, we assumed that two kinds of content operations would be necessary in a ubiquitous network. First, an operation would be needed to acquire embedded content. We expected that a user would acquire embedded web content in a real-world situation and then move the content to various destinations. The user would then obtain information on physical objects as web content from RFID tags or optical IDs in the real world. Appropriate equipment would provide a unique ID individually bound to each item of web content. The user would then use a network accessible personal device to obtain the embedded web content from a content server when the server found the content corresponding to the ID. Second, content would be browsed, stored, and moved about by personal devices. As a result, embedded content would be moved to other areas and manipulated according to user context. Each user would browse, compare, and apply the content by using various devices belonging to ubiquitous networks.

2.1.1 Acquisition of Embedded Web Content

Ubiquitous computing technology enables us to embed various devices and small computers in real-world situations and connect these devices through ubiquitous networks. Such devices provide web content in the form of digital data, and we call such content embedded web content. Embedded web content is local content that we might not be able to access from the Internet, but that can be provided to other devices as information concerning physical objects, having both regional and temporal specifications.

2.1.2 Distribution of Embedded Web Content

Users acquire embedded web content through personal devices that can connect with embedded devices via local networks. A local network is not necessarily connected to the Internet and might only be connected to a content server in the immediate area. That is, users have to be within this area to obtain the content. Such web content might come with special features depending on which users are within the area at a given time. The content server might be able to provide different levels of information service to users who have the embedded web content. The web content can thus

function as an access key for location-dependent services as well as digital documents. For example, the content can become a ticket for a particular content service or a key to an actual entrance gate. The embedded content is shared among various devices and applied to a variety of purposes according to user context. The user context should involve not only user profile information, but also any applicable device information. For example, users can provide their favorite items and browsing styles to the content servers in a local area so that content can be processed effectively. Figure 1 outlines the procedure. A user provides the user context identifier (UC_x) of a pair of shoes in a shop. She has several kinds of context IDs ($UC_a - UC_d$). Each UC is bound to a particular device and browsing style. Content servers can provide suitable digital content to a corresponding device when the user provides a particular UC. She may see a digital image on her handy phone (i.e., cell phone), browse a short web content item on her PDA, or listen to a voice announcement on a digital media player. She can store the content on her handy phone and move on to other areas. She may compare several content items with her friends on multiple displays and retrieve related web content from the Internet at home.

2.2 Objectives

The goal of our research is to develop a means of providing web content through embedded devices using a ubiquitous computing technology. That is, we wish to develop a practical method of disseminating embedded web content. We have two research objectives as follows.

2.2.1 Interactive Network Configuration

Users should be able to configure networks encompassing user devices and embedded devices in the real world interactively and quickly when embedded web content is shared among devices. Appropriate network management equipment will be necessary to manage connections among devices. This equipment will have to manage peer-to-peer connection of devices based on user contexts. Device selection should be done in a simple way since users cannot operate complicated user interfaces in real-world situations and can only provide a certain amount of identifying information to the environment.

RFID tags and optical IDs provide a useful and easily applied mechanism to distributing identifiers within a content distribution environment.

If a content-providing device or browsing device reads tags and IDs in real-world situations, the device searches for corresponding devices from the device management equipment and obtains all the device information regarding the devices with which it should communicate. In this way, shared content can be transferred to an individual device. A network management mechanism based on user contexts is needed to acquire embedded content in such real-world situations.

Thus, one of our research objectives is to develop such an interactive network management mechanism.

2.2.2 Web Content with Autonomous Functions

Embedded content is acquired and distributed among various devices. We therefore have to browse and process acquired web content on these devices. The content should also be easy to use to retrieve related content. The usage of embedded content depends on the processing capability

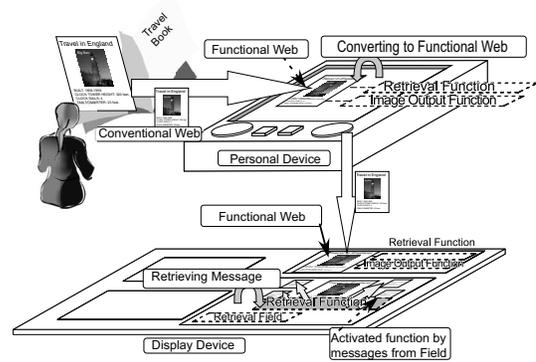


Figure 2: Adding function to the *Field* of content processing device

of each device. Content should be changed according to the purpose of its use by a processing device and the environment in which it will be used. Embedded web content is processed as an ordinary document that can be used not only as information but also as a query source for related content retrieval and part of a newly edited document. We attempted to attach some functions to web content so that the content would work autonomously on various devices. We called web content with an autonomous function *Functional web content*. The development of *Functional web content* and a corresponding management mechanism is our second research objective.

3. FUNCTIONAL WEB CONTENT

Even conventional web content involves scripts as functions. Java-script is the most widely used of these. Such scripts in web content can work with a browser to change the view of content and process predefined operations. Perl script, which can run on web servers, creates such views and functions dynamically. Current technology is based on the transmission of html or xml documents from server to client, which are bound to a particular URI and processed by the web browser. On the other hand, *Functional web content* is generated from conventional web content by an operating device. The device attaches functions that can not only operate in the content's original view, but can also search for and create other content. Functions attached to web content receive operation messages from individual operational fields on the device. An operational field is a lower layer where the *Functional web content* is located. For instance, if *Functional web content* is transmitted to a PDA, the PDA's screen displays several regions and the content within them as icons. Each region has functions that send messages to the *Functional web content*. We call the region where *Functional web content* can be processed a "Field".

3.1 Field for Adding and Activating Functions

If a user finds a physical object he is interested in, he can transmit the embedded web content from the object to his personal device by using it to read the RFID tag or optical ID. The obtained content may then be browsed on his device or transmitted to other devices. In this paper, we propose the *Functional web* mechanism as a means of directly operating embedded content.

If a device can convert conventional web content into *Functional web content*, several predefined functions are attached from the *Root Field*. The user can find other *Fields* in the *Root Field*. A *Field* can then send operating messages when the user moves the content into each *Field*. These messages are emerged from *Field*.

For instance, if a user decides to go on a journey after reading a travel book, he can transmit the embedded web content from the book to his device (Figure 2). His friend can likewise acquire other web content from travel posters. Their devices can then convert the conventional content to *Functional web content*, and this will be attached through a function from *Field*. The two friends can then go to a travel agent with this content stored in their devices. The converted content will then be transmitted to a travel navigation display device at the travel agent. The display device will show the content as an icon on a *Root Field*. When a user drags the icon onto a *Retrieval Field*, the *Field* can send a retrieval message to *Functional web content* with a specified search domain and additional keywords for travel. A function of the *Functional web content* is activated, acquires message parameters, and then retrieves the related content from the Internet or a local database. The results are then sent to the *Retrieval Field*. Subsequently, the *Field* displays the retrieved content on the screen.

In such applications, the retrieval function does not depend on the travel agent's device. If the user can create or select a *Field* on their device to convert *Functional web content*, he can manipulate content as he chooses.

3.2 Field Description Language

We defined a *Field description language* to describe the specifications for *Field*, which is described by XML. The language has four kinds of categories, as described below.

3.2.1 Field Appearance and Attribute Descriptions

Field is displayed as a rectangular region on the display screen. Therefore, it has attributes regarding its position and keywords.

A *position* tag describes the position of the original point, with *height* and *width* used as sub-tags. *fieldKeywords* describes the keyword set that explains the properties of *Field*. The *and* sub-tag explains the intersection of keywords. The *or* sub-tag explains the aggregation of keywords. Let us look at an example of the *Retrieval Field* attributes. Here, the original point for *Field* is (100,200), and its height and width are each 400 pixels. The query keywords are also shown ((kw1 AND kw2) AND (kw3 OR kw4)). Keywords are included in a query when *Functional web content* is put onto the *Retrieval Field*.

```
< position x="100" y="200" />
< size height="400" width="400" />
< fieldKeywords>
  < keywords>
    < and>
      < and>
        < keyword name="kw1" />
        < keyword name="kw2" />
      < /and>
    < or>
      < keyword name="kw3" />
      < keyword name="kw4" />
    < /or>
  < /keywords>
< /fieldKeywords>
```

```
</or>
</and>
</keywords>
</fieldKeywords>
```

3.2.2 Field Configuration Description

The *conf* tag describes the layer configuration. The *parent* tag describes the lower *Field*, which is semantically higher. The *child* sub-tag describes the upper *Field*, which is semantically lower. Messages are sent from the lower layer to the upper layer. The following example shows that the parent *Field* is a *Root Field* as described in http://webboard.nict.go.jp/root_field.fdl. The child *Field* is described in http://webboard.nict.go.jp/image_stripping_field.fdl. This *Retrieval Field* lies between the *Root Field* and the *Image Stripping Field*. If functional web content is put onto an *Image Stripping Field*, *Functional web content* can retrieve related content and strip image data from the retrieved content.

```
< conf>
  < parent>< uri>http://webboard.nict.go.jp/
    root_field.fdl</ uri> </ parent>
  < child>< uri>http://webboard.nict.go.jp/
    image_stripping_field.fdl</ uri></ child>
</ conf>
```

3.2.3 Event Description

The *event* tag describes which event corresponds to which function. Currently, we have defined three kinds of events. The *changeSize* sub-tag defines a message that changes the size of *Field*. The *onField* sub-tag describes a function that the system activates when *Functional web content*, or conventional web content, is put on the *Field*. The *receiveMsgs* sub-tag describes functions activated by a received message sent when *Functional web content* is moved onto a *Field*. The following example shows that the *retrievalDone* function is activated when the size of the *Retrieval Field* changes. When *Functional web content* is put onto this *Field*, the *webRetrieval* function is activated. If conventional web content is put onto the *Field*, the *addField* function is activated.

```
< event>
  < changeSize func="retrievalDone"
    name="redisplayContent" />
  < onField name="putNormalWeb"
    func="addFunctions" arg="overRide" />
  < onField name="putWeb"
    func="webRetrieval" arg="" />
  < onField name="putField" func="mergeField"
    arg="aggregateKeywords" />
  < receiveMsgs func="retrievalDone"
    name="displayContents" />
  < receiveMsgs func="mergeKeywordsOffield"
    name="mergeKeywords" />
</ event>
```

3.2.4 Function Description

The *functions* tag describes functions activated by events. The *function* is a sub-tag that describes an individual function bound to an *event* tag. The *function* tag has an *addFunc*

tion tag and a *doFunction* tag as sub-tags. The *send* tag sends an *arguments* sub-tag to the *Functional web content* or other *Fields* from which a message was sent. The *addFunction* sub-tag attaches a function to the *Functional web content* or the *Field* from which a message was sent. The *doFunction* tag describes how data is to be processed. The following example shows how the *webRetrieval* function sends a *retrieval* message to *Functional web content* when the function is activated. As a result, *Functional web content* runs its own retrieving function autonomously when content is moved onto the *Retrieval Field*. The *addFunction* adds a retrieval function or replaces that of the *Functional web* when conventional web content is moved onto the *Field*.

```
<functions>
  <function name="webRetrieval">
    <doFunction as="send"
      message="retrieval"
      application="webboard">
      <bindings>
        <arguments>
          <argument name="fieldKeywords"
            value="AND"/>
        </arguments>
      </bindings>
    </doFunction>
  </function>
  :
  :
  <function name="addFunctions">
    <addFunction as="webRetrieval"
      useAPI="doGoogleSearch"
      wsdl="http://api.google.com/
        GoogleSearch.wsdl">
      <bindings>
        <arguments>
          <argument name="key"
            value="tags:searchkey"/>
          <argument name="start">0</argument>
        </arguments>
      </bindings>
    </addFunction>
  </function>
  :
</functions>
```

3.3 Web Content with Functions

Functional web content has attached functions from the *Root Field*. Two kinds of functions are attached to conventional web content. The *Retrieval Function* has a related content retrieval capability. This function works much like similar functions to retrieve content from the Internet or local database systems. The procedure depends on the attached functions. In many cases, the *Field* attaches Web service descriptions as a function. *Functional web content* converted from conventional web content works with various *Field* descriptions cooperatively and autonomously.

3.4 Functional Web Description Language

3.4.1 Keyword Description

The *keywords* tag describes keywords that explain content

specifications. The format is the same as for *Field description language*. The following example is similar to the example given in Section 3.2.1

```
<contentKeywords>
  <keywords>
    <and>
      <and>
        <keyword name="kw1"/>
        <keyword name="kw2"/>
      </and>
    <or>
      <keyword name="kw3"/>
      <keyword name="kw4"/>
    </or>
    <and>
  </keywords>
</contentKeywords>
```

3.4.2 Content Encapsulating Description

The *encapsulation* tag describes the original content. The original web content is encapsulated by the *content* sub-tag. The *original* sub-tag describes the URI where the original content is stored. In the following example, the original content is on <http://rw.nict.go.jp/beatle.xml>.

```
<encapsulation>
  <original>
    <uri>http://rw.nict.go.jp/beatle.xml <uri>
  </original>
  <contents>
    ..... Original content inserting here.....
  </contents>
</encapsulation>
```

3.4.3 Event Description

The description of the *event* sub-tag is the same as that for *Field description language*, except for the *onField*, *onPointer*, and *onClick* sub-tags. The following example is similar to the example given in Section 3.2.3. An *onClick* event occurs when a user clicks on or touches *Functional web content*. An *onPointer* event occurs when a pointer is on *Functional web content*.

```
<event>
  <onField name="iconify" func="iconify"/>
  <onPointer name="point" func="displayTips" />
  <onClick name="display" func="displayContent"
    arg="doubleClicked" />
  <onClick name="focus" func="focus"
    arg="singleClicked" />
  <onClick name="strip" func="strip"
    arg="dragged" />
  <receiveMsgs name="refresh"
    func="refresh" />
  <receiveMsgs name="Retrieval"
    func="webRetrieval" />
</event>
```

3.4.4 Function Description

The *function* tag is described in the same way as for *Field description language*. However, some of the functions may be modified by a particular *Field*. The following example is also similar to the example given in Section 3.2.4. *Functional web content* replaces the original *webRetrieval* function with a new function provided by the *Retrieval-Field* when the content is moved onto the *Retrieval Field*. *Functional web content* retrieves related content using a new *webRetrieval* function when the content is moved onto the *Retrieval Field*.

```

<functions>
  <function name="webRetrieval">
    <doFunction as="relatedContentRetrieval"
      wsdl="http://wbserver.nict.go.jp
        /retrieve_relatedcontent.wsdl"
      useAPI="tf_idf_searchAPI">
      <bindings>
        <arguments>
          <argument name="key" value="tags:searchkey"/>
          <argument name="start">0</argument>
          <argument name="maxResults">10</argument>
        </arguments>
      </bindings>
    </function>
    <function name="refresh">
      <doFunction as="refreshContent"
        useAPI="reloadContent"
        Application="webboard"/>
      <bindings>
        <arguments>
          <argument name="timeInterval" value="14400"/>
        </arguments>
      </bindings>
    </function>
  :
  :
</functions>

```

4. PROTOTYPE

We developed a network management device and a *Functional web* mechanism that can run on a WebBoard system. This enabled embedded web content distribution.

4.1 Functional Web Distribution Network

We have to select suitable devices based on each user's status to acquire embedded web content from physical objects. In real-world situations, a user generally cannot operate a complex interface without using a keyboard and mouse. Device configuration selection should be done as simply as possible. The network also has to be configured for usable devices. We therefore developed a network management device and a communications protocol for it based on RFID technology. If a user can configure a network connection, he can browse and use various composite forms of content on various devices through cooperative interaction between devices. Our network management device enables easy network configuration. The device can bind a unique RFID identifier to a particular network configuration for a virtual composite device. A user only has to provide each an RFID tag to the content server when acquiring content of interest

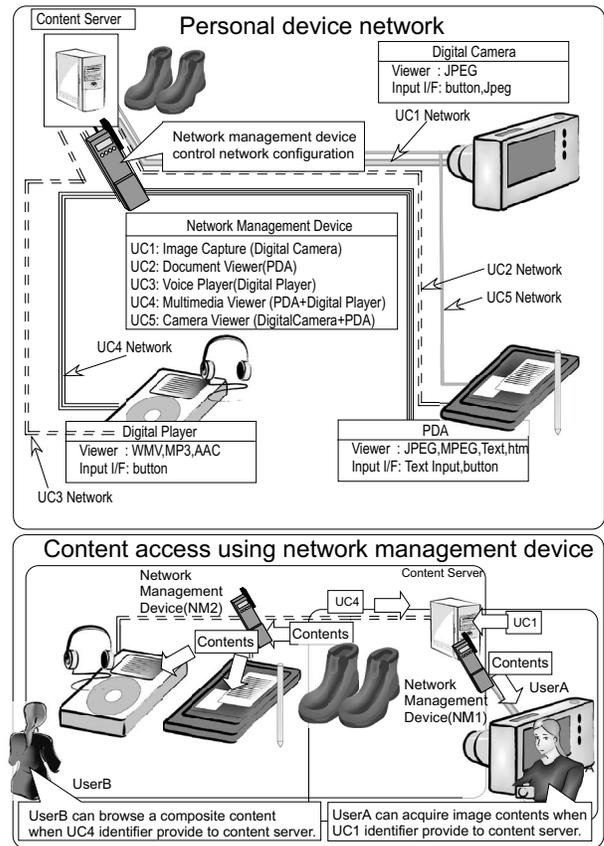


Figure 3: Active configuration of network using network management device

for his current device combination. Figure 3 shows an example where RFID tags are used. Each RFID tag stores an identifier regarding the individual user context (UC). The UC indicates the browsing device and purpose. UC1 is for a digital camera used as an image viewer, while UC4 is for a multimedia viewer consisting of a PDA with a digital player. If users present a UC to a content-providing device attached to physical objects, the device will read the UC and obtain content suitable for single or composite devices.

4.2 WebBoard System

We earlier developed a system, called the WebBoard system, consisting of a personal computer and a PDP display with a touch panel. The original WebBoard system could browse web content and retrieve related content through interactive operation. We modified the original WebBoard system to accommodate *Functional web content*. The new system inherits the functions of the original with the addition of *Fields* and *Functional web content* operation methods. This WebBoard system also has an RFID tag reader and a communications function with a Network Management (NM) device. The system searches other communicable devices bound to each identifier when RFID tags are read. The system can operate *Functional web content* directly through screen-touch control. The basic functions of the WebBoard system are detailed in Fig. 4. This figure shows an operational example of related content retrieval

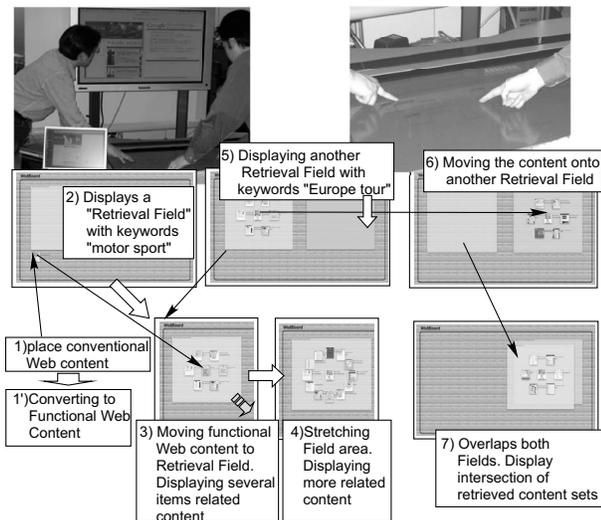


Figure 4: Example operating content on WebBoard

using two *Fields*. Each *Field* has a keyword set used by the retrieval function. The *RetrievalFields* attach the Google API retrieving function to conventional web content when the *RootField* converts this content to *Functional web content*. The content then retrieves its own related content using *Fields* keywords when the user drags it onto the *RetrievalField*. If the user stretches the size of the *Field*, the number of displayed content items will increase. *Functional web content* on overlapping *Fields* consists of related content items retrieved through the intersection of keyword sets of two *Fields*.

5. IMPLEMENTATION OF PROTOTYPE SYSTEM

We developed a more practical prototype system that we tested during the NICT open house. In this section, we will describe the implementation of the prototype of our Virtual Insect Catching System.

5.1 Prototype System

We developed a practical prototype system, which we called the Virtual Insect Catching System, for children to use who came to the NICT open house in July 2004. The system operates according to the following procedure.

1. Children could acquire embedded web content in the real world. The content was bound to RFID tags. The children could search, read RFID tags, and acquire content from the real world by using an NM device developed through this research. They could then go outdoors where an NM device and a PDA were attached to an insect cage. There they could acquire embedded web content about various insects from physical objects through the PDA on the insect cage when an insect image bound to an RFID was read by the NM device. The NM device searched a content server storing insect data, acquired suitable embedded web content, and temporarily stored this content. If the NM device found a PDA on the insect cage that could be

used to browse the content, the device started to search for that particular content. The NM device transmitted the acquired content after finding the PDA. As a result, children could capture the desired insect and put it into the insect cage.

2. In the next step, children moved to an operation table and wall display that had been developed based on the WebBoard system. The insect cage was put onto an RFID reader so that the children could move the acquired insect-related content on the WebBoard screen. The table-type WebBoard had a *Root Field* and a *Retrieval Field*. The *Root Field* converted conventional web content to *Functional web content* and showed an icon image. The center of the screen was the *Retrieval Field*. An appropriate insect icon, which had been converted to *Functional web content*, received the message regarding related content retrieval using Google API when the icon was moved onto the *Retrieval Field*. Users could retrieve content related to the captured insect and move to the wall-display WebBoard. This WebBoard had three *Fields*: the *Root Field*, *Image Splitting Field*, and *Editing Field*. Images were split and displayed when users placed related content on the *Splitting Field*. Users could also place the insect icon on the title area in the *Web Editing Field*, which was displayed as a title. The *Editing Field* is a composite *Field*. *Field description language* does not support the description of composite *Fields* that operate cooperatively. With this prototype, each *Field* worked individually and changed the content views on *Field*. The final output as normal web content depends on the prototype functions. Enabling such composite operation on several *Fields* remains for future work.

5.2 Prototype Testing

We tested the prototype system on July 31, 2004 on 48 children. The children were paired and each pair had to search for insect printed RFID cards in a laboratory using a network management device as an insect capturing net and a PDA on the insect cage. The children could acquire embedded web content from the insect printed RFID cards, which was similar to capturing the actual insect using a net. The acquired content was described by an RDF, which included a few keywords, a small amount of information about the insect, the filename of the icon image, and content providing device information. The PDA generated an html document from the acquired content. The acquired web content was transmitted to each child's PDA on the insect cage, which was somewhat like putting a captured insect into its cage. The children could also search for related web content from the Internet using the acquired embedded web content, which was much like searching for information from an encyclopedia. The children could also select several web content items and then merge the image content from these to create new web content. This procedure was like cutting pictures from a magazine and creating a poster for a presentation. Such edited web content can be published as normal web content on the Internet.

Our prototype system worked smoothly, and the children could correctly operate the WebBoard *Functional web* system. None of the operations, except for inputting comments, required use of a keyboard or mouse. The children could easily acquire embedded web content from physical materials,

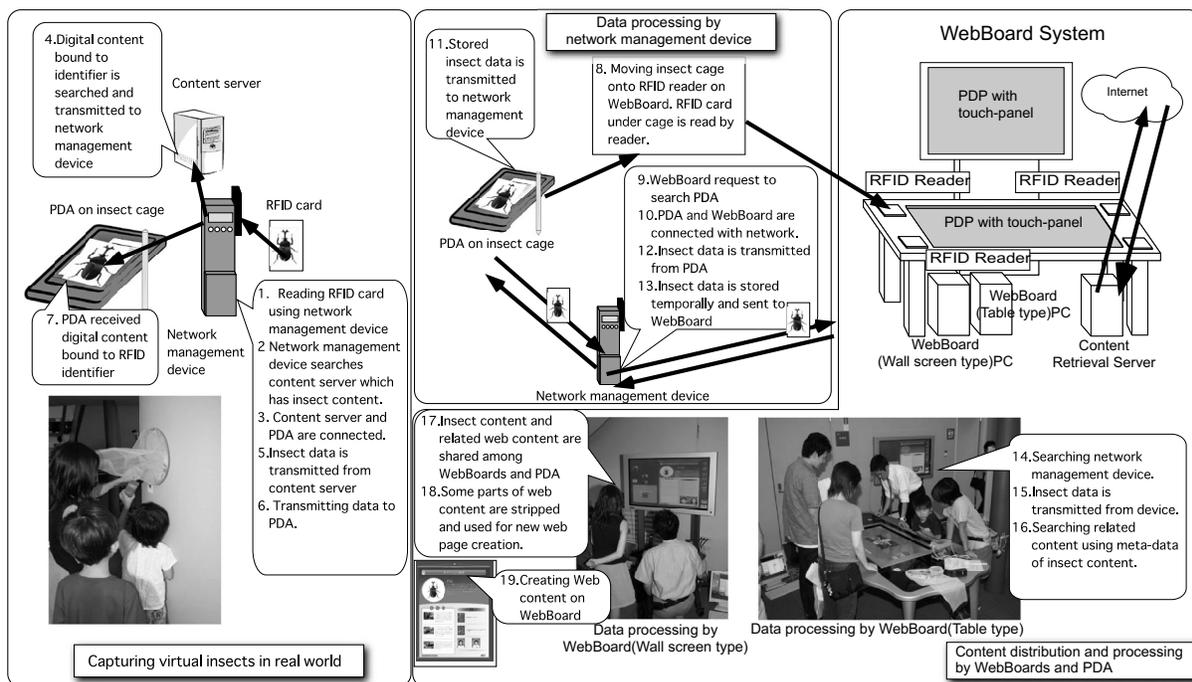


Figure 5: Digital content distribution in the prototype system

retrieve related content from the Internet, and create new web content.

However, one shortcoming of the current prototype became apparent. Some children asked to search for web content related to insects not included in the group used in the test. When this occurred, a keyboard and mouse were needed to search for this content directly. In the future, we hope to devise a way for *Functional web* to support a soft keyboard or other virtual input device functions.

6. RELATED WORK

In this paper, we focused on content distribution network management methods and functions for content operations using multiple devices in real-world situations. Our research objectives were to develop an actively configured network and content access control mechanism that enabled the fusion of real-world digital content and network services provided through ubiquitous networks. The technologies we discussed could run on our prototype and complemented our work. Our proposed mechanism worked on a peer-to-peer network. A variety of approaches have already been proposed in the ubiquitous computing area. The main issue has been the synthesis of network services. Wapplet [?] and STONE [?] focused on synthesizing network services. Their proposed technologies aimed at functional integration on a ubiquitous network. Our main objective was to create useful content management technology on a ubiquitous network, which was quite different from their objective.

We proposed a method of accessing web content in real-world situations. The WebBoard system enabled web content access and distribution based on personal devices and ambient displays indoors and out [?]. Each device changed its role as was necessary. Personal devices become content

browsers when users acquire embedded web content from an ambient display, and they become content providers when the users add stored content to the ambient display. The underlying communication is based on a peer-to-peer connection through ubiquitous networks.

Our developed application, "WebBoard", uses a device that browses and stores content on a board. The device has to have functions that enable communications, storage of content, and browsing capabilities. MediaBox [?, ?] and Personal Server [?] also have communications and content-storage functions. However, these are portable communications attachments mounted on ubiquitous devices. They only provide content to other devices through networks or wireless communications methods and do not allow the capability of browsing related content. The content is only conventional digital data in the devices. They have to be operated through conventional applications. The distribution and operating mechanisms are quite different from our proposed *Functional web*. The *Functional web* aims at useful access control technology for ubiquitous computing environments in the real world. *Functional web* on WebBoard has the potential of organizing material from mixed digital and physical environments.

This organization has recently focused on the Hypermedia research area. Physical Hypermedia [?] uses three kinds of tags that combine physical material with digital objects. Simple object tags and collectional tags are linked to the digital world. Tool tags are linked to operations in the digital world. HyperReal [?] provides a hypermedia framework to construct context-aware and mixed realities. The proposed model specifies the relation between real and virtual space. The Ambient Wood Journal [?] aims at facilitating learning experiments in outdoor environments. It can record children's activities on detailed logs in portable devices. The

recorded logs are displayed as journals on the web. This research is extremely interesting. The mechanisms may work on the *Functional web*. Our WebBoard may be provided with a comprehensive mechanism for the proposed hypermedia mechanism. *Functional web* on WebBoard should be of help in assisting us to organize physical material in the real world.

Several content access technologies for real-world applications have been proposed in the human interface context [?, ?, ?]. These technologies have focused on content operations and browsing mechanisms. For example, SmartSkin [?] focused on providing an interface for cooperative content operations using multiple devices in real-world situations.

Multiple-device content browsing has also been proposed within the context of frameworks [?, ?, ?]. A special proxy was developed for WebSplitter [?], which can divide web content into a number of elemental content items for each device. Adaptation depends on an associate policy file, and each device does not work cooperatively. Coles et al. proposed a framework for coordinated multi-modal browsing with multiple clients [?]. The proposed framework enabled the cooperative operation of multiple devices. Each device shared a browsing session by using consolidated proxy servers. This technology focused on session management between devices, and did not focus on the distribution of digital content from device to device. Karadkar et al. [?] proposed a content adaptation technology based on device specifications. This technology adapted digital content for a browsing device when the user accessed the content through a particular device. We have also developed temporal synchronous content for multiple devices based on extended SMIL [?]. These technologies are based on the expectation that users will browse content and play it using multiple devices indoors. The related mechanisms do not allow convenient outdoor use.

7. CONCLUSION

We described a mechanism for working with embedded web content in real-world situations. A network management device was used to control various devices indoors and out so that embedded web content bound to physical objects could be acquired. Users could cooperatively acquire desired digital content by providing their own context information such as their properties, device specifications, and configurations.

We also explained the *Functional web* concept, which enabled the addition of functions for content operations. A *Field* was used to add suitable functions and send particular messages regarding *Functional web content*. As a result, users could use web content directly and intuitively.

We also defined a description language for the *Functional web*, which could describe the specifications for *Field* and functions attached to conventional web content.

To confirm the effectiveness of our proposed methods, we developed a practical prototype called the Virtual Insect Catching System. It was designed to be operated by children, and we tested it during an NICT open house with 48 participants. The prototype system worked well, and the children could acquire embedded web content and search for related web content from the Internet. In addition, they could create new web content by merging the content items they had acquired. None of the system operations required the use of a keyboard, except for the inputting of comments

on WebBoard. During the test, the children only used touch-screen panels.

These methods can be applied to various content services that we expect to become more widely used as ubiquitous networks and devices provide constant access. Our *Functional web* mechanism was designed for such a ubiquitous information environment.

Unfortunately, current description languages do not support composite configuration of *Fields*. The extension of description languages and their application to other areas of content service will be the focus of future work.

8. ACKNOWLEDGMENTS

This paper was supported in part by the Scientific Research Fund Foundation (A)(2) for the year 2004 under the project title "Multi-modal Searches, Views on Mobile Content, and the Generation of Broadcasting Content" (Project No: 14208036, Representative: Katsumi Tanaka). We would like to thank Dr. Koji Zettsu of NICT for his useful discussions. We also wish to thank Dr. Hiroshi Narazaki, Mr. Takanari Shirasaka, Mr. Takashi Fukushima of KOBE Steel, Ltd. and Mr. Takahiro Matsumoto of Visionary Engineering for supporting prototype development.

9. REFERENCES

- [1] M. Murase, T. Iwamoto, T. Nagata, N. Nishio, and H. Tokuda. Implementation and evaluation of wapplet framework. In *Proceedings of IEEE International Workshop on Networked Appliances (IWINA)*, pages 275–284, Jan 2002.
- [2] M. Minami, H. Morikawa, and T. Aoyama. Ad-hoc service composition framework for networked functions stone: Service synthesizer on the net. In *Proceedings of JPSJ DICOMO*, pages 13–18, Jun. 2000.
- [3] Y. Kidawara, K. Zettsu, T. Uchiyama, and K. Tanaka. Device cooperative web browsing and retrieving mechanism on ubiquitous networks. In *Database and Expert Systems Applications (DEXA2004)*, Springer LNCS3180, pages 874–883. IEEE, Sep 2004.
- [4] B. Ullmer, H. Ishii, and D. Glas. Mediablocks: Physical containers, transports, and controls for online media. In *Proceedings of SIGGRAPH'98*, pages 379–386, July 1998.
- [5] B. Ullmer and H. Ishii. Mediablocks: Tangible interfaces for online media. In *Extended Abstract of CHI'99*, pages 31–32, May 1999.
- [6] R. Want, T. Pering, G. Danneels, M. Kumar, M. Sundar, and J. Light. The personal server: Changing the way we think about ubiquitous computing. In *Proceedings of UBIComp2002*, pages 194–209, 2002.
- [7] K. Gronbak, J. F. Kristensen, P. Orbak, and M. A. Eriksen. Physical hypermedia: Organizing collections of mixed physical and digital material. In *Proceedings of the 14th ACM Conf. on Hypertext and Hypermedia (HT '03)*, pages 10–19. ACM, August 2003.
- [8] L. Romero and N. Correia. Hyperreal: A hypermedia model for mixed reality. In *Proceedings of the 14th ACM Conf. on Hypertext and Hypermedia (HT '03)*, pages 2–9. ACM, August 2003.
- [9] M. J. Weal, D. T. Michaelides, M. K. Thompson, and D. C. DeRoure. The ambient wood journals -

- replaying the experience. In *Proceedings of the 14th ACM Conf. on Hypertext and Hypermedia (HT '03)*, pages 20–27. ACM, August 2003.
- [10] J. Rekimoto, Y. Ayatsuka, and H. O. M.Kohno. Proximal interactions: A direct manipulation technique for wireless networking. In *Proceedings of INTERACT2003*, Sep.-Oct 2002.
- [11] N. Kohtake, J. Rekimoto, and Y. Anzai. Infopoint: A device that provides a uniform user interface to allow appliances to work together over a network. *Personal and Ubiquitous Computing*, 5(4):264–274, 2003.
- [12] J. Rekimoto, Y. Ayatsuka, and M. Kohno. Synctap: An interaction technique for mobile networking. In *Proceedings of MOBILE HCI 2003*, 2003.
- [13] J. Rekimoto. Smartskin: An infrastructure for freehand manipulation on interactive surfaces. In *Proceedings of CHI2002*, 2002.
- [14] A. Coles, E. Deliot, T. Melamed, and K. Lansard. A framework for coordinated multi-modal browsing with multiple clients. In *Proceedings of the 12th Int'l World Wide Web Conf. (WWW 2003)*, pages 718–726. ACM, May 2003.
- [15] R. Han, V. Perret, and M. Naghshineh. Websplitter: A unified xml framework for multi-device collaborative web browsing. In *Proceedings of MOBILE HCI 2003*, pages 221–230. ACM, Dec 2000.
- [16] U. P. Karadkar, R. Furuta, S. Ustun, Y. J. Park, J. C. Na, V. Gupta, and Y. P. T. Ciftci. Display-agnostic hypermedia. In *Proceedings of the 15th ACM Conf. on Hypertext and Hypermedia (HT '04)*, pages 58–67. ACM, August 2004.
- [17] Y. Kidawara, K. Zettsu, and M. Katsumoto. A distribution mechanism for an active user profile in a ubiquitous network environment. In *Proceedings of PACRIM*. IEEE, Aug 2003.