Design of Enterprise Web Servers in Support of Instant Information Retrievals*

Robin Qiu
Dept of Information Science
Pennsylvania State University
Malvern, PA 19355
robinqiu@psu.edu

Zhijun Zhang
College of Information Science and Technology
Drexel University
Philadephia, PA 19104
zhijun_w_zhang@drexel.edu

Abstract - *As the world is building "the Internet of things", the successful deployment of instant information retrievals web servers will be essential for enterprises to provide satisfactions for customers. For instance, the precise and pertinent information of a product should be delivered promptly and securely. By taking advantage of radio frequency identification technology, information retrievals for a product labeled with a smart tag can be performed much more efficiently than ever before. This article presents a web server design scheme to facilitate the information retrieving in a future product identification and tracking system, aiming at meeting the following competitive requirements: responsive, evolution-prone, and locally contextual. Bucket are used for identifying product states, while binary tree search algorithms are used to sort, store, and retrieve the precise and pertinent service for a request that carries a product identity.*

Keywords: Identification, information system, global identification code, instant information retrievals.

1 Introduction

Currently, no matter where a user stays on a supply chain, when a product from a manufacturer arrives, if he/she wants to retrieve some useful data for the product and accordingly conducts relevant business operations, he/she might have to search and browse through the web site of the manufacturer or use some identification number (input manually or automatically) to locate its information through a specific software application that directs his/her request to a designated and known enterprise service server. The process of getting and processing product information using the existing mechanisms is time-consuming and not convenient. In the very near future, when all the information silos in business are integrated and the world becomes the e-world where any single product in the world can be identified and tracked through the networks, the current methods will fail to deliver satisfactory services to customers [10].

To deliver a satisfactory service to meet the future needs, a well-defined service model should be architected and supporting technologies be investigated, determined, and enabled. A well-defined service model should include at least the following: 1) a well-defined identification scheme, 2) a least-effort identification mechanism, and 3) a systematic and responsive content delivery scheme.

There are various identification schemes, such as the Universal Product Code (UPC) or so-called "bar code", Vehicle Identification Numbers, International Standard Book Numbers, and Electronic Product Code (EPC). One of the most successful identification schemes is the UPC. A UPC code is created using a 12-digit numeric sequence scheme. A UPC code essentially provides two separate numbers, one for a manufacturer identifier and another an object model number. A series of light and dark lines are used to represent a UPC code, while a human readable numeric equivalent is given together with the standard symbol. A UPC code is typically used for the identification of a type of objects rather than a single object item. Note that many variant barcodes are adapted for different applications [9, 10].

Although the radio frequency identification (RFID) technology has been for many years, the use of RFID tags has been limited due to the high cost of ownerships. With the advances of semiconductor technologies, RFID as a promising tagging technology has advanced substantially recently [9]. An RFID transponder or tag (it is also called a smart label) can hold much richer information than ever before. More importantly an RFID tag can be read-write. The data capability in light of volume is unceasingly increasing, while the cost of an RFID tag continues to drop. The prices of an RFID tag will drop into the range of a few cents in a couple of years. Paper thin RFID tags and the use of small and cheap RFID readers with the capability for wider reading ranges and faster data processing will become reality [2, 9]. It is the data capacity of an RFID tag that creates an opportunity of labeling any single object item in the physical world, which makes possible a tagged product uniquely identifiable and trackable in the whole world. In conjunction with the RFID technology EPC is designed and developed for uniquely tagging a physical object, aiming at promoting the efficiency and effectiveness of identification and tracking systems [3].

The EPC scheme as an emerging technique is created using a 96-bit numbering scheme for RFID applications. By taking advantage of the increased data capacity of an RFID tag, an EPC code utilizes essentially three separate numbers, one for a manufacturer identifier, another a product type number of the manufacturer, and the other the object serial number of the given product type made by the manufacturer. EPC makes possible the unique identification of any single object in the world [3].

The advances of networking and Internet technologies have brought the world into an information era [7, 9]. Rich information is available on the Internet. Product information can be accessible anywhere and anytime using a software application (e.g., a data warehousing or web application) provided the location of the object information service server and access privileges are given. However, the current implementation of mapping the object's identification into the service server address is difficult. It is usually done using proprietary mechanisms in an application (e.g., directory service, database searching, object naming service) [9], resulting in tremendous efforts in software development, deployment, training, and maintenance. Such an application typically is time-consuming in use. Nevertheless, the corresponding application is of poor interoperability, scalability, and capability of integration [7, 11].

Therefore, there is a need for a method to promptly access and retrieve the pertinent information of a globally and uniquely identifiable product through commonly and publicly used hardware and software such as the Internet [2, 7, 11]. In Section 2, we review an innovative identification scheme, which provides a direct link from an identifiable product to its service provider. In Section 3, the GIC-based generic service model is briefly reviewed. In Section 4, an innovative web server design and implementation in support of instant information retrievals is discussed. In Section 5, at last, we conclude this article by highlighting the future research.

2 An innovative identification scheme

By taking advantage of the well-deployed and adopted Internet, GL AgilityTech, Inc. proposes an innovative identification scheme called global identification codes (GIC) for identifying and tracking objects (products or services) worldwide, no matter whether these objects are physical or logic [7]. By embedding the location of a designated service server into an identification code, the GIC scheme essentially eliminates proprietary mechanisms for mapping an object to its service server [10, 11].

A GIC is a code of a finite alphabetic number. It mainly consists of two essential parts: one is the unique identification of a tagged product/service; another is the Internet protocol (IP) address of a designated service provider for the product/service. Symbolically, a GIC code is defined as "Header + Object (Global) Identifier + IP Address + Optional Parameters". The actual allocation of bits and the length of each part may be determined by the header information. The proposed GIC scheme realizes two integrated fundamental functions. On one hand, an object assigned with a GIC code can be uniquely identifiable in the whole world. On the other hand, through a networked computer system (an intranet or the Internet, wired or wireless), the information on the GIC tagged product/service can be straightforwardly accessed and retrieved from a designated service provider, which is specified by the IP address in the GIC code.

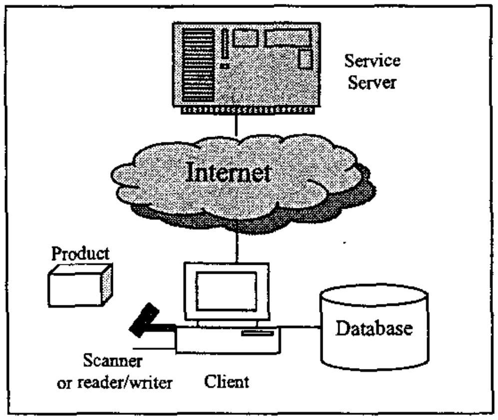
For applications using the GIC scheme, an object identifier can be UPC, EPC, or other specific identification code. For simplicity, examples using EPC as the product/service identifier in a GIC code are presented in the article. An EPC code is created using a 96-bit numbering scheme [3]. It consists of an 8-bit header and three data partitions, i.e.,

02.0006A66.56271F.0003476AB
8 bits.28 bits.24 bits.36 bits
Header.Enterprise.Product.Serial Number

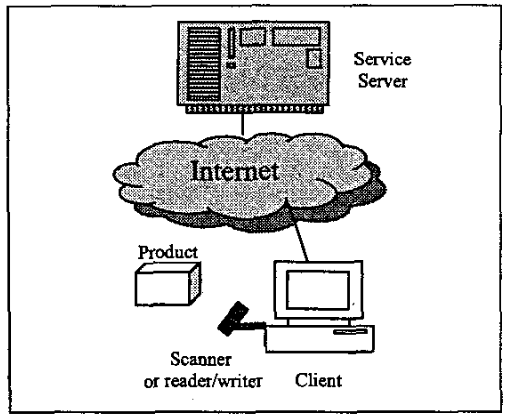
The provided information will allow unique identification for more than $1.1\text{x}10^{18}$ items manufactured by each enterprise [3]. Accordingly, a GIC is then symbolically defined as

02.IP.0006A66.56271F.0003476AB.XXx
8 bits.IP.28 bits.24 bits.36 bits.finite bits
Header.IP Address.Enterprise.Product.Serial Number.Other Parameters

One generic implementation of the GIC scheme is illustrated through a service model shown in Fig. 1 (b) [7]. An object is tagged with an RFID transponder. The transponder can be read-only (write once) or readable/rewriteable. An RFID reader can be located where it is needed, for instance, a plug-and-play device in a computing system. An application (e.g., Internet browser) with a GIC software module parses the data (i.e.,



(a) Current approach



(a) GIC approach

Figure 1. Product/Service Tracking and Identification Information Models

a GIC code held in the transponder) read from the transponder, and requests the information (e.g., package, product, or specification) from a given service provider if an IP is available. The service provider returns the requested information. The returned information can be either simply displayed or processed to meet the needs of the requesting application. Apparently, no burdens are added on the user side, i.e. no any proprietary mechanism (e.g., database, directory service, or object naming service) as shown in Fig. 1(a) is required.

3 GIC-based service model

We have developed an instant information retrievals (IIR) based conceptual prototype using the proposed GIC scheme and enabled RFID technology by taking into account the system interoperability and ease of integration. The prototype takes advantage of SUN One technology, where an JSP based web form is used to deliver the pertinent content based on the request triggered by a user's click, which reads a GIC from a GIC-tagged product [10, 11].

Web services can be used to implement most common business domain functions. Instead of deploying dedicated servers for IIR systems, web services should be deployed as enterprise common computing resources; in other words, all the business logics implemented in the web services server can be shared by any other enterprise information system. The delivery of the pertinent information of a product can thus be performed through a normal web server for an enterprise. Fig. 2 illustrates such a generic service model [7], where a web server delivering information can be implemented using all the enabled web technologies. The web server retrieves the pertinent information from the designated web services server whenever a request is solicited from a browser (i.e. a user).

The proposed service model simply uses commonly deployed computer hardware and software rather than solution-specific, industry-dependent, or proprietary software applications, while the service provider can reside anywhere in the networked world. To make this revolutionary approach to promptly accessing and retrieving pertinent information happen and be widespread employed, it will be necessary for a well-formed standards defined across nations and accepted and adopted worldwide. A standardized rather than proprietary parser (i.e., a RFID reader driver for commonly used computing devices) can be developed. As the RFID technology continuously advance, an RFID reader can be seamlessly integrated with a wireless mouse, cellular phone, or personal data assistant while keeping its cost affordable for consumers, i.e., minimum or even no increase in cost for computing accessories and mobile computing devices, which leads to the creation and deployment of a user least-effort approach to information retrievals [7].

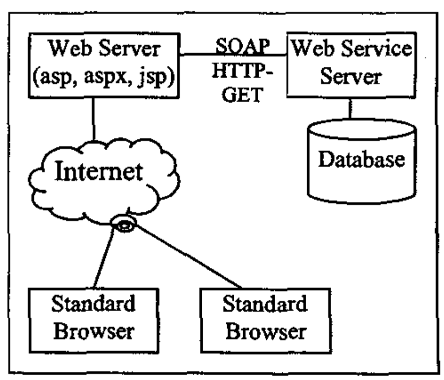


Figure 2. GIC-based Generic Service Model

4 Web server design scheme

Context-sensitive information delivery relates to website personalization. Currently there are two typical solutions for web content personalization. One solution is to use client-side cookie to remember information about the user, therefore requires nothing to be done on the server side. However, this is not a reliable solution because users may change devices and multiple people can share the same device. The other solution is to require users to identify themselves using user ID and password and then do a database lookup to find the user's information. The most popular websites today probably would not have more than 50 million registered users, therefore making it acceptable to use a database lookup to find personalization information. With GIC, we want to be able to handle billions of instances of the same product and thus need a better approach.

There has been work on adapting content delivery based on current server load [1], degrading the service provided to lower-priority request when the Web sever becomes loaded. Much of the work for accelerating content delivery has been on caching content so that they become immediately available when requested [5]. Research on content-based delivery strategy include WebGraph [6], which identified the static and dynamic components on each Web page and only re-generate the dynamic part when the page is requested.

To have a web server meet the requirements on responsiveness and context-awareness, the web server has to be constructed based on a well-defined and designed serving model. A context space model is proposed in [8], where products' services are categorized using states defined on a supply chain. When the state information of a product is piggybacked, the precise and pertinent service can be delivered in a very responsive manner. The following discussion focuses on how a web server can be constructed for providing a responsive and context-sensitive service for a tagged product.

4.1 GIC and quick access to product information

With GIC and standard RFID reading devices, a user could quickly retrieve information related to the RFID-tagged product. RFID reading devices can be built to work with information access devices such as computers, personal digital assistants, or smart phones. Once the GIC is read, the IP address in the GIC identifies the service provider of the information. The "enterprise," "product," "serial number," and "other parameters" can be passed as part of the HTTP request to the service provider. Based on certain history information the service provider has about this particular product, an appropriate Web page can be delivered to the user.

For example, if the GIC is:

02.8A.07.20.11.0006A66.56271F.0003476AB.5A

where "8A.07.20.11" is the IP address in hexadecimal format, 0006A66 is the enterprise ID, 56271F is the product ID, 0003476AB is the product serial number, and 5A is other parameter, then the HTTP GET request could be:

http://138.7.32.17/gic.jsp?enterprise="0006A66"&product="56271F"&serial="0003476AB"&other="5A"

In the above example, we assume that the page is a Java Server Page (JSP) and the name "gic" is the agreed-upon name by all such service providers.

4.2 Design considerations for the web server

The goal for GIC-based information retrieval systems is whenever a request arrives, the service provider would send back information most relevant to the particular product. For example, if the product has not been registered after purchase, the registration page would be sent back. If the product has an open request, then the delivered page would include the status and perhaps the answer to the request. We refer to this kind of information delivery as context-sensitive information delivery.

For the purpose of context-sensitive information delivery,

- Each product can be thought of as being in one of a limited number of states.
- A graph can be constructed to represent all the states and the transition among the states. We call this the state graph.
- Based on which state the current product is in, it can be decided what kind of information should be delivered to the user.
- The state graph may change over time. However, an old state can only be retired when no products are currently in that state.

Therefore, the key for designing such a Web server is to find out which state the product is in based on the information in the HTTP request.

There are two situations in which such mapping becomes trivial:

1. The state information can be stored in the GIC and thus is passed to the service provider as part of the HTTP request. For example, if the RFID is re-writable and each time the state changes, the state information is updated on the RFID, possibly in the "other information" field.
2. The number product is moderate enough that it is easy to use a table in memory to store the mapping from product serial number to the

current state. When a HTTP request comes in, the Web server could do a quick look-up of this table to determine which state the product is in.

However, if none of the above two situations is true, then a mechanism needs to be designed for the Web server to provide context-sensitive information as quickly as possible. The next section will discuss a proposed solution for this situation.

4.3 A product-state-aware solution

In this solution, we intend to map a product serial number to a product state. The serial number comes from a large number space (e.g. a serial number could be 36-bit, thus can have about $6.5x10^9$). The number of product states is rather small, typically ranging from less than ten to dozens.

We consider each product state as a bucket containing the serial numbers for the products that are currently in that state. For each bucket, we use a balanced binary search tree to store the serial numbers that are in this bucket. Unlike ordinary balanced binary search trees such as the AVL tree [4], ours will have the following unique features:

- Instead of having one key value stored in each tree node, we are going to have at least three key values stored in each node. These key values are sorted. The first value is the smallest key value among all the keys stored in the subtree whose root is the current node. The last value in the node is the largest key value stored in this subtree.

- If we refer to the sorted list of key values stored in a tree node as key_1, key_2, ..., key_{n-1}, and key_n, then key values between key_1 and key_2 are stored in the left child of the current node. Key values between key_{n-1} and key_n are stored in the right child of the current node.

- The key values are not stored as is. Instead, they are stored as a base and an increment. The first key value is the base for not only all the other key values in the same tree node, but also all the key values in the whole subtree.

- The size of a tree node will be enough to store at least three key values. However, because the key values are stored as base and increments, it is very likely that a tree node has enough space to store more than three key values. To avoid over-complicating the algorithms, we will use at least a byte to store any value. For example, if the key value can be up to five bytes, then we need 15 bytes in each tree node to store at least three key values. But when the base and the increment are both small enough to be stored in a byte, than the tree node would be able to hold 15 key values.

In C, the tree node can be declared like the following:

```
struct tree_node
{
    char keys[15]; //15 bytes to hold 3+ key values
    char base_size; //number of bytes for the base
    char incr_size; // number of bytes for the increment
    struct tree_node *left; // pointer to left child
    struct tree_node *right; // pointer to right child
}
```

With this data structure, we will have a balanced binary search tree for each product mode. When a request arrives with a product serial number, that number is used as the key to search the binary search trees. If the key is not found in any of the buckets, then the product will enter the first state, typical meaning a registration page will be returned.

For each bucket, a.k.a. balanced binary search tree, the key will be compared against the first and last key values in the root node to see if the key falls between them. If not, then they key is not in this bucket. Otherwise, we are going to compare the key with all the keys in the root node. If there is a match, then the product is currently in the state that corresponds to this bucket. If not, then we need to examine which two key values the current key falls between. If it falls between $key_1$and key_2, then we go to the left child and recursively search for the key. If the key falls between key_{n-1} and key_n, we go to the right child and recursively search for the key. Otherwise, the key does not exist in the current bucket.

To reduce the relative overhead caused by the pointers and etc. in the data structure, it is not difficult to increase the number of bytes used to store key values. Doing so would not increase the complexity of the algorithm.

Of course, it is possible to have too many products outstanding that there is not enough memory on the server to store all the information required. Then we could run multiple servers and distribute the buckets among the multiple servers. Give the memory capacity of current Web servers, and the world population, it is hard to image to have one bucket that requires more memory than a Web server can handle. If that situation does occur, then a solution would be to split the binary search tree and put each subtree on a separate serve. Then we can use a load balancer to direct request to the appropriate server based on the product serial number that comes with the request.

5 Conclusions

In this paper, we proposed a web server design scheme for facilitating information retrievals in a GIC-based product identification and tracking system. Bucket were used for identifying product states, while binary tree search algorithms were used to sort, store, and retrieve the precise and pertinent service for a request that carries a product identity and its state. The proposed design needs validation and further research for a large-scale application. In addition, an effective mechanism to locate buckets should be investigated, as these buckets are most likely distributed among many web servers.

Acknowledgement

The research was fully supported by GL AgilityTech, Inc. All rights and titles on all potential patentable inventions and relevant intellectual property solely belong to GL AgilityTech, Inc.

References

[1] T. Abdelzaher and N. Bhatti. "Web Server QoS Management by Adaptive Content Delivery," *Proceedings of the International Workshop on Quality of Service*, London, England, June 1999.

[2] K. Bonsor, "How Smart Labels Will Work," *http://www.howstuffworks.com/smart-label.htm*, Feb 2003.

[3] A. Brock, "The Electronic Product Code (EPC): A Naming Scheme for Physical Objects, " *White Paper of MIT Auto-ID Center*, 2001.

[4] R. Kruse, et al., *Data Structures & Program Design in C*, Second Edition, Prentice Hall, 1997.

[5] E. Levy-Abegnoli, A. Iyengar, J. Song, and D. Dias, "Design and Performance of a Web Server Accelerator," *Proceedings - IEEE INFOCOM*, Vol. 1, pp. 135-143, 1999.

[6] P. Mohapatra and H. Chen, "WebGraph: A Framework for Managing and Improving Performance of Dynamic Web Content," *IEEE Journal on Selected Areas in Communications*, Vol. 20, No. 7, pp. 1414-1425, Sept. 2002.

[7] R. Qiu, "Global Identification Code Schema: An Approach to Instant/Automatic Information Retrievals," *White Paper of GL AgilityTech, Inc.*, January 2003.

[8] R. Qiu and S. Ye, "Design and Development of Instant/Automatic Information Retrievals Systems," *2003 IEEE International Conference on Systems, Man and Cybernetics*, Washington D.C. Oct. 5 – 8, 2003.

[9] S. Sarma, D. Brock, and K. Ashton, "The Networked Physical World: Proposals for Engineering the Next Generation of Computing, Commerce & Automatic-Identification," *White Paper of MIT Auto-ID Center*, 2000.

[10] S. Ye and R. Qiu, "Global Identification Code Scheme for Promptly Retrieving the Pertinent Information of a Worldwide Uniquely Identifiable Object," *Proceedings of the 4th International Conference on Control and Automation*, Montreal, Canada, pp. 1000-1004, June 9-12, 2003.

[11] S. Ye and R. Qiu, "Web Services Oriented Approach to High Availability of Product Information," *2003 International Conference of Industrial Engineering & Engineering Management*, Shanghai, China, Dec. 6-8, 2003.