GLOBALLY UNIQUE PRODUCT IDENTIFIERS – REQUIREMENTS AND SOLUTIONS TO PRODUCT LIFECYCLE MANAGEMENT

Kary Främling*, Mark Harrison, James Brusey****

**) Helsinki University of Technology, P.O.Box 5500, FI-02015 TKK, Finland*
***) Institute for Manufacturing, Cambridge University, Mill Lane, Cambridge CB2 1RX, United Kingdom*

Abstract: Managing product information for product items during their whole lifetime is challenging, especially during their usage and end-of-life phases. A major challenge is how to keep a link between the product item and its associated information, which may be stored in backend systems of different organisations. In this paper, we analyse and compare three approaches for addressing this task, i.e. the EPC Network, DIALOG and WWAI. *Copyright © 2006 IFAC*

Keywords: Coding schemes, communication protocols, decentralised systems.

1. INTRODUCTION

Product lifecycle management (PLM) can be interpreted in at least two ways. A "traditional" interpretation of PLM is the management of design and manufacturing data of different product types, including their versions and variants. With this *product-type PLM* interpretation, the product lifecycle consist of phases such as initial design, manufacturing, marketing, or creating new versions. In this case, PLM signifies managing product information related to the corresponding product type and the lifecycle of the product type.

With the increasing degree of customisation of manufactured products and the need to operate and recycle the products efficiently, the concept of PLM needs to be expanded from product-type PLM to *product-item PLM*. The lifecycle of product items can for instance be divided into three different phases: beginning-of-life (BOL), middle-of-life (MOL) and end-of-life (EOL). The BOL phase includes design and manufacturing, MOL is the usage phase of the product and EOL is the disposal of the product. The product-type interpretation of PLM mainly corresponds to the BOL part and some parts of the MOL phase.

When moving from the first interpretation of PLM to the second, the way that product information is stored and accessed changes radically. With product-type PLM, product information is typically handled on a company or organisational level because they produce most of the product information. In product-item PLM, a large amount of the product information is produced during the usage phase of the product, outside the organisations that designed or manufactured them. Therefore the collection and usage of product information becomes more challenging in product-item PLM.

So-called product- or item-centric approaches to product information management offer a solution to product-item PLM (Kärkkäinen, *et al.*, 2002, 2003; Bajic and Chaxel, 2002; Chaxel, *et al.*, 1999; Parlikad, *et al.*, 2003). The biggest challenges in product-item PLM are the following:

1. It is usually impossible to store all product information with the product item itself, so parts of it need to be stored in "backend" systems.
2. In order to associate product items with the correct product information in backend systems, every product item needs to be uniquely and globally identified among all other product items.
3. Product items usually change their location during their lifecycle, so they tend to have only intermittent network access (typically through Internet). When they have network access, they may need to access, modify or synchronize product information with the backend systems.

The second and third challenges can be addressed in many different ways. In this paper, we analyse the

pros and cons of the three approaches identified by the authors that address these challenges:

1. EPC Network approach, where a number of existing product identifiers (such as the GTIN) can be embedded in a product-item identifier (the electronic product code, EPC) and also defines related information systems that associate the EPC with product information in backend systems.
2. The ID@URI approach, which uses existing product identifiers (item-level or not) and explicitly expresses where product information can be accessed in backend systems.
3. World Wide Article Information (WWAI) approach. WWAI uses existing product-item identifiers and links to product information in backend systems through a peer-to-peer (P2P) based lookup mechanism.

The structure of this paper is as follows. In section 2 we analyse requirements and tradeoffs of product identifiers, in particular from a product-item PLM point of view. Section 3 attempts to compare how well the three known approaches respond to these requirements in different situations. The final section presents our conclusions.

2. REQUIREMENTS AND TRADEOFFS OF GLOBALLY UNIQUE PRODUCT IDENTIFIERS

There are a variety of criteria by which we may qualitatively assess the value of a naming scheme and how appropriate it is for product-item identification. We believe that it is desirable for the naming scheme to be:

- Simple,
- Open,
- Long-lived,
- Standard,
- Extensible,
- Hierarchical,
- Providing some guarantee of uniqueness,
- Distributed,
- Allowing private numbering,
- Providing cost effective registration, and,
- Cost effective per item.

It is desirable for a naming scheme to be *simple* since complex ones will tend to be costly and difficult to implement. This will minimise the barriers for entry for software developers and systems integrators, as will the characteristic of being *open*. An open naming scheme is desirable since ones that are proprietary or encumbered by restrictive patent licenses are less likely to be widely adopted. Also, open schemes will not restrict users to one particular software package or hardware platform but allow multiple and varied implementations.

Globally Unique Product Identifiers (GUPIs) generated by the naming scheme should be *long-lived* and must last at least as long as the product that they are associated with and possibly longer. Therefore, the scheme should not encode transitory attributes into the identifier. It might be possible in some cases to access the product, and to update the GUPI, but in general we cannot assume this to be the case.

The naming scheme should be a *standard* one. A global naming scheme needs common acceptance. It is not enough for the scheme to be well specified and open; it must also be adopted widely. Of course, we must accept that to build consensus and to achieve standardisation requires a workforce and financial support. On the other hand, we should avoid creating new standards if old ones are sufficient. If new ones are required, some support for interchange of data with systems that use legacy standards is desirable.

Much effort will be applied to adopt any particular scheme, but even more effort will be required in the future if it becomes necessary to convert to some new scheme. Therefore it is important that the naming scheme is *extensible* and allows the set of possible unique names to grow.

A product-naming scheme that is *hierarchical* might allow the product type to be derived from its name directly, thus potentially simplifying some operations. Furthermore, a hierarchical structure may reduce the amount of duplication in the storage of information that is the same for a particular product type.

It is important for the naming scheme to provide some *guarantee* of the global uniqueness of the identifier. If the identifier were not unique, some other contextual information would be needed to fully identify the product. Although in some cases, context can be obtained, say through the position of the product, or from the order in which events are seen, it may not always be possible.

Although the easiest way to obtain unique identifiers might seem to be to centralise their naming and name resolution to network addresses, this would also be cumbersome. Rather, the scheme should *distribute* the resolution of network addresses in such a way that the failure of a node in the network should not disable name resolution nor product information lookup from other nodes. At the same time, it might sometimes be necessary to have *private* identifiers that are only intended for internal use. Preferably any private identifiers should also be identifiable as such.

The final two requirements have implications for cost effectiveness. First, any *registration* with a central body will add to the cost of using the scheme. In some cases, this cost may be small if the registration needs to be performed only once for a large range of identifiers. Second, the cost of identifying the *item*, whether it is via passive or active Radio Frequency Identification (RFID) tag, or simply the addition of a barcode, also increases the cost of using the scheme. In considering this cost, we must also consider how compact the naming scheme is, since tags that require less memory tend to be cheaper.

3. ANALYSIS OF RELEVANT NUMBERING SYSTEMS

In this section, we first present three currently existing approaches for GUPIs and how they address the needs of PLM. In the last sub-section, we attempt to analyse how well these approaches satisfy the requirements set out in section 2.

3.1 EPCglobal approach

The Electronic Product Code (EPC) is one approach for creating references between product items and the product agent or backend information services. EPC identifiers are URNs that uniquely name objects. The EPC URN naming scheme is a ratified published open standard, known as the "EPC Tag Data Standard", which describes how a number of existing product identifiers may be formatted as a URN for use in the EPC Network. These existing product identifiers include serialized versions of the Global Trade Item Number (GTIN), which is related to the UPC-12 / EAN-13 barcodes already found on many products. When the EPC URN identifier is stored on passive RFID tags, a very compact binary format is used, requiring a minimum of only 64 or 96 bits of tag memory to store a wide range and large number of identifiers. This is achieved by not encoding the URN into binary as 8-bit bytes per character – but instead encoding various identifier fields as binary-encoded integers and replacing the URN prefix with a compact 8-bit "header" code.

The Object Name Service (ONS) is the lookup mechanism used to obtain one or more URLs where authoritative information can be obtained for a given EPC. ONS is simply an extended implementation of the Domain Name System (DNS), using NAPTR (Naming Authority Pointer) DNS records. ONS provides a scaleable hierarchical lookup system, re-using existing DNS tools and protocols to perform the lookup. The root-level of ONS has been operational for well over a year and resolves the Manager ID (usually points to the manufacturer of the product). The root-level is administered by EPCglobal Inc., and the operation of the root-level servers is currently subcontracted to Verisign Corporation. The second tier of ONS provides for resolution of different product classes within a company. It may be implemented using an in-house DNS name server. Entries in the root-level ONS lookup system are currently only provided for subscribers of EPCglobal Inc.

The ONS records provide not only a set of one or more URLs of information services, but also meta-data to indicate the type of information service provided by each URL in the set. This allows computer programs to automatically select between web pages, EPC information services, web services, XML data files and other services that may be added in the future, without needing to attempt to guess this from the URL pathname.

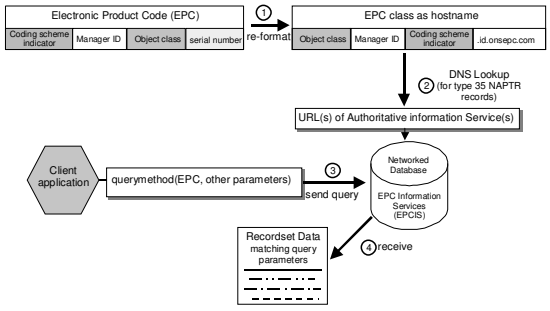

Fig. 1. Product information lookup with EPCglobal approach.

For Global Trade Item Numbers (GTIN), the Object Name Service provides records only at product class resolution – no serial-level resolution. The EPCglobal Architecture Framework Document (Traub *et al.*, 2005) identifies "Discovery Services" as a future component of the network that will provide for dynamic serial-level lookup across the entire supply chain, in a way that is both massively scaleable and secure. To date, EPCglobal have not yet chartered a work group to standardize Discovery Services.

In the EPC Network, product item information may be accessed from various networked databases using a standardized interface framework, EPC Information Services (EPCIS) as illustrated in Fig. 1. The EPCIS specification is currently at Last Call Working Draft stage and will standardize how client application programs may request current or historical data about EPC-tagged objects, together with higher-level semantic annotations such as the business steps and transactions associated with a particular observation event. EPCIS defines a modular framework for query and capture of such information, together with a standardized reporting format/schema and transport bindings to web services and existing electronic data interchange (EDI) technologies such as EDI INT AS2.

In addition to Tag Data Standards, ONS and EPC Information Services, the EPC Network intends to develop an end-to-end architecture of layered open standards, ranging from the air interface (reader-tag radio communication) all the way up to interfacing with existing business information systems. This also includes standardization of the Reader Protocol (software interface for reading/writing to tags), Reader Management (network monitoring of readers), Application Level Events (filtering, collection and reporting of observation events). EPC standards are developed through a community participation process involving end-users and technology providers. Following ratification, they are published by EPCglobal and freely available for download. To date, standards have been ratified for Tag Data

Standards, Object Name Service, Application Level Events. It is expected that Reader Protocol and EPC Information Services will be ratified by the first half of 2006.

3.2 ID@URI approach and DIALOG information system

In the DIALOG approach (Främling, 2001; Huvio, *et al.*, 2002) an ID@URI notation has been used for creating a GUPI, where the ID part identifies the product item at the URI (Uniform Resource Identifier (Berners-Lee, *et al.*, 1998)). If the URI is a Uniform Resource Locator (URL), it is straightforward to link to a product agent or backend information services so no ONS-type approach is needed. The uniqueness of a URL is guaranteed by the DNS (Domain Name System) infrastructure. For an ID@URI to be a GUPI, the ID part should be unique for that URI. At the minimal level the ID@URI reference can be embedded as a barcode or using a passive RFID tag. In that case the URI should preferably remain the same during the product's entire lifecycle because changing it requires physical access to the product item itself. For more intelligent devices, such as smart cards or car engine control units, this should not usually be an issue because they can update the URI themselves if needed. It is also possible to embed a list of alternative ID@URI references if uninterrupted access to the backend system is essential. Since the URI part uses existing standards and since there exists many possible standards for the ID part, this approach does not need any new identifier standards.

Product information can be accessed through a middleware system called DIALOG as illustrated in Fig. 2. The DIALOG system is mainly used for testing and verifying new concepts and models for research purposes. It has also been used in two industrial pilots in a multi-enterprise setting in 2002 and in 2004 for tracking shipments in project deliveries (Kärkkäinen, *et al.*, 2004). The current DIALOG implementation supports three protocols and data formats for message passing. The communication protocol to use can be specified as a part of the URI. Available protocols are:

SOAP (www.w3.org/TR/SOAP/). Programming language-independent protocol. Data is transferred as text using the XML notation.
HTML <FORM> (www.w3.org/TR/html401/). Programming language-independent protocol. Data is transferred as text using the HTML form format (can also be XML-encoded).
Java RMI (Sun Microsystems, 2002). Mainly used in development and in intra-company installations. RMI is flexible and easy to use, but firewalls and version management tend to be problematic.

Supporting different communication protocols is technically simple. An average of about twenty lines of code has been needed to implement a new messaging protocol, which represents less than 1% of the total middleware implementation. In addition to selecting standardised messaging protocols, a major challenge is to standardise the communication interfaces, specifically messages and their contents. The DIALOG software is distributed using an open source policy; the message interfaces are publicly available so that any software producer could implement them and have their applications communicate successfully.

In practice, an open source solution is not sufficient for creating a standardised communication interface. Instead the policy will be to select and support one or possibly several communication interfaces. Both EPCIS and WWAI could be good communication interface candidates, but they are not the only ones. The semantic web (Berners-Lee, *et al.*, 2001) community has also produced several standards for representing and communicating structured information that could be useful. The Product Lifecycle Support (PLCS) initiative (www.oasis-open.org/committees/plcs/) could also provide a good communication interface standard, as well as many other standardisation initiatives.

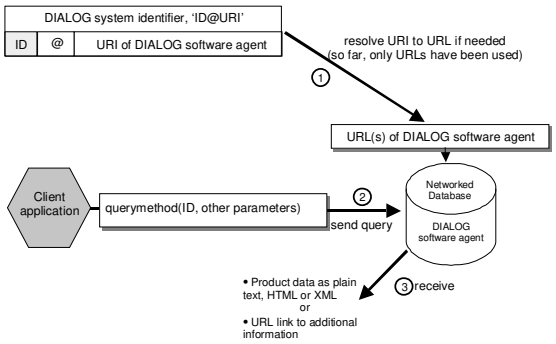

Fig. 2. Product information lookup with ID@URI, DIALOG implementation.

3.3 World Wide Article Information system

A different approach is offered by peer-to-peer (P2P) systems that are mainly known for file sharing of music and movies. However, P2P also has many desirable features for identifying nodes in the network as well as individual items. New nodes and items can be dynamically added at any time and are immediately integrated into the network. The network protocol usually takes care of assigning unique identifiers both for nodes and items automatically. Therefore there is no need for an external authority to manage codes as in the EPC approach. Other advantages of P2P solutions are that all nodes can maintain complete control of what data is distributed to whom (even though most file sharing applications do not check or restrict who gets access), good fault-tolerance (breakdown of one node affects the whole network very little) and possibilities to do load-balancing by using nodes that are "close" (in the network communication sense).

Table 1 Comparison of different GUPI approaches for the requirements presented in section 2

Requirement	EPC network	DIALOG	WWAI network
Simple?	some complexity in converting EPC to network address	network address directly accessible	requires P2P network lookup
Open?	EPCglobal ratified standards are feely available online and build upon existing open standards, e.g. XML, XSD schema, web services	makes use of existing open standards	identifier structure is open, but for the moment supported by only one company
Long-lived?	supports changing manufacturer's URLs without changing tag	URI on tag will need to change if manufacturer's address changes	WWAI address not tied to a specific URL
Standard?	ratified global standards established for use with retail goods	no standard of its own, but makes use of existing standard technologies	no standard of its own, but makes use of existing standard technologies
Extensible?	the header provides a mechanism for extending the EPC code	no limit to the number of bits in the ID. URI part could also use future network address resolving methods, e.g. ONS	unclear to what extent network address resolving can be modified if needed
Hierarchical?	includes manufacturer, and product type identifier parts	yes, if appropriate ID is selected (GTIN, EPC, other)	can include part type and item identification parts
Guarantee of global uniqueness?	centrally allocated Manager ID, item-level uniqueness decided and controlled by individual organisations	URI globally unique, item-level uniqueness decided and controlled by individual organisations	centrally allocated, item-level uniqueness decided and controlled by individual organisations
Distributed name resolution, product info. lookup?	only one root ONS exists for the moment, may increase in the future	as distributed as DNS, information lookup is vulnerable to node failures	P2P-type name resolution, failure of one node doesn't affect others
Supports "private" identifiers?	can have private identifiers (using a private ONS) but no way of identifying them as such	can have private identifiers	can have private identifiers
Registration cost?	as defined by EPCglobal cost of membership	DNS registration	as defined by Stockway or other certificate providers
Item identification cost?	compact representation supports cheaper RFID tags	long identifier, needs more expensive RFID tags	identifier length ≥ EPC and ≤ ID@URI

The World Wide Article Information (WWAI) protocol (www.wwai.org) developed by Stockway is based on P2P principles. Existing company codes as issued by EAN/UCC or other standardisation bodies identify nodes of the network. When a node has joined the network, it can autonomously issue identifiers for individual items (e.g. product items). New nodes are dynamically discovered when appropriate. The WWAI protocol defines messages that enable nodes to exchange any kind of information and link any kinds of objects to each other by named relations. From a P2P point of view, the main criticism against WWAI is that it requires certificates issued by a certification authority in order to become an information provider in open networks. It seems like this is motivated by the need to find a compromise between existing coding standards and ensuring the uniqueness of the codes, as well as ensuring data integrity. On the positive side, certificates automatically guarantee the authenticity of the information provider.

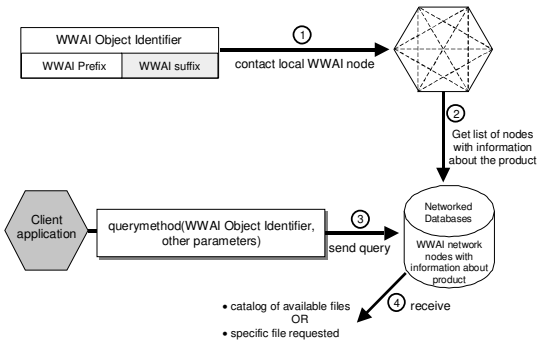

Fig. 3. Product information lookup with WWAI approach.

A WWAI node lookup (Fig. 3) is usually only needed when a product identifier for a given company is seen for the first time. After that, network addresses of known nodes are cached so that new node lookups do not need to be performed unless the cached address fails or changes for some reason. In order to get access to other nodes in the network that have

information about a specific product item (or some other item, e.g. a document, or a multimedia clip), it is sufficient to find one node with information about the product. This is because every time that a new node adds some information about a product item, this is automatically communicated to the other nodes that already have some information about that product item. Therefore there is no need for a separate "discovery" mechanism for accessing all information from all sources/organisations that have information about the product item.

3.4 Evaluation of different approaches

Table 1 gives a comparison on how well the different approaches correspond to the requirements set out in section 2. Although it might be possible to rate each approach, based on how it performs against each of the requirements, such ratings tend to be subjective. For this reason, no ratings have been provided.

4. CONCLUSIONS

Based on the comparison of Table 1 it is not possible to identify any "global winner" for PLM applications. The EPC Network has three key strengths with respect to PLM: First, it is an internationally accepted standard that is supported by a world-wide standards body (GS1). Second, the lookup mechanism helps to insulate the data on the tag from change. This is particularly important in a domain where the tag must last for the lifetime of the product and may only be accessible intermittently. Third, because it is becoming widespread, it may be the case that products have an EPC tag anyway, and that this tag can also be used for PLM. Certainly if some other approach was used, it may be necessary to think about how to avoid any confusion with existing EPC systems. Nevertheless, the EPC Network was not designed with PLM in mind, and some changes or extensions to the architecture may be required. Also, the registration cost may be too much of a barrier to entry for some users.

WWAI seems to be more technically sophisticated than the other approaches. The main challenge is that it has a very small industrial support compared to EPCglobal, so it may have difficulties to impose itself as a standard unless adopted by bigger players.

The DIALOG approach might be the most general-purpose one of the three because it places few restrictions on the format of the data on the tag. It is probably a good solution for "high-end" products with computing power and for smaller ad-hoc installations. Nevertheless, some steps may need to be taken to address the longevity of URLs used.

REFERENCES

Bajic, E. and F. Chaxel (2002). Auto-ID mobile information system for vehicle life cycle data management. In: *Proceedings of IEEE SMC 2002*. Institute of Electrical and Electronics Engineers Inc., (Yasmine Hammamet (Ed.)), Tunisia, pp. 387-392.

Berners-Lee, T., R. Fielding, U.C. Irvine and L. Masinter (1998). *Uniform Resource Identifiers (URI): Generic Syntax*. Available online (March 5th 2004): "http://www.ietf.org/rfc/rfc2396.txt"

Berners-Lee, T., J. Hendler and O. Lassila (2001). The Semantic Web. *Scientific American*, May 2001.

Chaxel, F., E. Bajic and J. Richard (1999). Automotive vehicle data management based on holon-product paradigm. In: *Proceedings of IEEE SMC '99*. Institute of Electrical and Electronics Engineers Inc., Tokyo, Japan, pp. 434-439.

Främling, K. (2002). Tracking of material flow by an Internet-based product data management system (in Finnish). *Tieke EDISTY magazine*, **No. 1, 2002**. Publication of Tieke, Finland.

Huvio, E., J. Grönvall and K. Främling (2002). Tracking and tracing parcels using a distributed computing approach. In: *Proceedings of the 14th Annual Conference for Nordic Researchers in Logistics (NOFOMA'2002)*, (Solem, O. (Ed.)), 29-43. Trondheim, Norway, 12-14 June 2002.

Kärkkäinen, M., K. Främling and T. Ala-Risku (2002). Integrating material and information flows using a distributed peer-to-peer information system. In: *Collaborative Systems for Production Management* (Jagdev H.S., Wortmann J.C., Pels H.J. (Ed.)), 305-319. Kluwer Academic Publishers, Boston, USA.

Kärkkäinen, M., T. Ala-Risku and K. Främling (2003). The product centric approach: a solution to supply network information management problems? *Computers in Industry*, **52:2**, 147-159.

Kärkkäinen, M., T. Ala-Risku and K. Främling (2004). Efficient Tracking for Short-Term Multi-Company Networks. *International Journal of Physical Distribution and Logistics Management*, **34:7**, 545-564.

Parlikad, A.K., D.C. McFarlane, E. Fleisch, E. and S. Gross (2003). *The Role of Product Identity in End-of-Life Decision Making*. Auto-ID Centre White Paper.

Sun Microsystems (2002). *RMI Specification*. Available online (December 13th, 2002): http://java.sun.com/products/jdk/1.2/docs/guide/rmi/spec/rmiTOC.doc.html

Traub, K., G. Allgair, H. Barthel, L. Burstein, J. Garret, B. Hogan, B. Rodrigues, S. Sarma, J. Schmidt, C. Schramek, R. Stewart, K.K. Suen (2005). *EPCglobal Architecture Framework*. Available on-line (November 2005): http://www.epcglobalinc.org/standards_technology/Final-epcglobal-arch-20050701.pdf