Integrating material and information flows using a distributed peer-to-peer information system

Mikko Kärkkäinen, Timo Ala-Risku and Kary Främling
Helsinki University of Technology, TAI Research Centre, PO Box 9555, FIN - 02015 TKK, FINLAND

Abstract

The management of item-level information is one of the biggest challenges currently facing supply chain management. Managing product-related information grows ever more difficult because increasing product customisation adds to the amount and variety of product-related information, and increasingly complex supply networks complicate the integration of product information databases. A potential solution for the problems of item-level information management is moving to product centric information management, namely centralising the information to individual products and accessing the relevant information directly through them. This paper presents an approach and a system for managing item-level information. The system is based on software agents, peer-to-peer information sharing, and a coding scheme that utilises the already allocated domain names of the Internet. Applications of the system to tracking and logistics control are reviewed.

Keywords

Information management, customisation, material flow, information flow, peer-to-peer information systems.

INTRODUCTION

The management of item level information is one of the biggest challenges currently facing supply chain management (Kärkkäinen & Holmström, 2002; Tuttle, 2002; Zipkin, 2001). This challenge is mainly caused by two different factors. Firstly, the amount of item-level information is increasing, and secondly, the information has to be managed in supply networks of increasing complexity.

The amount of item level information is increasing because there is demand for a higher number of product variants and customised products (Bowersox *et al.*, 2000). Also, more strict governmental requirements on product lifecycle management, traceability, and after-sales support are emerging. For example, in the European Union all consumer electronics have to be recycled, their minimum warranty time has been set to two years and some member states demand even much longer warranty times (Hamilton, 2001). Thus, companies are forced to retain increasing amounts of product related data and also amend it as the product advances in the supply chain or is inspected or repaired after purchase. There is also

an increasing customer pressure for traceability in many product groups from pharmaceuticals to grocery products (Töyrylä, 1999). All these requirements contribute to the amount and variety of product related information to be retained, which considerably complicates the task of item-level information management.

Supply networks are increasingly complex as multiple companies, even companies from different continents, participate in processing individualised products, and the participating companies in the supply networks do not remain the same over the product life-cycle (Hewitt, 2000). A growing tendency for outsourcing has led to a significant increase in the number of partners that participate in global supply networks. Furthermore, the companies that participate in the production of a product variant can change over time, depending, for example, on the fluctuations of currency rates or minor changes in the product design. (Andel, 2001). The increasing complexity and continuous change of supply networks presents a considerable challenge for the management of product-related information, as it is impossible to integrate the information systems of all the companies in the network.

The current practice for managing product-related information is that individual companies store information that they create in company databases or paper-based filing systems. In such cases, retrieving the correct information may, in the worst case, require hours of work and the intervention of several people with various kinds of competence. (Töyrylä, 1999). The challenge quickly grows even more severe, as the network of companies that produce components and related information grows in size and complexity. The increasing complexity of the networks leads to the dispersion of the information to increasingly many locations. The problem is intensified by the fact that companies rarely have common product codes for specific products or parts (Wilder, 1999). Also, the increasing quantity and variety of the information to be managed makes the definition of standard transactions for exchanging the information difficult. There is, thus, a need for the development of more flexible information sharing approaches.

One solution for the problem of information dispersion would be the creation of a central product information database for the whole supply network. However, there are some major obstacles to doing so. Firstly, companies in supply networks are independent entities, and it is, thus, difficult to convince them to hand over their information to be managed by an external party (Euwe and Wortmann, 1998). Secondly, building a central database for the supply networks is a very complicated task and integrating all the companies involved to the database consumes a lot of resources. This problem is multiplied by the flexible nature of the networks. Integration complexity is still increased by the difficulty of defining standard transactions due to the great variety of information to be exchanged. And thirdly, most companies are members in several supply networks, which makes central supply network databases undesirable for them. For example, it is not feasible for a components supplier to integrate with the product information databases of several Original Equipment Manufacturers (OEM's).

Another possibility for collecting and sharing product related information is interacting directly with the products themselves, i.e. moving to "product centric" information management (Auto-ID Centre, 2002; Ashton, 2000; Kärkkäinen et al., 2001). Research organisations as well as companies have developed and proposed

approaches and systems with which information can be directly linked to individual products.

This paper presents a new methodology, and a system utilising this approach for item-level information management. In the first part of the paper, existing models and systems for product centric information management are reviewed. The second section presents our proposed system, the Dialog system being developed at Helsinki University of Technology (http: //dialog.hut.fi/). Different application areas for the Dialog system are presented in the third part, and concluding remarks and areas for future research are reviewed in the final section.

EXISTING APPROACHES FOR PRODUCT CENTRIC INFORMATION MANAGEMENT

As presented earlier, traditional methods using company specific product information storage for managing product related information are becoming insufficient for the task. This is because of the growing amount and variety of information to be managed, and the increasing complexity and flexibility of supply networks that participate in generating that information.

A proposed solution to this problem has been linking the necessary information directly to the objects. We call this approach product centric information management. Many different approaches and systems have been developed for product centric information management. Development efforts are based on two fundamentally different approaches: developing proprietary solutions or developing standards for creating compatible solutions (Shapiro & Varian, 1999 pp. 227 - 296).

Two kinds of proprietary solutions have emerged so far, solutions developed by a single company to solve their problems, and proprietary solutions of service providers. A good example of a company's internal solution is the retention of maintenance data at a consumer electronics manufacturer. The company retains the maintenance related information of all its products in a centralised database, from which the information concerning a certain product can be captured by reading a bar code from the product. The main weaknesses of this kind of solutions are the necessity of transferring all needed information to a centralised database, which is both a time and resource consuming effort, and the fact that the system is operable only after an integration period. This kind of approach is undesirable for an independent maintenance service provider, because in order to operate with the products of several OEM's the company has to integrate with the systems of all the OEM's. Solutions analogous to this maintenance information management solution are the tracking solutions of express parcel carriers (Coia, 2001; Booker, 1999), and the tracking service Savi Technologies offers to its customers (Dierkx, 2000). All these systems use their own product coding, and are functional only with those codes and reading points that are connected to their systems.

Different service providers have also been emerging with the offer of linking products to information network addresses based on different coding schemes (Airclic, 2002; Wicol, 2002; Stebbins, 2000). Usually these services utilise existing code base, most often the EAN 13 or UPC code. The weakness of these kinds of system is that the codes used do not give the possibility to manage item-level

information. This is because the codes used only distinguish different types of products, not product individuals within a product type. The functionality of the Wicol system is illustrated in Figure 1. The other proprietary solutions are quite similar to the Wicol solution.

In these proprietary solutions, a company gets the information only through the server of a particular service provider. This is a major disadvantage to the users of the system because, in order for the system to work, all companies in the supply network should use the same service provider, which is infeasible for companies belonging to several supply networks. Thus, the solution is not scalable in an environment with complex, interlinked supply networks.
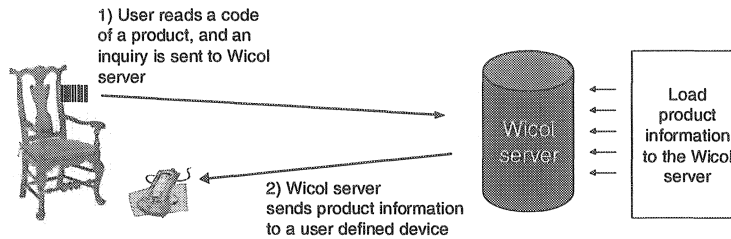

Figure 1 - The Wicol system (applied from Wicol, 2002)

The lack of universally readable item level coding and lack of scalability are the most severe problems with the proprietary solutions. Standardisation could solve these two problems. Auto-ID Centre, a research community based at Massachusetts Institute of Technology and Cambridge University facilitates a large initiative developing standards and technology to enable product centric information management (Ashton 2000; Auto-ID Centre 2002). It is developing a new standard code base for item level coding (the Electronic Product Code, ePC) and a method for linking product related information to the code and of transmitting it to the system that reads the code (the method is based on an Object Naming Server (ONS). The system developed at the Auto-ID Centre is illustrated in Figure 2.
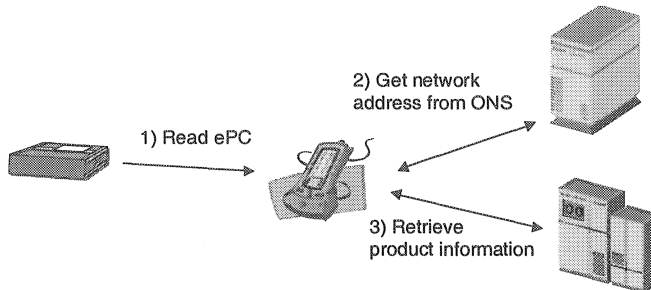

Figure 2 - The system developed at the Auto-ID Centre

The ePC is designed to be long enough to enable giving a unique identity to all products. Information related to individual products can then be stored in information networks and be accessed with the code. An information network address is attached to each code in an object naming server (ONS) located in a

predefined address on the Internet. (see Auto-ID Centre, 2002). If the standardisation of the product coding is successful, and the ONS servers can be constructed with the necessary functionality and scale, an open system will be available. Then, companies belonging to several supply networks can utilise the system with all their partners. Thus, the solution circumvents the biggest problems caused by proprietary solutions.

However, there are also some noteworthy weaknesses in the Auto-ID Centre solution. Firstly, standardisation of product coding is difficult, and it is difficult to achieve a global acceptance of the coding. For example, it took over ten years for the EAN/UPC coding to achieve a strong foothold in the consumer goods industry and grocery business. The proposed solution is supposed to have an even wider appeal. Auto-ID Centre has been able to gather a significant base of companies to its network, which we think will enable the creation of the standards, but the process will not be quick.

Secondly, visiting the ONS before accessing the information may increase traffic in the network, and delay access to the information. The reason for this is that the ONS first has to be contacted in order to retrieve the network address attached to the product code, before being able to look up and access the information network address.

We propose an approach conceptually similar to that of Auto-ID Centre, but which does not require establishing a new global standard. This approach is based on the use of existing standards for defining an alternative way of ensuring global uniqueness of the product code. The technical solution for accessing product information is based on intelligent agents that share information in a peer-to-peer fashion. Therefore, our solution could provide an immediately usable solution for practical applications, which would suffer less than the current Auto-ID approach from the problems associated with product standardisation and setting up object naming servers.

THE DIALOG SYSTEM

The Dialog system aims at solving the challenges of item level information management without the need of developing new standards for product coding. In this section, the Dialog system will be presented in the following order: First we will present the principles of product coding utilised in the Dialog system, then we will proceed to review the basic technology used in developing the software agents for the system, and finally we will describe the connection and information exchange between these agents.

Product coding in the Dialog system

The Dialog system does not require the development of new product codes, because it utilises already allocated, unique Internet domain names of companies to create globally unique codes for products and consignments. In our approach, the connection between a tangible object and the information network address that contains information regarding the object is defined by two pieces of information:

1. Identification part (string containing numbers and/or text of free choice)
2. URI (Uniform Resource Identifier), which is the Internet address of an agent associated with the tangible object

These two pieces of information guarantee that the resulting combination is globally unique as long as the identification part is unique at the given URI address. The organisation that owns the URI can arrange this by carefully allocating the identities inside that URI. The difference of our approach to Auto-ID Centre's is that our approach uses an existing method of providing objects with unique identity, namely the domain naming service. There is, therefore, no need to build a global base of codes for the products to be identified.
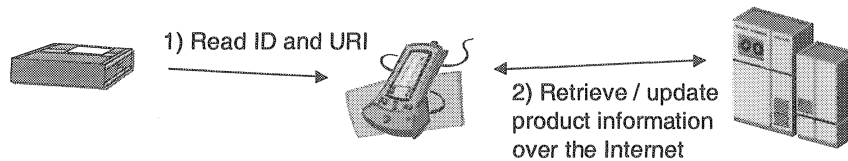

1) Read ID and URI

2) Retrieve / update product information over the Internet

Figure 3 - The Dialog system

The identification part may, in principle, be of any format whatsoever. However, if RFID tags are used it is most convenient to use the fixed id of the tag. RFID tags normally have a globally unique identification number, such as the 64-bit code defined by the ISO standard for RFID tags working at 13.56 MHz. The system then links these codes to the internal references of the company. When operating with bar codes, it is often easiest to use identity coding already in use in the company owning the URI. Examples of this kind of codes are dispatch note references, order numbers, or combined product type and serial numbers.

One weakness of the coding scheme we propose is that it uses two separate codes in order to access the information. However, the two codes can be read in a single operation using automatic reading devices. The solution is easy with RFID technology, if the tag's own identity is used as the ID part. The tag id and the programmable data area, where the URI part can be stored, are easily retrieved in one single read operation. When using bar codes, the identification and the URI can either be coded as two separate bar codes or coded into one single bar code. For coding the two parts into one field, a predefined separator between the codes has to be used. The Dialog project proposes using a coding convention similar to Email addresses, i.e. identification@URI (for instance 12345@dialog.hut.fi) (Främling, 2002). These ways of action enable circumventing the need for attaching and reading two separate codes.

Dialog agents

The URI part of the Dialog product code indicates the location of the tangible object's "agent". The agent is a background service running at the computer indicated by the URI. It offers various interfaces for functionalities like location updates, product information requests, maintenance information requests etc. Each interface has its own characteristics in terms of the information to exchange,

restrictions on data security, authentication, authorisation etc. Therefore security considerations can be treated in different ways depending on how "dangerous" the service provided by each interface is for the Dialog system itself and for the information systems of companies using the Dialog system. The current versions of Dialog agents are programmed in Java, and they provide interfaces for 1) receiving location updates, 2) for linking the Dialog identification part to internal company reference numbers and 3) for retrieving and displaying product information.

As all URI addresses, the one used in the Dialog system can also contain other information concerning the agent service, namely a protocol specification, desired port number and a directory tree. The company that has issued the tangible object is normally the owner of the URI it is carrying, so the protocol (RMI or other), port etc. to be used can be decided by the company without preliminary agreements with other actors. Therefore issues like data security and firewall management remain under the control of the companies themselves instead of being imposed by third-party software companies.

In order to connect to existing information systems, the agents use Java Database Connectivity (JDBC) (Sun Microsystems, 2002), which offers a standardised interface to most existing database products using the standardised query language (SQL). The Dialog system does not require modification of existing information management systems, since it creates its own data structures and links to existing company data when installed. In case similar data structures already exist in the company system, there is a possibility to parameterise database, table and field names so that the Dialog system can use them directly. However, existing systems will normally need some new functionality so that they forward and display the new information to the users of the system. One future functionality that will be offered by Dialog agents is the possibility to automatically manage containment hierarchies, where only the outermost container is tracked, but where the location of all its contents are updated appropriately. Other new functionalities will mainly concern service discovery and negotiation in manufacturing systems, handling of human mobility and other issues of making tangible objects "intelligent".

The strength of Dialog agents is their lean composition. They can be downloaded from the Internet and have a low installation overhead, which is important for scalability. A typical installation of current software components take less than 10 minutes, including configurations for database and reading device connections. Dialog agents exchange information in a peer-to-peer fashion, which also increases scalability, and the degree of scalability should be good enough to allow companies of any size to use them. The system also supports direct data exchange by hand-held devices.

Dialog connection protocols

In the Dialog system, connections between a Dialog agent at a reading point, and the Dialog "tangible object" agent, which contains the product related-

information, are made in a peer-to-peer[15] fashion over the Internet. The Dialog system utilises methods of distributed information management, where information is transmitted directly between the place where it is needed and the place where it is stored. In a peer-to-peer system, the information is usually stored by the party that has created it, so unnecessary copies of information are not made to the companies in the supply network or to intermediate databases operated by third-party companies.

Furthermore, the Dialog always opens up a bi-directional communication between the agents exchanging information. Bi-directional information exchange is needed in a variety of situations. For example, when maintenance operations are carried out, it is important that both the maintenance workers receive the necessary information to perform the work, and that their operations are recorded in the product's data.

In Dialog, the requirements of real-time, bi-directional information exchange are dealt with using synchronised, distributed programming technologies. The current versions of Dialog agents are programmed in Java and use the RMI protocol (Sun Microsystems, 1998). Java and RMI have been selected mainly because they make it possible to develop lightweight software components, which give little network overhead and are easy to implement and install.

Still, the system supports heavy traffic. Tests performed with the current software components indicate that the scalability of the system is sufficient. A low-range computer (Pentium 233 MHz, 80 Mbytes RAM, running Linux with Kernel version 2.2.14 and the MySQL database) was capable of handling 100 simultaneous threads making 10 sequential information requests in 76 seconds. A typically configured server machine is likely to be able to handle requests at least tens of times quicker. So even in the unlikely event of hundreds of connections making information requests simultaneously, the system delays should not become greater than those caused by the Internet connections themselves.

In the future, alternative technologies may also be supported. Examples of such technologies are Corba (Orfali et al, 1997) and SOAP (W3C, 2000). Corba is programming language-independent and of comparable rapidity as RMI, but not as straightforward to use as RMI. SOAP is XML-based and is also programming language-independent, but it is considerably slower than RMI (about five times according to tests performed) and gives more complicated installation procedures. Capabilities to handling other Java-specific technologies like Jini (Oaks & Wong, 2000) and JXTA (JXTA, 2002) are also considered. Jini allows for dynamic service lookup, which is necessary for agents that actively need to discover services, like transportation or assembly services for instance. Communication based on established information exchange standards, at least XML and standards using XML like RosettaNet and ebXML, might also offer interesting solutions. The choice of technology mainly depends on 1) if several programming languages need to be supported, 2) if open XML standards are available for the communication, 3) if communication needs to be bi-directional in real time or not and 4) performance issues.

---

[15] Peer-to-peer computing is not a very well defined concept, but its main idea is to use direct connections between computing devices of any size, thus avoiding the use of intermediate server machines and buffering techniques.

Strengths and weaknesses of the Dialog system

The Dialog system is designed to be quick to take into use, scalable, and flexible enough to be used in many different problem areas. Also, being a peer-to-peer system, Dialog facilitates equality between participating companies. Dialog can be taken into use quickly, because there is no need for the standardisation of the product codes used to connect the tangible items to their agents. There is, also, no need to individually establish connections between agents, as agents are able to discuss over the Internet when the URI is obtained. This increases the scalability of the system, as does the fact that a Dialog agent is rapid to install. For example, for installing a checkpoint agent for tracking application it is enough to download an archive file from the internet, decompress it to a new directory and start it by double-clicking the start-up script (agents can be obtained from http: //dialog.hut.fi/).

Peer-to-peer systems provide more equality between the companies participating in the network. This is because all parties participate on equal terms, no matter what their size is. The biggest companies often dictate information exchange standards and centralise the information in their systems in current collaboration networks, but with peer-to-peer communication this is not the case. Every company retains the ownership of the information they have created, but may exchange the information in a standardised fashion.

The biggest weakness of the Dialog system is the need for it to become widely accepted and used, which is partially the same problem as for the Auto-ID effort. In order to minimise this weakness, the Dialog system is intended to be as open as possible. Therefore project results, software specifications and demonstration software are made publicly accessible at the address http: //dialog.hut.fi/ . The Dialog system is also tested and can already be rapidly taken into use for closed applications.

Due to the growing number of intrusions into protected systems and virus attacks, data security and firewalls have become often addressed issues. However these are not a bigger issue for the Dialog system than for any other Internet-based solution. Security still remains one of the biggest issues to solve for Internet-based systems. The peer-to-peer inspired principles used in Dialog should improve data security since data does not need to be copied between companies and/or passed by third-party companies. Data exchange itself is secured by using existing authentication and encryption technologies.

Server or network down-time are also important issues, especially in tracking applications, where no location update should be allowed to disappear. This is why message persistence (Monson-Haefel & Chappell, 2000) is implemented for all applications requiring it, which guarantees that no messages are lost due to network or server downtime, so they can always be retrieved and sent when the system is operational again.

APPLICATIONS OF THE DIALOG SYSTEM

In this section we present the first two application areas of the Dialog system. So far, the system has been demonstrated on tracking deliveries for international

314

investment projects. It is used for the tracking of transport units, and the system works with both bar codes and RFID tags. Under current development efforts is an extension to the system that enables delivering transportation-related information between the consignors and consignees without the need for traditional systems integration, hence decreasing the costs related to automation of data exchange in transportation. These applications are briefly presented in this section.

Forwarder independent tracking

Customers are increasingly demanding transportation service providers to provide them with information concerning the whereabouts of their goods in transit. Because of this demand many companies have invested tens of millions of dollars in developing tracking systems to keep track the location of consignments (Coia, 2001; Dierkx, 2000; Booker, 1999). However, these systems have two basic weaknesses. Firstly, they work only when the goods are handled inside the company's handling system, and secondly, the information is usually only visible on a www-page, and delays are not updated in the recipient's information system unless time consuming integration efforts are carried out.

Tracking is especially important in project deliveries. This is because the deliveries of project components are always time specific, delays in their delivery may, at the worst, hinder the whole project if the knowledge of the delay is not obtained pro-actively. (Kärkkäinen et al., 2001). Further, the supply networks of investment projects are often so large and complex that it is impossible to use only one forwarder to deliver all the components for the project. The first application area of the Dialog system is in forwarder-independent tracking of investment project deliveries. It is also an interesting application from a system's point of view, since the location of the tangible object is a basic requirement for many other services Dialog agents can perform.

The tracking application has been demonstrated to industrial project partners in November 2001. In April 2002, the system was used in its first pilot implementation. This pilot project involved a producer of heavy machinery, which was the company where tracking information was updated and who owned the URI address programmed into the RFID tags attached to the shipments. Two checkpoints were installed along the delivery chain. The first checkpoint was the packaging department situated in the same country as the heavy machinery producer but located in a different town at a distance of hundreds of kilometres. The second checkpoint was at the destination (construction) site situated in another country. The aim of this pilot was to ascertain the functionality of the system, and to detect the practical problems associated with forwarder-independent tracking.

The codes were programmed into RFID tags, because bar codes can lose their readability in the harsh environment of the project delivery chain. Since RFID tags are used, their pre-programmed serial numbers were used as the identification part of the product code. The checkpoints communicate the location of deliveries directly to the producers system every time an RFID tag is read.

Conclusions of the pilot project were very positive, taking into consideration that it was the first installation in an industrial environment. In addition to the default system set-up time, firewall configuration and database protocol issues

introduced extra installation delay of a couple of days, but these delays were expected beforehand. Main validated features of the pilot installation are that firewalls are usually not a problem at least at the checkpoint end and that the system operates well even through modem Internet connections. Once all three software components were running, the tracking system has been working perfectly and no problems have been encountered during the first month of operational use.

Another tracking-related pilot installation is scheduled for 2002. This pilot is used to test extended functionalities of the system. The pilot installation is concerned with the tracking of "composite containers" (i.e. product inside box, inside bigger box, inside container), where only the outermost container is directly accessible, but tracking should still be performed for all contents of it. Knowing the contents of a container is actually a product information management application, where the container is the "product" and the contents of it are the "product information".

Product centric merge-in-transit

Another interesting application area for the Dialog system is using it as an enabler for merge-in-transit distribution. Merge-in-transit is a distribution model, where shipments originating at different dispatch departments are consolidated into a single customer delivery within the delivery chain (Bradley et al., 1998; McLeod, 1999; Dawe, 1997). This enables fulfilling a customer order with only one drop-off, without the need for warehousing all the goods in one location.

The benefits obtainable with the merge-in-transit model include improved customer service, reduced inventory, reduced transportation costs, and improved supply chain transparency (Bradley et al., 1998). For example, the distribution model can save up to 20% of distribution costs of a maintenance, repair and operating (MRO) goods wholesaler (Kärkkäinen et al., 2002).

Merge-in-transit applications are usually built between a few large companies, because managing the information flows associated with merge-in-transit distribution demands very effective information systems integration (McLeod, 1999; Richardson, 2000). The costs of the integration efforts often prevent smaller companies from participating in merge-in-transit distribution. Thus, larger-scale operations with a high number of suppliers, for example in a wholesaler distribution chain, have yet to address the presented challenges.

In order to make the implementation of a large-scale merge-in-transit system feasible, the delivery information needs to be transferred in electronic form. EDI is a proven concept for transferring such logistics messages, but it is not adopted in smaller companies due to high implementation costs (Tadjer, 1997). It also requires one-to-one connections to be formed between all parties needing to exchange data, which makes it infeasible for large supply networks. The prices of EDI-integration between business applications range from a few hundred euros to tens of thousands (Wilde, 1997; Lankinen, 2001; Seikkula, 2002), depending on the application to be integrated with. Also, when using the batch-oriented EDI, there are severe problems in matching the received information to the physical delivery, i.e. in synchronising material and information flows, in situations where many small deliveries are processed (Johnston and Yap, 1998).

The Dialog system offers a low-cost alternative for enabling a common platform for multiple companies, and simultaneously ensures that material and information flows are synchronous. Shipment identification can be done with the Dialog coding scheme, and delivery data required can be passed over the Internet between the supplier's and handling facility's agents with the Dialog system. The delivery data include the information needed for the handling of the parcel, namely the customer order number that the parcel belongs to and which parcels it is to be merged with, as well as the information of normal shipping notes. Using the information received via Dialog, the warehouse management system can automatically instruct operators on where to transfer the parcel to wait for consolidation.

The implementation costs of the Dialog system are likely to be much lower than those of EDI-messaging. No connection needs to be created between the parties, as the identities of the parcels provide all the information needed to create the connection on the Internet. The only implementation costs of the Dialog system are the costs related to installing the software agent and linking it to the database of the business application.

This makes a merge-in-transit network operating with the Dialog system also more scalable than one based on one-to-one connections. A new member, for example an additional supplier for a wholesaler, can be added simply by installing an agent at her server. The existing members need not make any changes in their application or create any connections with the new member.

The Dialog system was tested in a limited pilot installation with a MRO wholesaler and a logistics service provider that practise merge-in-transit distribution. In the pilot, a client agent was installed on a computer of the logistics service provider, and delivery documents were retrieved from a remote server over the Internet. The installation (including the Java Runtime Environment) took seven minutes, after which the agent was operational.

The representatives of the wholesaler and the logistics service provider saw the following to be the most promising benefits achievable with the Dialog system:

- Removal of costs with EDI transactions
- Facilitation of electronic communications, especially the opportunity for automated communications with smaller partner companies not capable of implementing an EDI-connection
- Availability of up-to-date information everywhere in the chain

Compared to one-to-one messaging approaches, the main benefits in implementing merge-in-transit operations with the Dialog system result from lower costs in implementation and operations, thus making merge-in-transit distribution more attractive for smaller companies.

CONCLUDING REMARKS

Product centric information management is interesting option for managing product related information. In this paper, we have presented a new system for product centric information management; the Dialog system being built at the Helsinki University of Technology. The basic functionalities of the Dialog system

are operational, and applications using it are already being developed. This rapid implementation is possible because the product coding used is based on existing coding standards and therefore need no new central authority to ensure uniqueness and scalability.

The leanness and simplicity of installation of the software components is also important for rapid implementation in the logistics network. This relative maturity signifies that a natural next step is to proceed to extensive piloting and field-testing and simultaneously to try to offer the Dialog system to standardisation bodies.

On the application side, two distinct areas are targeted by our future research efforts. Firstly, we are studying how the product centric system can alter supply chain management, as a majority of the information needed to provide value-adding services to products can be communicated via the products themselves. Secondly, we are trying to find out, what information should be managed and exchanged with the system. More specifically we are trying to address the question: "What information is important to different supply network members?"

Even though the Dialog system is currently focusing mainly on supply chain management and issues related to it, it is not in any way limited to it. The proposed architecture should also be usable as such for handling mobility at a more general level, e.g. human mobility and the finding of information services for tangible objects in general.

REFERENCES

[1]   AirClic, (2002), The home page of AirClic Inc., *http: //www.airclic.com/* , last visited 14th of January 2002

[2]   Andel, T., (2001), "The consumer electronics challenge", *Material Handling Management*, Vol. 56, No. 3, pp. 48 - 49.

[3]   Ashton, K., (2000), "Internet Things - MIT, Embedded Technology and the Next Internet Revolution", *Tag 2000*, 25.5.2000, Baltic Conventions, The Commonwealth Conference & Events Centre, London.

[4]   Auto-ID Centre, (2002), The home pages of the Auto-ID centre of Massachusetts Institute of Technology, *http: //www.autoidcenter.org/main.asp* , last visited 10th of June 2002

[5]   Booker, E., (1999), "Service Maps To Needs – DHL Lets Customers Dictate Which Services The Company Delivers", *Internetweek*, July 12 1999, p. 17.

[6]   Bowersox, D., Closs, D., and Stank, S., (2000), "Ten Mega-Trends that will Revolutionize Supply Chain Logistics", *Journal of Business Logistics*, Vol. 21, No. 2, pp., 1 - 16.

[7]   Bradley, P., Thomas, J., Gooley, T., and Cooke, J., (1998), "Merge-in-transit yields benefits", *Logistics Management and Distribution Report*, Vol. 37, No. 10, p. 30.

[8]   Coia, A., (2001), "Express services unite Europe", *Logistics Management and Distribution Report*, Vol. 40, No. 9, pp. E3 - E5+

[9]   Dawe, R., (1997), "Move it Fast... Eliminate Stepst", *Transportation and Distribution*, Vol. 38, No. 9, pp. 67 - 74.

[10]   Dierkx, K., (2000), "Taking Supply Chain Visibility to the Next Level: Powering the Smart Supply Chain", *Council of Logistics Management Annual Conference*, 26.9.2000, Council of Logistics Management, New Orleans, presentation slides.

[11]   Euwe, M.J. and Wortmann, J.C., (1998), "Logistics control enabled by IT – a window in the future", *Human Systems Management*, Vol. 17, No. 2, pp. 123 - 133.

318

[12] Hamilton, J., (2001), "The European Union's consumer guarantees directive", *Journal of Public Policy & Marketing*, Vol. 20, No. 2, pp. 289 - 296.

[13] Hewitt, F., (2000), "Demand satisfaction communities: New operational relationships in the information age", *International Journal of Logistics Management*, Vol. 11, No. 2, pp. 9 - 20.

[14] Johnston, R. B., and Yap, A. K. C., (1998), "Two-Dimensional Bar Code as a Medium for Electronic Data Interchange", *International Journal of Electronic Commerce*, Vol. 3, No. 1, pp. 86 - 101.

[15] JXTA (2002), "Project JXTA", *http://www.jxta.org/*, Last visited 14 March 2002.

[16] Kärkkäinen, M. and Holmström, J., (2002), "Wireless product identification: Enabler for handling efficiency, customisation, information sharing", *forthcoming in Supply Chain Management: An International Journal*

[17] Lankinen, J., (2001), Telephone conversation with a salesman of a Finnish EDI service provider Oy Datatie Ab, December 5th, 2001.

[18] Kärkkäinen, M., Holmström, J., Främling, K., and Artto, K., (2001), "Intelligent products – a step towards a more effective project delivery chain", *HUT working paper*.

[19] Kärkkäinen, M., Punakivi, M., and Ala-Risku, T., (2002): "Merge in transit - a key for effective order fulfilment in B2B e-commerce", Twelfth International Working Seminar on production Economics, February 18 - 22, 2002, Igls/Innsbruck, Austria

[20] McLeod, M., (1999),"Cutting both ways", *Supply Management*, Vol. 4, No. 7, pp. 24 - 25.

[21] Monson-Haefel, R., Chappell, D. (2000), "Java Message Service", O'Reilly & Associates, USA, 220 p.

[22] Oaks, S., Wong, H. (2000), *"Jini in a Nutshell"*, O'Reilly & Associates, USA, 400 p.

[23] Orfali, Robert, Dan Harkey and Jeri Edwards (1997*), "Instant CORBA"*, John Wiley & Sons, New York, 313 p.

[24] Richardson, H., (2000), "Planning for the year 2000", *Transportation & Distribution*, Vol. 35, No. 11, pp. 73 - 76.

[25] Seikkula, S., (2002), Telephone conversation with a salesman of a Finnish EDI service provider Elma Oyj, February 28th, 2002.

[26] Shapiro, C., and Varian, H., (1999), *Information Rules*, Harvard Business School Press, Boston

[27] Stebbins, J., (2000), "Motorola to use barcode scanner to net connect", *The Australian IT*, June 20, 2000, p. 47.

[28] Sun Microsystems (1998), "RMI Specification". http://java.sun.com/products/jdk/1.2/docs/guide/rmi/spec/rmiTOC.doc.html, Last visited 14 March 2002

[29] Sun Microsystems (2002), "JDBC™ Data Access API", http://java.sun.com/products/jdbc/, Last visited 13 March 2002.

[30] Tadjer, R., (1997), "Shopping around for the best Internet EDI deal", *Network Computing*, Vol. 8, No. 14, pp. 26 - 27.

[31] Tuttle, A., (2002), "Who do you trust", *Industrial Distribution*, Vol 91., No. 3, pp. 17, 20, 22.

[32] Töyrylä, I., (1999), *Realising the potential of traceability – A case study research on usage and impacts of product traceability*, Finnish Academy of Technology, Espoo.

[33] W3C (2000), "Simple Object Access Protocol (SOAP) 1.1", *http://www.w3.org/TR/SOAP/*, Last visited 14 March 2002.

[34] Wicol, (2002), *www.wicol.net*, The www - homepage of Wireless Information Collector WICOL Ltd., last visited 10th of June 2002

[35] Wilde, C., (1997), "New Life for EDI?", *InformationWeek*, Issue 622, March 17, 1997, pp.65 - 67.

[36] Wilder, C., (1999), "Purchasing analyzer", *Informationweek*, May 17, 1999, p. 79.

[37] Zipkin, P., (2001), "The limits of Mass Customisation", *MIT Sloan Management Review*, Vol. 42, No. 3, pp. 81 - 87.