International Conference
on Industrial Engineering and Systems Management

IESM' 2011

May 25 - 27, 2011

METZ- FRANCE

Intelligent Products in Real-Life Applications *

Kary FRÄMLING [a], Juha LOUKKOLA [a], Jan NYMAN [b], André KAUSTELL [c]

[a] *Aalto University, School of Science and Technology, Finland*
[b] *ControlThings Oy, Finland*
[c] *Electrical Building Services Centre, Posintra Oy, Finland*

Abstract

Intelligent Products are capable of collecting information and reacting on it proactively, e.g. estimating needs for maintenance or repair. With increased computing power and communication capabilities, products may also become pro-active. We claim that when such products will have the means to communicate between themselves and with other systems, only then will the true potential of the Internet of Things concept have been accomplished. In this paper we describe a number of real-life applications that have been implemented using the Intelligent Product and Internet of Things concepts. These applications reveal some of the potential for the future of these technologies.

Key words: Intelligent Products, Internet of Things, Closed-loop PLM, Product Instance, Middleware

1 Introduction

For Product Lifecycle Management using Smart Embedded Systems [13], the number and variety of information systems that need to communicate is greater than in most other application domains. Indeed, it is not one application domain; it is, rather, a collection of application domains that need to use and share partially identical information. The Internet of Things has been proposed as a concept that encompasses at least some parts of the application domains considered. The Internet of Things will be an extension of the Internet that makes it possible to access information about any tangible "thing" over the Internet. The Internet of Things concept was probably first coined by Ashton (2000) but other early and publicly accessible sources are e.g. [9] and [8]. Still, earlier concepts such as "Ubiquitous Computing" proposed in 1991 [21] and "Mirror Worlds" proposed in 1993 [7] contain many common elements with the Internet of Things concept.

The notion of Intelligent Product is still rather undecided and we don't currently have a single established definition [4]. However, there is an increasing need to treat products as instances due to mass customization and the efforts to invest in after-sales services. In each case, a specialization of the product occurs – whether it be in terms of different functionalities, different delivery paths, different usage modes – making it increasingly important to maintain information unique to each item in some way. Even two initially identical product items will have different owners or be used in different conditions. The product usage phase may even begin before it is sold because many products nowadays go through an individual initialization procedure where "normal" operation profiles are recorded and stored. The product's control system can also be fine-tuned so that individual

---

* This paper was not presented at any other revue. Corresponding author Kary Främling. Tel. +358505980451.
Email addresses: Kary.Framling@tkk.fi (Kary Främling), jmloukko@cc.hut.fi (Juha Loukkola), Jan.Nyman@controlthings.fi (Jan Nyman), andre.kaustell@posintra.fi (André Kaustell).

differences in sensor outputs and actuator operations are taken into account, as for engine control in modern cars that continues adapting the control system to changes in the engine during its whole lifetime.

The focus of this paper is to describe a representative number of real-life use cases where Intelligent Products and the Internet of Things are joined together. We evaluate those use cases against the classification provided in [17] for assessing how advanced current Intelligent Products are, thereby also providing an attempt to foresee in what direction those Intelligent Products can be expected to evolve. After this introduction, Section 2 exposes the background and existing definitions of Intelligent Products, Section 3 describes how Intelligent Products were implemented in the context of Closed-loop Product Lifecycle Management, while Section 4 presents a number of real-life applications where the mentioned concepts have been applied, followed by conclusions.

2 Background

It seems like Intelligent Products were first discussed in an after sales and service context in 1988 by Ives and Vitale [10]. The first examples of Intelligent Products in the after sale context were computers running programs that tracked the configuration and performance, and could request for service and maintenance. The benefits in efficiency of service and reliability of operation could be substantial and was the basis for successful start-ups and new lines of business for established companies.

Only later did the idea of integrating intelligence and control into the product spread to manufacturing [16] and supply chain control [14]. In these application domains, new auto identification (Auto-ID) technologies, such as Radio Frequency Identification (RFID) have made the tracking and tracing of products throughout the entire supply chain possible. When product individuals in a logistic/production setting are not only given a traceable individuality, but also the associated content (e.g. delivery terms, contract terms, exceptions, etc.), and also decision power is delegated to them, we enter the realm of Intelligent Products.

Such Intelligent Products have the means to communicate between themselves and also with logistic service providers. Intelligent Products link the Auto-ID technology to the agent paradigm and Artificial Intelligence. Agent technology has already been considered as an important approach for developing industrial distributed systems (e.g. intelligent manufacturing systems) [11, 12]. Intelligent Products can also play an essential role in product lifecycle management by their capability of collecting usage information and reacting on it proactively, e.g. estimating needs for maintenance or repair [18]. By using sensor technologies the conditions of products can be continuously monitored. The access to information on how products have been used could significantly improve the way that products are recycled when they arrive to their end-of-life. Sensor technologies can also contribute to improvements in manufacturing nodes and to the logistics of the entire supply chain, by giving real-time status information (e.g. identification, location and other conditions) of the products.

What is common to such tracking and tracing in the supply chain and to product lifecycle management is that information needs to be represented at the item level and communicated between different organizations. From an information system perspective, a shipment is indeed just a “product” with a relatively short lifecycle, where the actual products that were included in the shipment may have a much longer lifecycle. However, currently used information systems typically focus on managing batches and accounts using centralized databases, hence representing item-level information and communicating it between organizations can be a challenge for them, in case of mass-customization of products. Therefore, there is increasing interest in the development of Auto-ID technologies and Intelligent Products which is being reflected in on-going work, current project proposals and future research areas.

There has been several proposals for how to classify “how intelligent” and intelligent product is [14, 16, 20]. In this paper we will use the three-dimensional classification proposed in (Fig. 1) [17] to assess what level of Intelligent Products has been achieved in the case studies of Section 4. The three dimensions are:

1. *Level of intelligence*: how “intelligent” the product is.

2. *Location of intelligence*: whether the product has network connectivity and to what extent the "intelligence" is embedded into the product versus located on network-connected servers.

3. *Aggregation level of intelligence*: whether the "product" is a self-contained instance only or if it aggregates information and "intelligence" also of its constituents. For instance, a modern vehicle contains many subsystems that are usually aggregated to the vehicle level, a shipment may contain many items that are Intelligent Products by themselves but aggregated together so that the shipment is the "gateway" to that information.

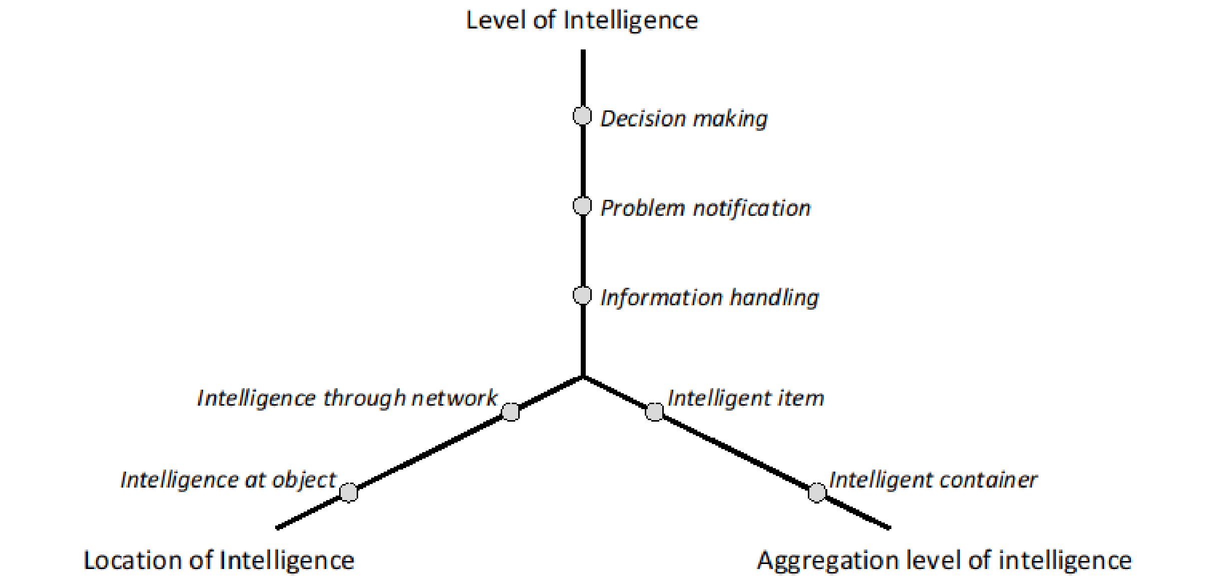

Fig. 1. Classification of Intelligent Products according to [17].

3 Intelligent Products implemented by DIALOG

One of the main challenges with the Internet of Things is how to access the information that is not stored locally in the thing itself but is available over the Internet. In 2001, an ID@URI notation and the associated DIALOG information system were developed at Helsinki University of Technology (now a part of Aalto University). DIALOG made it possible to query and update product information about tangible things over the Internet throughout the product lifecycle as shown in Fig. 2. DIALOG was defined based on experience gained from earlier e-commerce projects where computer programs had been developed based on the peer-to-peer paradigm mainly for exchanging sales forecasts between different organizations. The initial application area of DIALOG was to develop a forwarder independent tracking-and-tracing system for worldwide project deliveries using RFID technology in 2002.

DIALOG is a "generic" software in the sense that it provides protocol- and interface-neutral message passing mechanisms with message persistence functionality, security mechanisms etc. that are abstracted away from the "business logic" itself, implemented by "agents" (Fig. 3). This signifies that different protocols and messaging interfaces can be easily supported. Due to DIALOG's generic architecture, it has also been used for collecting in-use and product lifecycle information in various application areas such as automotive, white goods, energy consumption monitoring, weather information collection etc.

For the Internet of Things to become a reality, a distributed messaging architecture with standardised communication interfaces needs to be created for the purpose of product tracking and product data gathering. This is what was done e.g. in the PROMISE EU project through the creation of a new messaging interface, called PROMISE Messaging Interface (PMI) [19]. In the PROMISE world (Fig. 4), the messaging between the participants, e.g. products and the Product Data Knowledge Management systems, is done by passing messages between nodes over a messaging infrastructure defined by the PMI. PMI uses self-contained messages that can in principle be sent over any protocol, such as HTTP, SOAP, SMTP or similar. The PMI cloud in Fig. 4 is intentionally drawn in the same way as is usual for the internet cloud, i.e. PMI is intended to play the same role in the Internet of Things as HTTP (and partially HTML) does for the internet.

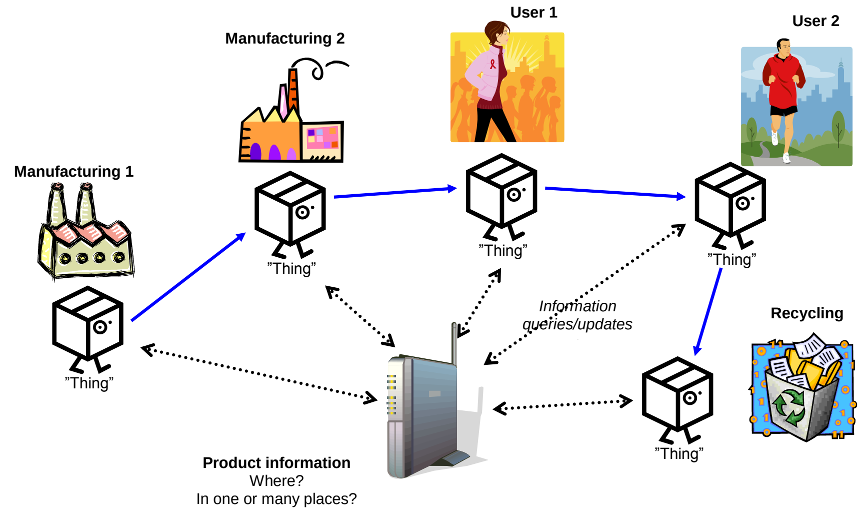

Fig. 2. Internet of Things. The Thing is the unique instance with its properties, while the different users of that Thing have different views and interfaces to it.

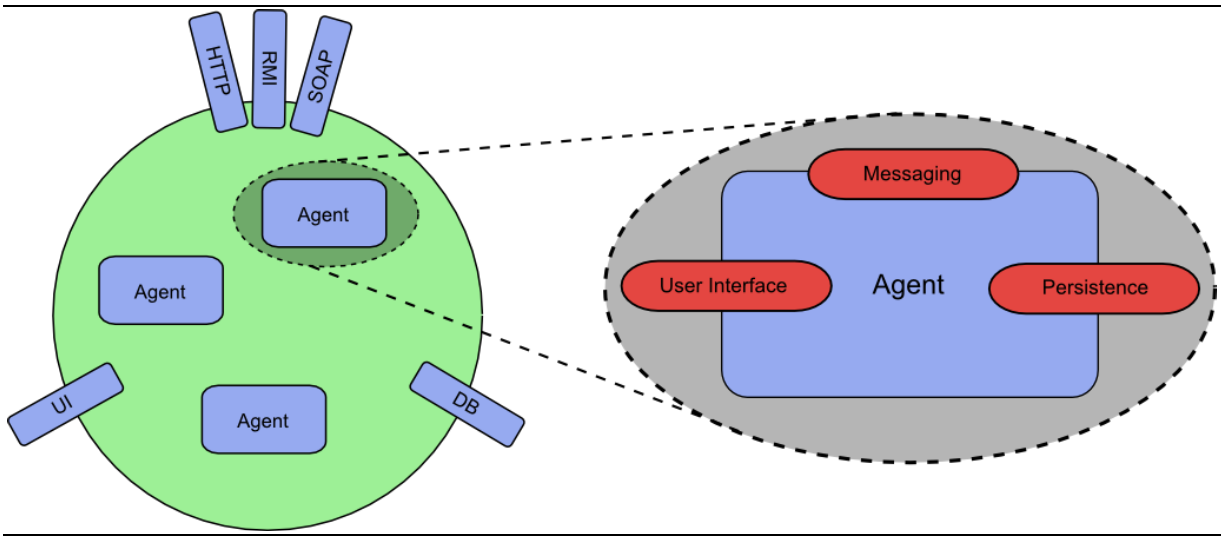

Fig. 3. DIALOG-node and agent with their external interfaces.

A defining characteristic of PMI, as such, is that nodes do not have predefined roles, as it follows the "peer-to-peer" approach to communications. That means that products can communicate directly with each other or with back-end servers but PMI can also be used for server-to-server information exchange of sensor data, events and other in-use product information. A "full" PMI node capable of sending as well as receiving requests does have to include both HTTP client and server functionality, but a more limited node can just have the HTTP client functionality, if it is assumed that it will only send messages to other nodes. An example of such "limited" nodes are ones associated with RFID tag readers, or generally, nodes that are unreachable from the outside because of a firewall, which periodically send product data to a product monitoring system according to a "subscription" that is specified when the product is installed.

PMI defines different operations such as a read or write of the value of a particular info item. The info items represent actual values such as sensor readings of a device, such as a car. A PMI node is a communications end-point in a PMI network, and manages communications for one or several devices. The parameters for the method calls are XML strings whose structure is defined by an XML schema. The XML string conveys additional request information, such as the involved device, information item, sub-type of request, etc.

In addition to reads and writes, PMI also provides callback methods for asynchronous communications. Examples of asynchronous communications include a ”subscription” read, a call to the read method with parameters that specify that the target node should not respond directly with an value, but rather send multiple responses at a specified interval. The callback method interface also provides a mechanism for nodes to send events to each other (with or without a prior subscription, subject to the particular node implementation). Indeed, the callback method is an embodiment of the well-known Observer pattern [6] applied to messaging systems as proposed e.g. in [1,2].

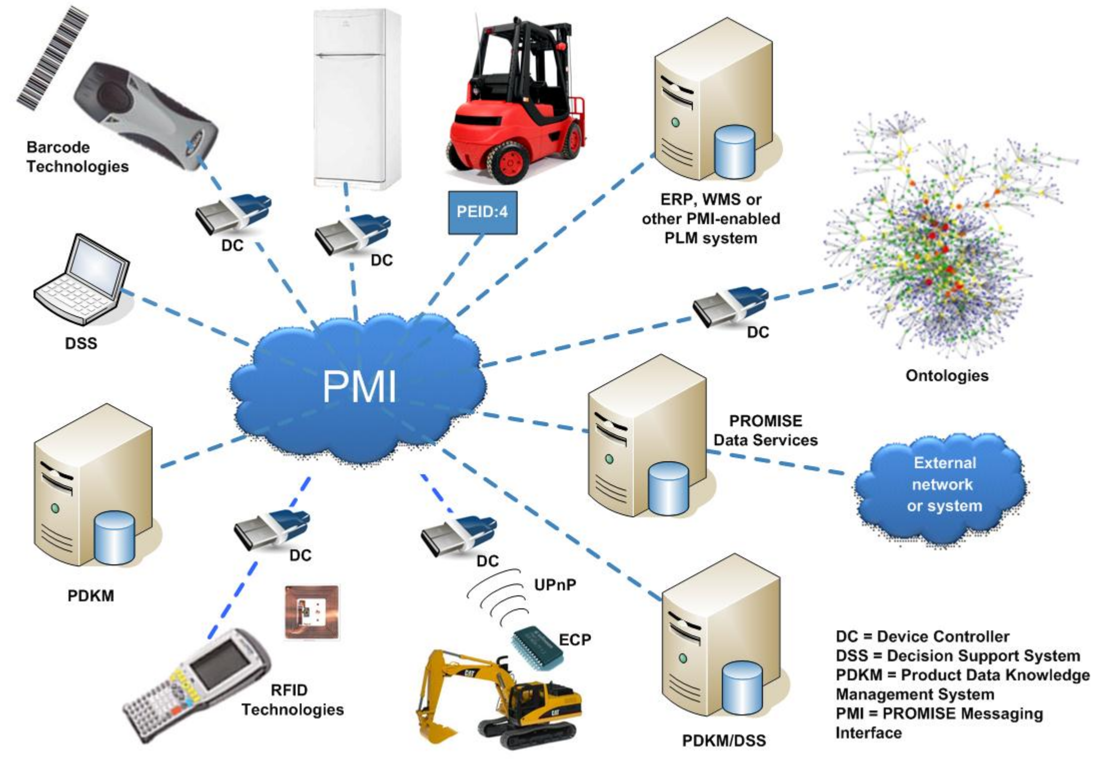

Fig. 4. Illustration of PROMISE architecture and connectivity [19].

4 Applications

Several Internet of Things applications have been done based on DIALOG since 2001, mainly for RFID-based tracking and tracing in supply chain management [9, 15]. However, in this paper we focus on applications where the product instances have at least some embedded computing power and possibilities to communicate over the internet either directly or through “proxies” or “gateways” that implement device-specific Device Controllers such as those shown in Fig. 4.

4.1 Automotive

In the PROMISE project a system was developed on DIALOG for connecting to vehicle Engine Control Units (ECU) over Bluetooth on mobile phones and sending that ECU information onwards to backend systems of e.g. service companies or manufacturers using PMI [3]. The OBD-II standard was used for communicating with the ECU so the system was applicable to all newer cars and some other vehicle types.

In 2009 the same system was implemented together with Finnish companies who are owners and users of big fleets of vehicles. One of the companies also handles service and maintenance of all vehicles. Professional-level data collection equipment from several different manufacturers has been installed into various vehicle types, ranging from specialized vehicles used at the Helsinki airport and trucks of various sizes used by a major Finnish logistics service provider. Protocols of different data collection equipment are implemented in DIALOG, while the actual information handling is handled in a uniform way for all of them by dedicated agents. The agents can filter, store, detect events etc. with the incoming information. They can also forward the needed information to other systems, such as maintenance scheduling or route planning systems, using PMI. Finally, agents can also provide support for graphical user interfaces, such as the one in Fig. 5 that shows the location and status in real-time of different vehicles at the Helsinki-Vantaa airport. Examples of status information are if the engine is running, if the vehicle is in use or available, as well as sensor values, alarms and other relevant information.

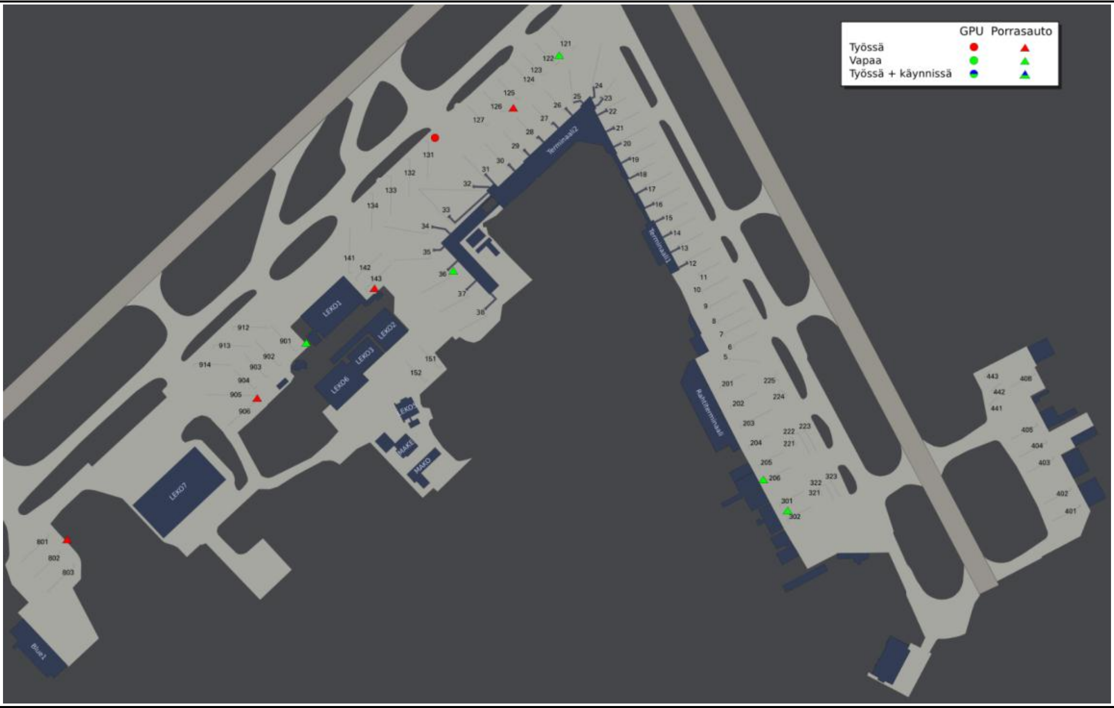

Fig. 5. Map view of nine staircase vehicles and one other vehicle.

In this application, the *location of intelligence* is mainly through the network, rather than at the object. However, as the processing power of the vehicle-integrated systems increases, more intelligence can also be implemented at the object. Remote updating of the embedded software e.g. through OTA (over-the-air) is also being implemented, which means that the intelligence at object can even be increased over the vehicle lifetime. Because the intelligence is already now distributed over many systems and agents, the limit between intelligence at object and intelligence through network is expected to become increasingly obscure.

The *aggregation level of intelligence* is mainly “intelligent item”. However, when a trailer is connected to a truck, the truck is both an item and a container where the trailer may be an Intelligent Product on its own.

The *level of intelligence* currently implemented includes both information handling and problem notification. In this context, examples of decision making functionality is the detection of upcoming failures, condition-based service and maintenance scheduling, dynamic route planning and driving style analysis. Such functionality is currently being implemented but is technically more challenging than the information handling and problem notification levels. True decision making capabilities can be implemented either by encoding expert knowledge directly or by creating it through data analysis and machine learning techniques. In practice, at least data analysis is usually necessary due to the great volumes of data generated. Obtaining the required data signifies that at least the information handling functionality has been running for a sufficiently long time. Automatic detection of upcoming failures, for instance, requires that data has been acquired from some time before the failure has occurred and that the diagnostics information about the failure is available.

4.2 Smart houses and Appliances

A demonstrator combining the messaging infrastructure with smart appliances was created as a part of the PROMISE project. To create the demonstrator, support for data gathering from a prototype of a refrigerator's statistical data collection unit was integrated to the PMI-Dialog messaging system. The data collection unit prototype was provided by Indesit, a major European domestic appliance manufacturer, and has functionality similar to what can be expected of smart household appliances available in the near future. The prototype implements various technologies that are patented by Indesit.

The prototype in question is a device that is going to be integrated with ordinary household refrigerators. By integrating to the refrigerator a device capable of collecting, storing and communicating statistical data obtained from the various embedded sensors, the manufacturer plans to enhance final product testing performance at the manufacturing plant, as well as the possibility of offering a preventive maintenance service contract to customers. This latter business scenario is the most interesting from a distributed systems point of view.

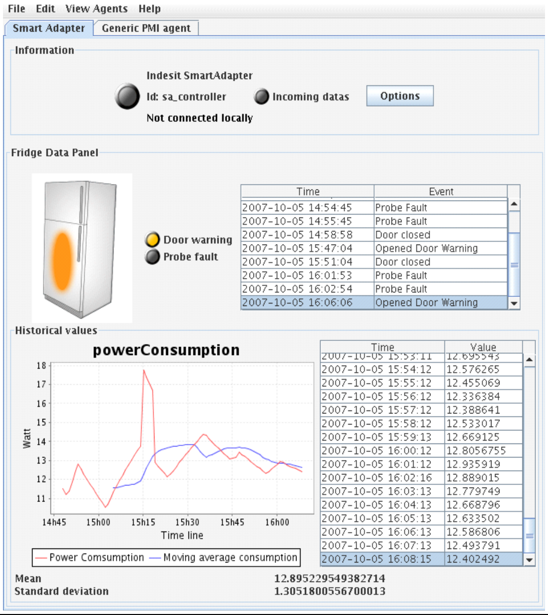

Fig. 6. Information gathered from the refrigerator-embedded information device displayed on a product agent's user interface.

The refrigerator-embedded information device is interfaced with the Dialog software using a purpose-built device controller agent, which provides a way of reading the values of the information items that the agent provides. For supporting actual I/O with the information device, the serial communications protocol running on RS-232 was implemented using the Java Communications API. For gathering data from the product embedded information device, a product agent was created. The agent's graphical user interface can be seen in Fig. 6. The product agent has subscribed to receive alarm events detected by the refrigerator-embedded device and continuous measurement data on the electrical power consumption of the device. From this kind of real-time data a product agent coupled with a data mining system could deduce the operating status of the refrigerator.

Several other prototypes contributing to a smart house have also been built around the same principles. Examples of these are data collection and transfer systems for air handling units, weather monitoring stations and wireless electricity monitors. The air handling unit contains several sensors which provide the on-board logic with valuable information, but can also forward information to a server which in turn can make higher level decisions and coordination of several devices. In this setup there are several components which can be placed in the classification by [17]. The intelligence of the measurement device is non-existent, but the gateway used to locally collect the data is, on the aggregation level, an *intelligent container*. On the level of intelligence we classify it between *information handling* and *problem notification*. The gateway then sends data to a server, which in turn is providing *intelligence through network*. Service orchestration turns the devices with low intelligence into a smart house, location-wise distributed, but enabling *decision making* on the intelligence dimension. It covers all *aggregation levels*, as it contains both *intelligent items* and several *intelligent containers*.

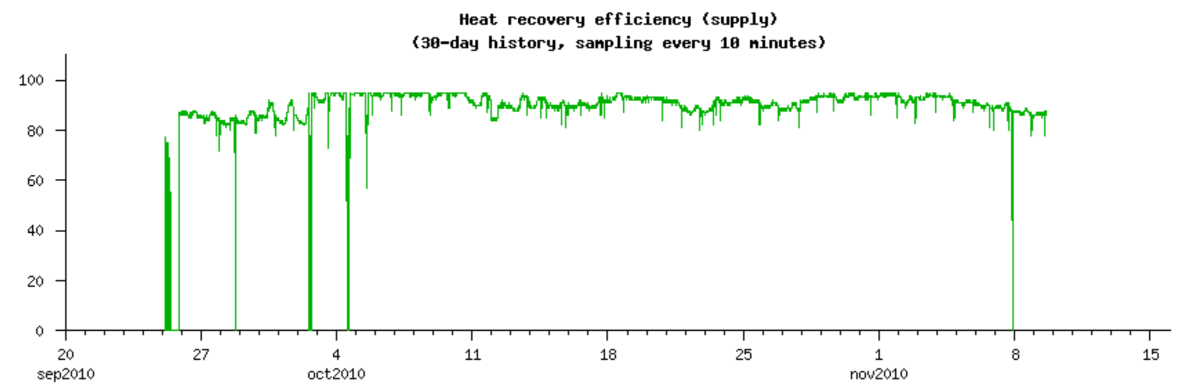

Fig. 7. Example of monitored data from air handling unit.

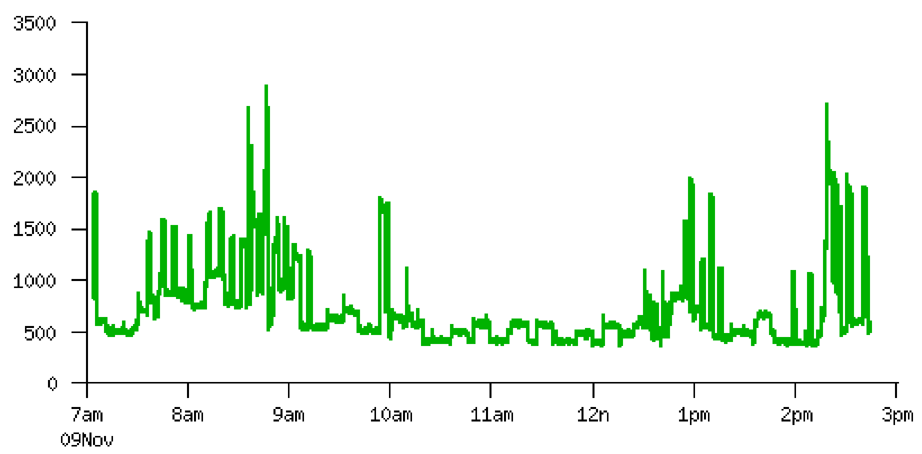

Fig. 8. Example of real-time energy monitoring plot, generated from data with 5 second sampling interval.

5 Conclusions

As illustrated by the real-life applications presented in the paper, intelligent products are moving rapidly into daily use both in industry and in everyday life. Furthermore, they tend to achieve the highest levels for all dimensions according to the classification in Fig. 3. The only dimension that is not fully implemented is the *level of intelligence*, where the *problem notification* level is partially covered while the *decision making* level is still largely missing. Current and future work on Intelligent Products is therefore expected to focus on developing these missing levels of intelligence. The increased availability of in-use data from products will enable new possibilities for data analysis, machine learning and decision making functionality. We expect that such applications will finally enable successful use of artificial intelligence and machine learning technologies in real-life applications.

Another key factor for Intelligent Products to be successful is the apparition of standardized interfaces for collecting and exchanging in-use product information. PMI is an interface that has been specified with this particular objective in mind. It is one of the technical foundations of the Quantum Lifecycle Management (QLM) work group of the Open Group (www.opengroup.org/qlm/) whose aim is to promote such interface standards. Without such a standard it will not be possible to fulfill the *intelligence through network* functionality in a universal way for all Intelligent Products. PMI is one proposal for such an interface standard but other proposals exist (see e.g. [5]) and will probably be proposed in the future.

6 References

[1] Kary Främling, Mikko Kärkkäinen, Timo Ala-Risku, Jan Holmström. Agent-based Model for Managing Composite Product Information. Computers in Industry, 57(1):72-81, 2006.

[2] Kary Främling, Timo Ala-Risku, Mikko Kärkkäinen, Jan Holmström. Design Patterns for Managing Product Life Cycle Information. Communications of the ACM, 50(6):75-79, 2007.

[3] Kary Främling, Jan Nyman. Information Architecture fo Intelligent Products in the Internet of Things. In: Autere V., Bask A., Kovács G., Spens K. & Tanskanen K. (eds.): Beyond Business Logistics; Proceedings of 20th NOFOMA logistic conference , Helsinki, Finland, 5-6 June, 2008. pp. 221-232.

[4] Kary Främling, Duncan McFarlane. Editorial for Special Issue on Intelligent Products. Computers in Industry, 60(3):135-136, 2009.

[5] Kary Främling, Ian Oliver, Jan Nyman, Jukka Honkola. Smart Spaces for Ubiquitously Smart Buildings. In: Proceedings of Third International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM) , Sliema, Malta, October 11-16, 2009. pp. 295-300.

[6] Gamma, E., Helm, R., Johnson R., and Vlissides, J., 1995: Design Patterns, Addison-Wesley, Reading, MA.

[7] David Gelernter. Mirror Worlds: or the Day Software Puts the Universe in a Shoebox...How It Will Happen and What It Will Mean. Oxford University Press, USA, 1993. 256 p.

[8] N. Gershenfeld, R. Krikorian, and D. Cohen. The internet of things. Scientific American, 291(4):76-81, 2004.

[9] Eero Huvio, John Grönvall, Kary Främling. Tracking and tracing parcels using a distributed computing approach. In Proc of NOFOMA'02, pages 29-43, 2002.

[10] B. Ives and M.R. Vitale. After the sale: Leveraging maintenance with information technology. MIS Quarterly, 12(1):7-21, 1988.

[11] NR Jennings, JM Corera, and I. Laresgoiti. Developing industrial multi-agent systems. In Proc. of the First International Conference on Multi-agent Systems, pages 423-430, 1995.

[12] Nicholas R. Jennings and Michael J. Wooldridge. Applications of intelligent agents. In Nicholas R. Jennings and Michael J.Wooldridge, editors, Agent Technology: Foundations, Applications, and Markets, pages 3-28. Springer-Verlag: Heidelberg, Germany, 1998.

[13] D. Kiritsis, A. Bufardi, P. Xirouchakis. Research issues on product lifecycle management and information tracking using smart embedded systems. Advanced Engineering Informatics, 17(3-4):189-202, 2003.

[14] M. Kärkkäinen, J. Holmström, K. Främling, and K. Artto. Intelligent products - a step towards a more effective project delivery chain. Computers in Industry, 50:141-151, 2003.

[15] Mikko Kärkkäinen, Timo Ala-Risku, Kary Främling, Jari Collin, Jan Holmström. Implementing inventory transparency to temporary storage locations: a solution design experiment in project business. International Journal of Managing Projects in Business, 3(2):292-306, 2010.

[16] D. McFarlane, S. Sarma, J.L. Chirn, CY Wong, K. Ashton. Auto id systems and intelligent manufacturing control. Engineering Applications of Artificial Intelligence, 16(4):365-376, 2003.

[17] Gerben Meyer, Kary Främling, Jan Holmström. Intelligent Products: A survey. Computers in Industry, Volume 60(3):137-148, 2009.

[18] PROMISE. http://www.promise.no/.

[19] PROMISE Architecture Series Volume 1: Architecture Overview, 2008, accessed May 12th 2009: http://cl2m.com/system/files/private/PROMISE AS Volume 3 Architecture Reference PMI.pdf

[20] Olli Ventä. Intelligent products and systems. Technical report, VTT, 2007.

[21] Mark Weiser. The computer for the twenty-first century. Scientific American, 265(3):94-104, September 1991.