

# Enhanced Product Lifecycle Information Management using “communicating materials”

Sylvain Kubler<sup>b,\*</sup>, William Derigent<sup>a</sup>, Kary Främling<sup>b</sup>, André Thomas<sup>a</sup>, Éric Rondeau<sup>a</sup>

<sup>a</sup>Université de Lorraine, CRAN, UMR 7039, Vandœuvre-lès-Nancy, F-54506, France

CNRS, CRAN, UMR 7039, Vandœuvre-lès-Nancy, F-54506, France

<sup>b</sup>Aalto University, TUAS Building, Otaniementie 17, Espoo, Finland.

P.O. Box 15500, FI-00076 Aalto, Finland

---

## Abstract

With traditional PLM (Product Lifecycle Management), people think towards the future: first comes product development, then manufacturing, then support and finally disposal – data flows only in forward direction. With the CL<sub>2</sub>M (Closed-Loop Lifecycle Management) concept, there are also flows going backwards, enabling better visibility and control of the product throughout its PLC (Product Life Cycle). CL<sub>2</sub>M uses “intelligent products” to gather information at any instant of the PLC and to make it available on downstream or upstream PLC phases. However, information is often deported on a database and is accessed remotely via a network pointer carried by the product. To bring the CL<sub>2</sub>M concept a step further, this paper addresses a dual challenge: *i*) providing a new kind of intelligent material capable of undergoing physical transformations without losing its communication ability and the data that is stored on it, *ii*) providing a framework to achieve a high degree of data synchronization (i.e. enabling data updates on the product, regardless of the network availability). An applicative scenario is presented, showing how this kind of material is put into practice in the context of “communicating medical garments” and how the requirements of the desired degree of data synchronization can be reached.

*Keywords:* Product Lifecycle Management, Closed-Loop Lifecycle Management, Intelligent product, Communicating material, Data synchronization

---

## 1. Introduction

Today, the study of the product life cycle (PLC) is an integral part of the company strategy to plan, design and manage their activities more effectively. In general, the PLC consists of three main phases: Beginning of Life (BoL), including design and production, Middle of Life (MoL), including use and maintenance, and End of Life (EoL), including recycling and disposal. It is a major challenge for a company to get real control of their products throughout their PLC. To enable such a control, the concept of Product Lifecycle Management (PLM) appeared in the 1990's as an extension of PDM (Product Data Management) [1, 2, 3]. However, over the last decade, the research consortium involved in the FP6 IP 507100 project PROMISE (PROduct lifecycle Management and Information tracking using Smart Embedded systems) claimed that traditional PLM systems have a lack of product knowledge and visibility in both MoL and EoL phases. That is

even more important today since companies recognize that future profits will not come from the manufacture of products (i.e. from BoL) in developed countries [4]. Companies in countries where labor costs are 10% or 20% of those in Europe will be able to carry out manufacturing activities at a much lower cost. However, production is only one phase of the PLC and there are other areas where companies can add values. They can, for instance, develop ideas for new environment-friendly products, provide customized and advanced products and improve the customer experience. The PROMISE consortium reported that with emerging technologies such as wireless sensors, telecommunication and product identification technologies, it is possible to expand the scope of traditional PLM to fill the existing gaps. This extension gave rise to the concept of “closed-loop PLM” [5], whose breakthrough challenge is to enable the information flow to include the customer and to enable the seamless transformation of information to knowledge. For instance, designers (in BoL) could be provided with real-time data about the conditions of use of their products (i.e. information from MoL) and of retirement (i.e. information from EoL). This is an important step in improving future product designs and product generations [5, 6]. This concept was renamed CL<sub>2</sub>M (Closed-Loop Lifecycle Management) [7].

---

\*Corresponding author

Email addresses: [sylvain.kubler@aalto.fi](mailto:sylvain.kubler@aalto.fi) (Sylvain Kubler),  
[william.derigent@cran.uhp-nancy.fr](mailto:william.derigent@cran.uhp-nancy.fr) (William Derigent),  
[kary.framling@aalto.fi](mailto:kary.framling@aalto.fi) (Kary Främling),  
[andre.thomas@cran.uhp-nancy.fr](mailto:andre.thomas@cran.uhp-nancy.fr) (André Thomas),  
[eric.rondeau@cran.uhp-nancy.fr](mailto:eric.rondeau@cran.uhp-nancy.fr) (Éric Rondeau)

Closing the product lifecycle information loops offers the possibility to analyze the different information feedbacks to extract new product knowledge. However, that requires the identification of a specific “item” from a given product family<sup>1</sup>, from its manufacturing to its disposal. For this to happen, Product Lifecycle Information Management (PLIM) systems have emerged in recent years [8]. Generally speaking, these systems associate a unique identifier (UID) to each produced item, which is a practical means of accessing remote item-related information. This principle is depicted in Figure 1. Today, in many applications, intelligent products/items only provide a network pointer (e.g. via a RFID tag) to a linked database and a decision making software agent [10, 11]. In such a situation, it is necessary to have a permanent connectivity of the item with the associated databases. However, it is not an easy matter to fulfill such a requirement in the context of PLC. Indeed, an item passes through numerous companies, undergoes various operations and is used in different ways, which prevent connectivity to the right database at the right time (e.g. a matter of network availability, of access rights, . . .).

To bring CL<sub>2</sub>M a step further, a dual challenge is addressed in this paper. The first one aims at providing a new kind of intelligent material capable of undergoing physical transformations without losing its communication ability and the data that is stored on it. Indeed, in the case where a standard product (e.g. a product fitted with a unique RFID tag) is cut<sup>2</sup>, the resulting pieces lose all product-related data and their communication ability. The proposal of such a new material has been studied in previous work [12, 13], giving rise to the “communicating material” paradigm. This paradigm and related tools are presented in section 2. The second challenge aims at providing a framework for data synchronization. Indeed, storing information on the item inevitably generates data replicas in the system that must be managed properly. To get the most out of the “communicating material” paradigm, data synchronization is one of the important techniques to be developed. Section 3 introduces the existing data synchronization strategies and then details how the “communicating material” paradigm makes it possible to reach a high degree of data synchronization. Finally, an applicative scenario in the framework of medical garment lifecycle is presented in section 4, which helps to show that the requirements of the desired data synchronization are met, and that the information continuum throughout the PLC is improved.

<sup>1</sup>The term “item” could also be referred to as “individual product” or “product instance” [8, 9]. For consistency purposes, “item” is used all along this paper.

<sup>2</sup>The term “standard products” is used in this paper to make the difference with products/items made of “communicating material”.

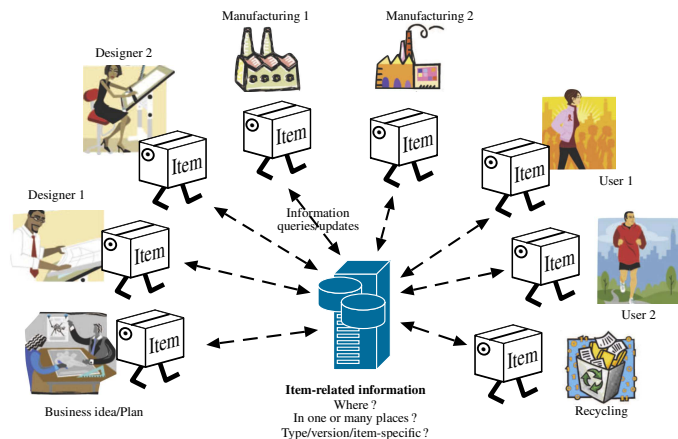


Figure 1: Illustration of a PLIM system

## 2. “Communicating material” and related work

The new paradigm of “communicating material” studied in [13] changes drastically the way to see the material that the item is made of. It aims at giving two main abilities to the material:

- *the ability of being intrinsically and wholly communicating*: even if the product undergoes physical transformation, the resulting pieces shall still be able to communicate,
- *the ability of managing itself its data*: pushing the “communicating material” paradigm to its extreme, the material should be able to manage its own data according to the events occurring in its environment. For instance, the material could decide itself to propagate/replicate specific data onto different material parts because a physical transformation is scheduled, thus avoiding data loss. Another example would be the mutation of the data when adverse events occurs.

This vision is far from being possible today, especially due to the technological limitations, but some current research seems to be promising. For instance, [14] studied a manner of giving the ability to the material to be intrinsically and wholly communicating, spreading a huge amount of RFID  $\mu$ -tags into the material. A prototype of “communicating textile” has been designed ( $\approx 1500 \mu\text{tags}/\text{m}^2$ ), making possible the dissemination of item-related information on all or parts of the textile<sup>3</sup>. Figure 2 shows this prototype being manufactured. Although current technologies do not allow building a communicating material up to the molecular level, the designing of new strategies as

<sup>3</sup>In previous work, the choice that has been made to get as close as possible to the “communicating material” paradigm was to spread a huge quantity of  $\mu$ -tags into the material, but other techniques or technologies could further be imagined and developed. For instance, a new form of “communicating wood” is currently studied in [15], where Nuclear Quadrupole Resonance is used for mass marking, thus enabling new identification code strategies.

those mentioned regarding the second ability (of managing itself its data) may be addressed. In this sense, our previous work developed solutions/tools to select the context-sensitive information at any stage of the PLC, and then to split it on all or parts of the “communicating material”. These solutions/tools form a two-step information dissemination process, which is described in a summarized form in sections 2.1 and 2.2.

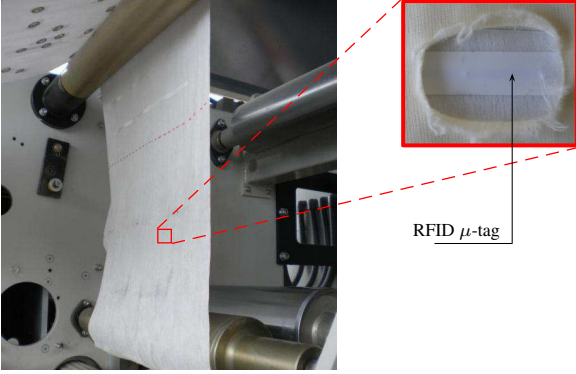


Figure 2: Prototype of “communicating textile” designed in [14]

### 2.1. Process step 1: relevant data identification

Abundant research in the framework of database systems deals with the problem of data distribution [16], but the approaches too often neglect the context of use of the data to assess its “relevance” to be stored on the mobile database/device [17, 18, 19, 20]. This problem is particularly important in the framework of PLC since the item’s environment dramatically changes. Indeed, an item moves through numerous companies with various core business sectors and many information systems [21, 4]. The appropriate information to embed on the item is therefore depending upon a variety of factors (user concerns, item’s environment, ...). Accordingly, an approach was developed in previous work [13] to assess such a relevance. This relevance gives the degree to which information might be useful for the subsequent actors of the PLC and, accordingly, might be useful to be stored on the item itself. The essential elements of this approach are described hereafter.

First, let us note that this approach uses the Logical Data Model (LDM). Figure 3 gives insight into a part of such an LDM. A given LDM entity corresponds, once implemented, to a relational table as shown in Figure 3 with the table/entity `MaterialDefinition`. The attributes listed in each entity correspond to the table columns, each row is referred to as a “tuple” and a table cell is called a “data item”. In this example, `MaterialDefinition` has 4 attributes and 4 tuples (i.e. 16 data items). In our approach, the relevance is computed at the level of the data item. However, only tuples related to the item (i.e. to the individual product) are assessed<sup>4</sup>. In this paper, these tuples

<sup>4</sup>An algorithm is developed in [13] to identify such tuples from the database.

are referred to as “item-related tuples”. For instance, in Figure 3, only tuple 3 is identified as an item-related tuple (represented with dashed background) and thus, only the data items that compose this tuple are assessed in term of relevancy. In our approach, several criteria are defined to represent the context of use of the item (actor’s expectations, LDM characteristics, geographic location of the item, *etc.*). For each criterion, the calculation of a relative priority  $\rho_x$  (with  $x$  a criterion) and an assigned priority  $\phi_x(l)$  (with  $l$  a data item) are performed. The first priority (relative) indicates how important the criterion  $x$  over the others is. The second one (assigned) corresponds to the priority value (between 0 and 1) computed for each data item  $l$  with respect to criterion  $x$  (with  $l = \{1, \dots, n\}$ ,  $n$  being the number of item-related data items from the database). Both priorities are then combined as in equation 1, which provides the relevance degree  $R(l)$  of data item  $l$ , with  $\text{size}(l)$  the size of data item  $l$  (expressed in bytes). This formula shows that the higher the priorities  $\rho_x$  and  $\phi_x(l)$ , the higher is the data item relevance. Finally, data items are classified in order of relevance (according to  $R(l)$ ). In the example given in Figure 3, the relevance value of the four data items that compose tuple 3 is displayed<sup>5</sup>. It can be noted that the data item located at row 3, column 4, noted  $T_{MD}\{3, 4\}$ , is the most relevant (relevance of 0.6) and shall be stored in priority on the item.

$$R(l) = \frac{\sum^x \rho_x \cdot \phi_x(l)}{\ln(\text{size}(l))} \quad \text{size}(l) > 1 \quad (1)$$

### 2.2. Process step 2: storage/retrieval of data on/from the “communicating material”

Process step 2 of the information dissemination process deals with the storage of data items on the “communicating material”. A RFID tag may store more or less information according to the available memory on the tag and thus, one data item may require more memory space than that available in one tag. The idea is therefore to split the set of data items among several tags that compose the “communicating material”. To do so, a specific application protocol, named *splitting protocol*, was developed in [12]. This protocol defines a *header* that is added in each RFID tag to know in which order the data items are split. This header requires, in each RFID tag, 16 bytes out of the  $n$  writable. Appendix A provides an example detailing the different *header* fields and, by the way, shows how the data item  $T_{MD}\{3, 4\}$  (*cf.* Figure 3) is split over a “communicating textile” (*cf.* Figure 9).

### 2.3. Need of data synchronyztation strategies

The “communicating material” paradigm enables to link information to specific parts of the material on a life-long basis. However, linking information to the item inevitably generates data replicas in the system that must

<sup>5</sup>The different steps to compute these relevances are not presented in this paper (see [13]).

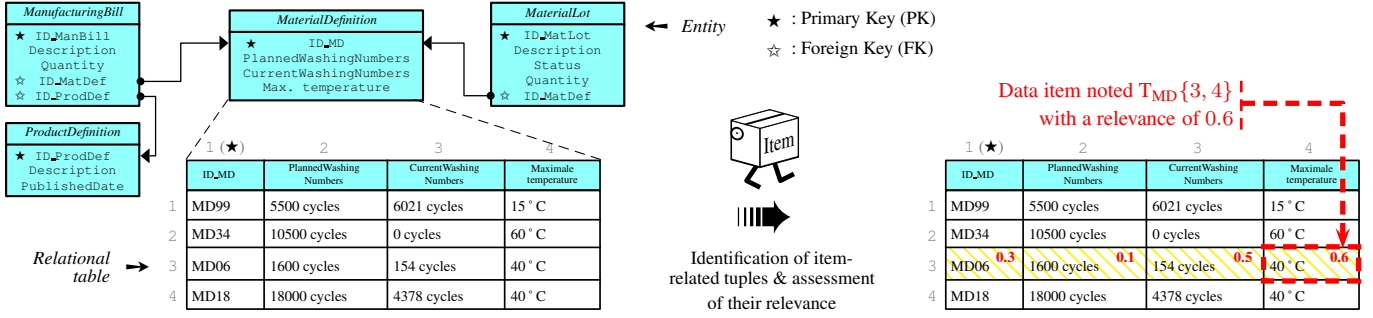


Figure 3: Illustration of a Logical Data Model (LDM) and the relevance computation of item-related information

be managed properly. For instance, if data carried by the item has to be updated without access to the database and later “synchronized” with it, some strategies/protocols are required. Such strategies are referred to as “data synchronization” [22]. Section 3 describes how it is possible to ensure a high degree of data synchronization using the “communicating material” paradigm.

### 3. Item data synchronization strategies in CL<sub>2</sub>M

#### 3.1. Related work

Ranasinghe et al. [23] have extensively studied three PLIM architectures *i)* EPC network architecture, *ii)* Dialog platform and *iii)* WWAI, providing a comprehensive comparison of their respective functionalities. The authors state that these three architectures have weaknesses in network disruptions and a relative lack of data synchronization. In other words, without network connectivity, databases cannot be reached to access or update important item-related information. In some situations, depending on the item-level intelligence [10], the item may be able to generate and gather information even during network disruptions (e.g. if the item owns computational power and storage capacities). This makes it possible to merge the item’s information into the database once connectivity is available. Nonetheless, this requires methods to synchronize the different information replicas that are stored both in the database and on the item. Suzuki and Harrison [22] defines four main degrees of data synchronization:

1. *always-on environment with synchronization, with replicated information on the item*: all item-related information is centralized in a database, which is always available. The item itself may contain a subset (or the entirety) copy of this information. The client can therefore access item-related information conveniently either on the database or on the item. The information can only be modified on the database, and is automatically updated on the item (if necessary). By definition, the data replication is performed in a synchronous mode with the database as primary copy,
2. *allows disconnected operation, with replicated information on the item*: similar to degree 1 but, this

time, the online database is not always available. This means that information is first modified on the database, and then updated on the item once connectivity is available. By definition, the data replication is performed in an asynchronous mode with the database as primary copy,

3. *allows disconnected operation, allowing updates from the item*: similar to degree 2, but updates can be performed on the item even when it is disconnected from the network. In contrast with previous degrees, the item-related information in the database is not allowed to be updated as long as the item is off the network. By definition, the data replication is performed in an asynchronous mode with the item as primary copy,
4. *allows disconnected operation, allowing updates from the item and from the database*: updates can be performed both on the item and on the database, even during disconnected phases. The synchronization mechanism is more complex since it requires specific policies (e.g. the resulting action may be to reject the update or to raise an alert if a conflict occurs between the data replicas from the item and the database). By definition, the data replication is performed in an asynchronous mode with possibility of update everywhere.

Ranasinghe et al. state that the current PLIM systems have difficulties for supporting data synchronization mechanisms, even for degrees 1 and 2. However, such synchronizations are unavoidable when an item has intermittent connectivity to the network resources. This is a particular problem for long-life items and for items moving in numerous locations. Added to that, the current PLIM systems do not provide strategies to select, at a given stage of the PLC, the information from the database that is relevant/useful to be stored on the item.

Accordingly, section 3.2 explains how it is possible to bring the strategies of data synchronization a step further when using the tools portfolio developed for the “communicating material” paradigm (see section 2).

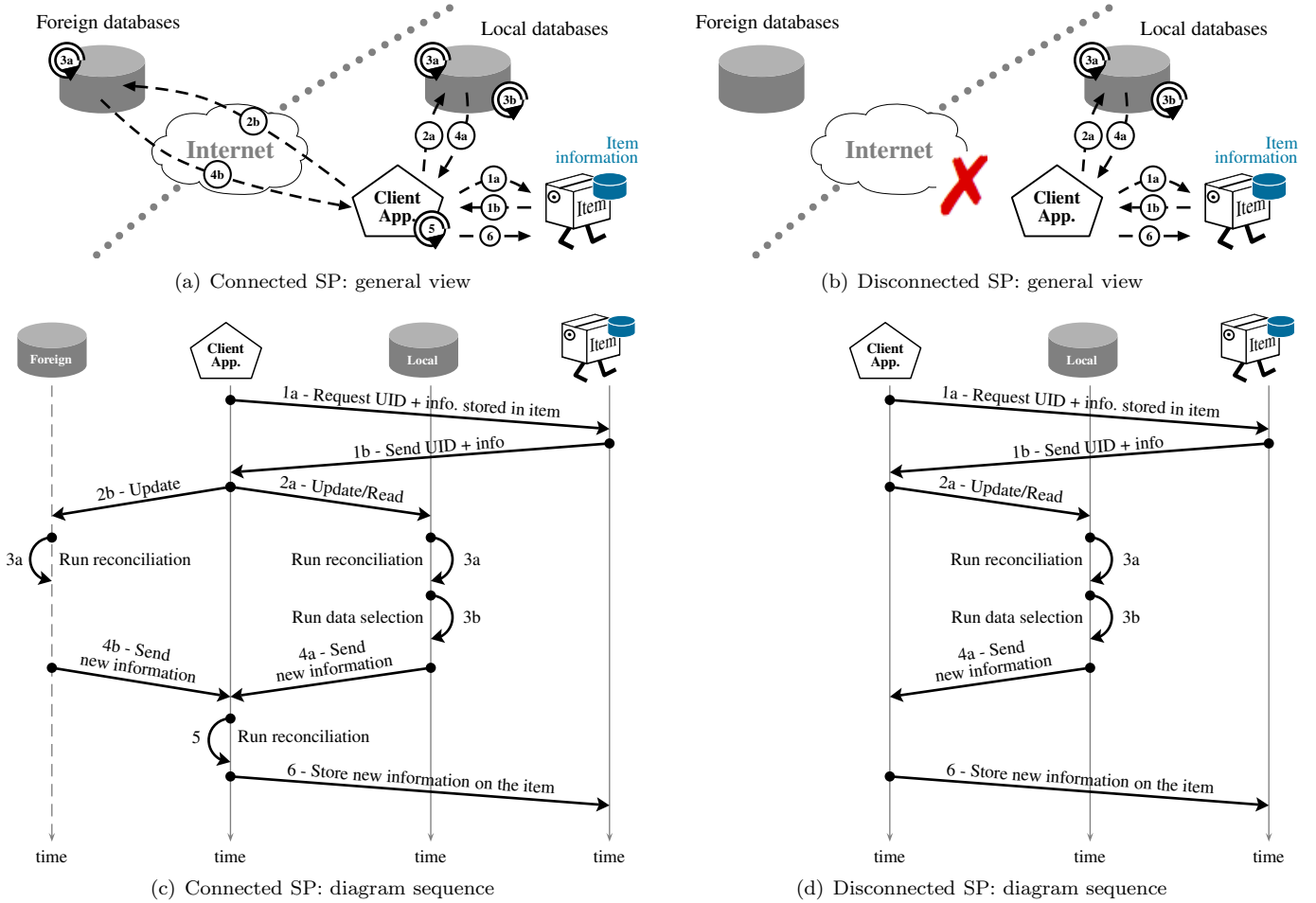


Figure 4: Description of the two types of synchronization points: Connected SP and Disconnected SP

### 3.2. Data synchronization of degree 4 using “communicating materials”

Even though futuristic, the paradigm of “communicating material” is a great opportunity to enhance current PLIM systems. However, to get the most out of the “communicating material” paradigm, data synchronization is one of the important techniques to be developed. In our vision, an item passes through several synchronization points (noted SP in this paper), which are moments when item-related information is written (or read) on (from) the item. Let us describe hereafter how it is possible to meet the requirements of a data synchronization of degree 4 using the “communicating material” paradigm. To do so, let us define the term “local databases” referring to databases owned by the actor who defines the SP, and the term foreign databases referring to databases owned by other actors in the PLC (e.g. supply chain pairs). Two situations may occur:

1. *connected SP*: the actor defining the SP has access to foreign databases as illustrated in Figure 4(a). The client application corresponds both to the hardware and software systems for reading (writing) informa-

tion from (on) the item. If the synchronization point or, more exactly, the client application performs:

- *a reading operation of the item*: information can be either retrieved on the item (see arrows 1a, 1b in Figure 4(a)/4(c)), or on local and foreign databases (see arrows 2a, 4a and 2b, 4b). Since a synchronization of degree 4 authorizes the modification of information contained in the database without necessarily advising the item and *vice versa*, problems of data consistency may occur when synchronizing information versions from both nodes. As a result, policies like reconciliation mechanisms should be defined<sup>6</sup> [24, 25] (arrows 3a in Figure 4(a)/4(c)),
- *a writing operation of the item*: a writing operation of the item requires the two-step information dissemination process introduced in section 2. The first process step aims, in this

<sup>6</sup>As example purposes, one type of reconciliation could be to keep the most recent data. This kind of mechanisms is not discussed in detail in this paper.

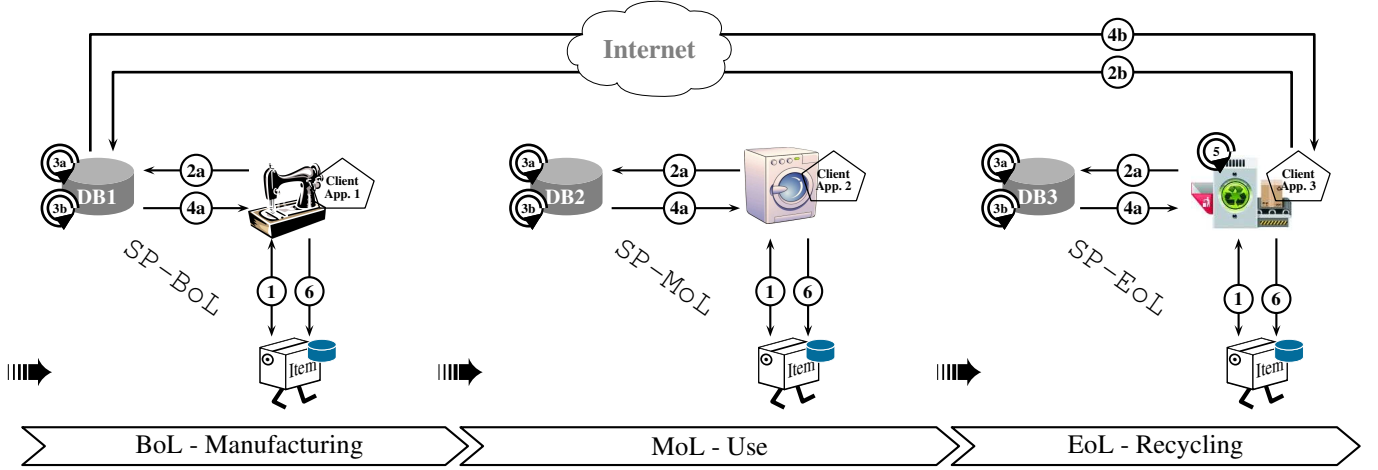


Figure 5: Illustration of the scenario related to the medical garments with three synchronization points

case, at identifying relevant information from local databases that should be stored on the item. This corresponds to the arrow 3b annotated “Run data selection” in Figure 4(a)/4(c). The second process-step (i.e. splitting of the selected information on the item made of “communicating material”) corresponds to the arrow 6,

2. *disconnected SP*: the actor defining the SP does not have access to foreign databases (e.g. matter of network availability or of access rights), as depicted in Figure 4(b). If the synchronization point performs:
  - *a reading operation of the item*: information can only be retrieved on the item (see arrows 1a, 1b in Figure 4(b)/4(d)),
  - *a writing operation of the item*: as in the previous case, the two-step information dissemination process is run to identify the relevant information to be stored and split on the item (respectively corresponds to arrow 3b and 6).

These explanations show how the use of items made of “communicating material”, together with the information dissemination process, could meet the requirements of a data synchronization of degree 4. The next section presents a scenario that focuses on the PLC of “communicating medical garments”.

#### 4. Applicative scenario

The initial item considered in this scenario is the “communicating textile reel” designed in [14] (see Figure 2). Three SP are defined by three different actors in the PLC in order to store and retrieve information on/from these garments, as depicted in Figure 5:

1. *a manufacturer (in BoL)*: the “communicating textile reel” is used for designing medical garments. At

this synchronization point, noted SP-BoL in Figure 5, the manufacturer wants to store on the medical garments useful information from his database (i.e. DB1) for the downstream actors. This information is related to the maximal number of washing cycles that the garment can undergo,

2. *a user (in MoL)*: the medical garments are then sent to a hospital and used on a daily basis. Hospital’s washing machines are equipped with RFID writers in order to read and update the information embedded on the garments [11]. This architecture is used in this scenario to update the current number of washing cycles that the garment undergoes. In this scenario, the washing machine is not allowed to access to foreign databases (i.e. DB1 and DB3) as illustrated in Figure 5. As a result, SP-MoL is a disconnected SP,
3. *a recycler (in EoL)*: the medical garments on which holes or unwashable stains appeared are finally sent to a recycler. During this operation, information that has been stored on the garments all along the PLC is retrieved, and then synchronized with both the local database (DB3) and the manufacturer’s database (DB1). This corresponds to the synchronization point noted SP-EoL in Figure 5, which is a connected SP.

Sections 4.1 to 4.3 describe how the information dissemination process is put into practice at each synchronization point, and detail the different information that is exchanged between the actors and the item.

##### 4.1. Retrieval and storage of item information in SP-BoL

Figure 6 illustrates the sequence diagram of the exchange of information between the different entities of the scenario (actors, databases, items; *cf.* horizontal axis) according to the PLC (SP-BoL, SP-MoL, SP-EoL; *cf.* vertical axis). First, let us focus on SP-BoL. The manufacturer (i.e. the Client App. 1 in Figure 6) retrieves both the UID

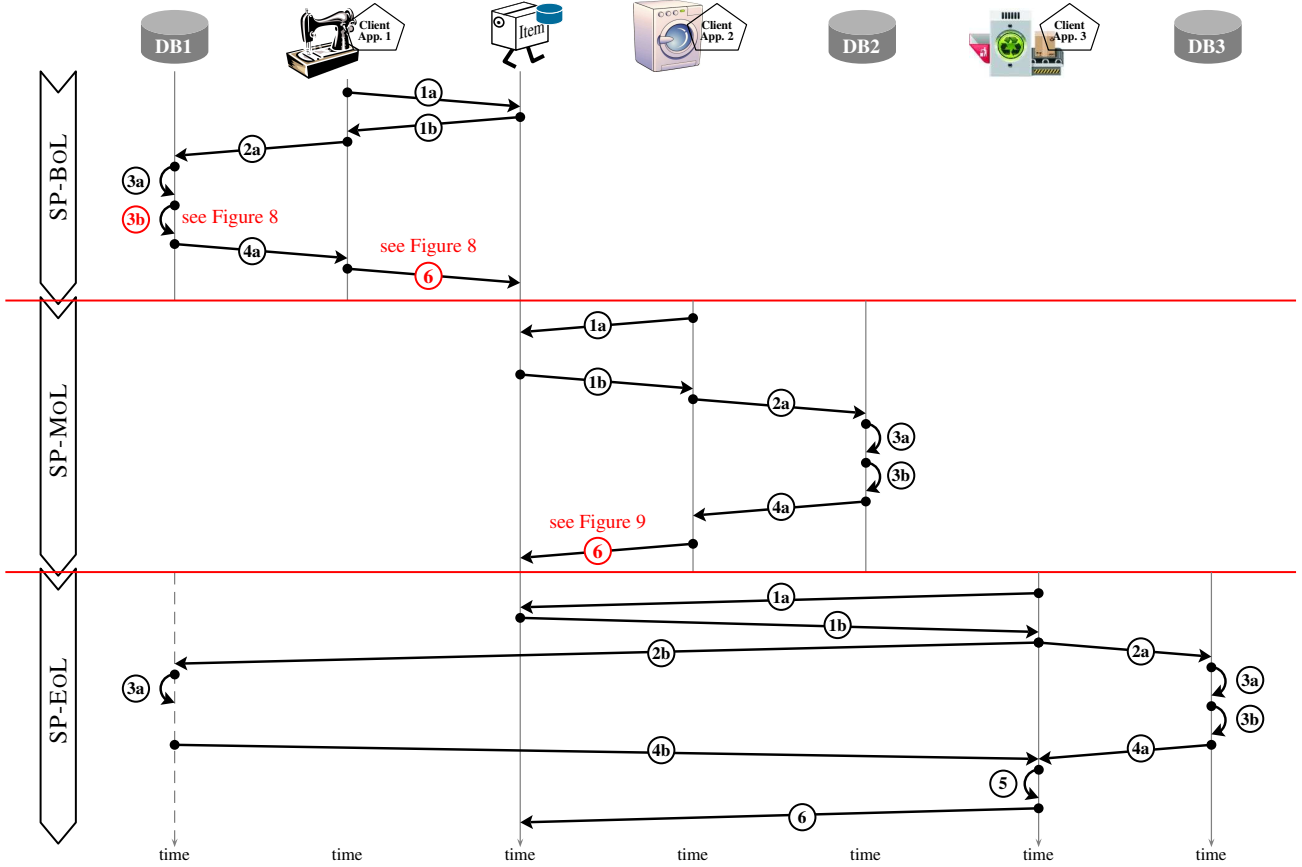


Figure 6: Sequence diagram of the exchanges of information between the different “entities” of the medical scenario

and the information from the item<sup>7</sup> (i.e. from the “communicating textile reel” so as to re-inject it in the local database DB1. This corresponds to the arrows 1a, 1b, 2a and 3a in Figure 6. The arrow 3a is in fact the run reconciliation that takes care of the synchronization of data versions between the item and DB1.

The “communicating textile reel” is then used to design medical garments as illustrated in Figure 7. Then, the manufacturer wants to store on each garment the information relevant to be conveyed to the downstream actors. This corresponds to the run data selection (i.e. the first step of the information dissemination process), represented by the arrow 3b in SP-BoL in Figure 6, which enables in this case the selection of relevant information from DB1. After having performed the data selection, three data items from `MaterialDefinition` are identified as relevant enough, namely  $T_{MD}\{3, 1\}$ ,  $T_{MD}\{3, 3\}$  and  $T_{MD}\{3, 4\}$  as emphasized in Figure 7.  $T_{MD}\{3, 1\}$  is the primary key that uses UIDs to identify each garment item,  $T_{MD}\{3, 3\}$  provides the current number of washings that is updated during the use phase by the washing machine, and  $T_{MD}\{3, 4\}$  provides the maximum washing temperature. At this synchronization point,  $T_{MD}\{3, 2\}$  (giving

<sup>7</sup>We make the assumption that no information has been stored on the “communicating textile” before SP-BoL, except for the UID.

the estimated maximum number of washing cycles) is not deemed as relevant to be stored on the garment since this information is primarily useful for the manufacturer who could know, at the end of the garment’s life, whether the estimated maximum number of washing cycles has been reached or not. These three data items are then split over the garment made of “communicating textile” via the *splitting protocol* as illustrated in Figure 7 (corresponding to arrow 6 in Figure 6). This demonstrates that products made of “communicating textile” are able to embed new information even after a physical transformation.

The medical garments are then sent to a hospital where a new synchronization point (SP-MoL) is defined and is used as many times as the garment is washed.

#### 4.2. Retrieval and storage of item information in SP-MoL

Let us focus on SP-MoL in Figure 6. When the medical garment is put into the washing machine, the set of information that the previous actor stored on it is retrieved as shown in Figure 8(a). This corresponds to the arrows 1a and 1b in SP-MoL in Figure 6. Let us note that a software application was developed in [12] to provide the user with the option of displaying this information, as shown in Figure 8(b). It can logically be observed in Figure 8(a) and 8(b) that  $T_{MD}\{3, 2\}$  is missing. Based on the retrieved data items (particularly on  $T_{MD}\{3, 4\}$ ), the washing ma-

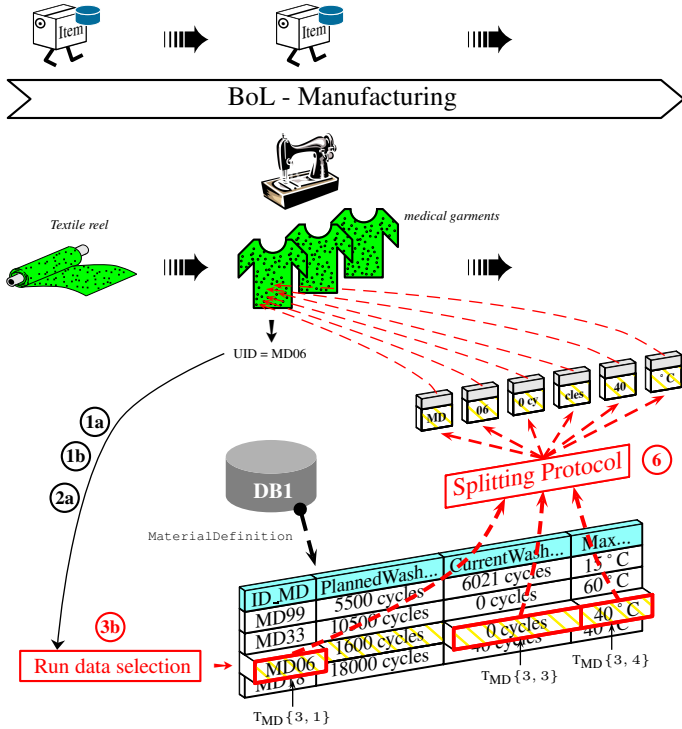


Figure 7: Data retrieval and storage (arrows 1a to 6 in SP-BoL)

chine automatically adjusts its washing program<sup>8</sup>. Once the washing is completed, the machine then increments the current value of the number of washing cycles (i.e. the current value of  $T_{MD}\{3,3\}$ ) in DB2 as illustrated in Figure 8(a), which corresponds to the arrows 2a and 3a in SP-MoL in Figure 6. By performing the run of data selection (arrow 3b), the same three data items are stored on the garment (arrows 4a and 6), but with the updated values (i.e. 800 cycles and not 799 anymore) as shown in Figure 8(a).

Since no connectivity with the other actors (i.e. the manufacturer and the recycler) is available at this moment, the storage of washing cycle information on the garment itself avoids breaking the item-information continuum in the PLC. After a time and when residual stains/holes are detected, medical garments are sent to recycling.

#### 4.3. Retrieval and storage of item information in SP-EoL

In this last synchronization point, an information feedback is established between the recycler and the manufacturer (see Figure 5), which helps the manufacturer to know whether the estimated maximum number of washing cycles (initially specified) has been reached. Information carried by the medical garment is first retrieved and rebuilt (arrows 1a, 1b in SP-EoL in Figure 6). Then, the information is updated in the local database DB3 (arrow 2a) and, in the same time, is sent back to the manufacturer database DB1 (arrow 2b). It is then necessary to

<sup>8</sup>This adjustment also depends on the information retrieved from the other garments involved in the same washing cycle.

use reconciliation mechanisms to conserve the right value (arrows 3a). For instance, in DB1 (BoL), the current number of washings is 0 and is then modified by 800 (i.e. the real number of washing cycles in MoL). This information feedback clearly results from the CL<sub>2</sub>M concept, but the memorization of the 800 cycles in MoL was made possible by the “communicating material” paradigm. By the way, we demonstrated in this scenario that a data synchronization of degree 4 can be reached with such a combination.

As mentioned, such information feedbacks allow for better assessment of the real PLC and may be used to provide better customer service while minimizing costs. For instance, in this scenario, the manufacturer could define specific metrics as the one specified in Table 1 (i.e. *Time-based metric*), which enables to assess the final quality of the garment and make decisions accordingly (see “Resulting actions” in Table 1). This scenario only focuses on the number of washing cycles, but other metrics could be defined for any other kind of information.

## 5. Conclusion

Concepts such as Internet of Things and Ubiquitous Computing redefine how we interact with information. It is not uncommon to use intelligent products to ensure an information continuum over the product life cycle - PLC (e.g. for traceability purposes). It is a formidable challenge of linking product-related information to the products themselves, thus making the information easily accessible. Over the last decades, research carried out in the framework of CL<sub>2</sub>M (closed-loop Lifecycle Management) attempted to achieve such a link. With CL<sub>2</sub>M, product information can be collected and used in a multi-organizational manner due to emerging technologies. The breakthrough challenge of this concept is to be able to continually improve design, manufacturing, use and end-of-life handling of products, thereby obtaining improved quality, less breakdowns and reduced need for spare parts. However, in most applications, information is deported through the network (stored in databases) and is accessed remotely.

This paper proposes to bring the CL<sub>2</sub>M concept a step further with the introduction of a new kind of material referred to as “communicating material”, which makes it capable of undergoing physical transformations without losing its communication ability and the data that is stored on it. Products made of “communicating material” are mainly used to convey, in a sustainable way, information between the different actors of the PLC.

Nonetheless, embedding information on such a material generates information replicas that must be handled by the system so as to maintain the data availability and consistency. To do so, some strategies/protocols exist, which are referred to as “data synchronization”. To get the most out of the “communicating material” paradigm, data synchronization is one of the important techniques to be developed. This paper introduces four main degrees of data synchronization and then describes how the use



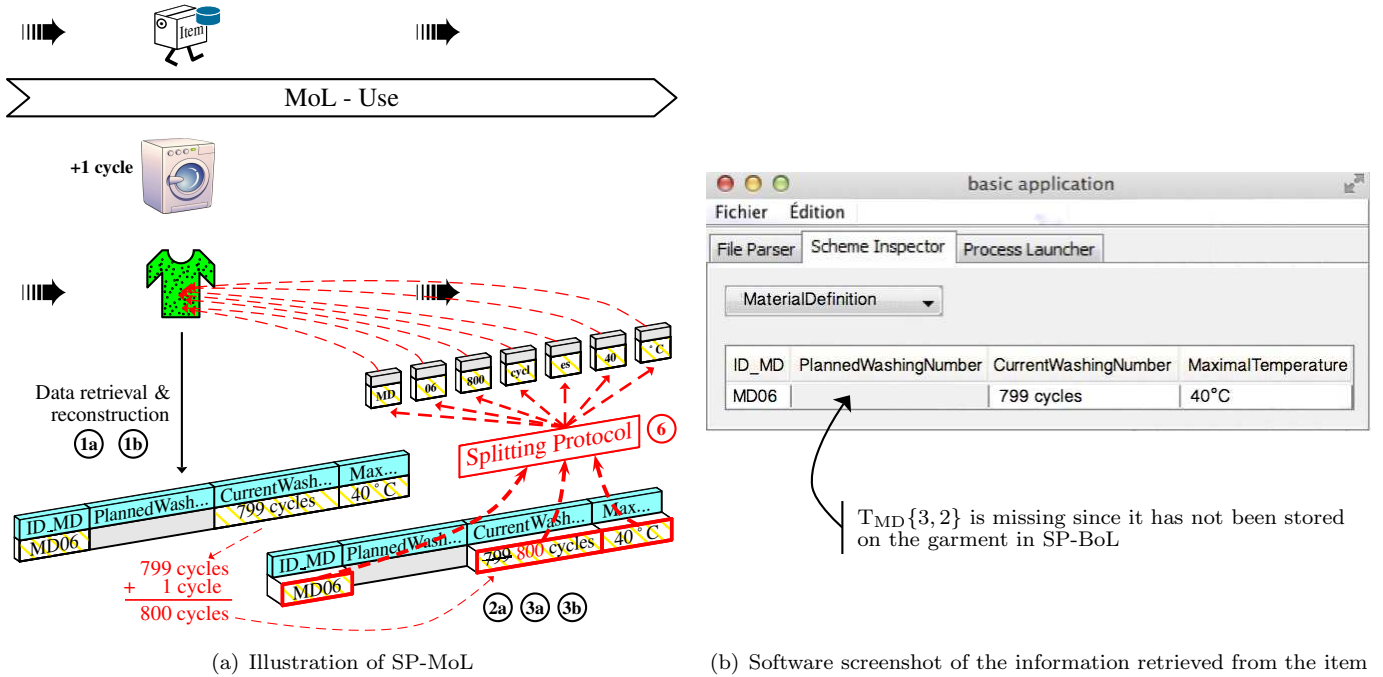


Figure 8: Data retrieval and storage (arrows 1a to 6 in SP-MoL)

Table 1: Situations that might occur when assessing the whole PLC of each medical garment

Final quality control	Time-based metric	Resulting actions
Reasonable quality	Estimated item's life $\simeq$ Real item's life	No design changes are required
Too good quality	Estimated item's life $\gg$ Real item's life	The design is too expensive and could be decreased somehow (material choice, <i>etc.</i> )
Too bad quality	Estimated item's life $\ll$ Real item's life	The design does not meet the user requirements and must be modified. Costs might increase

of the CL<sub>2</sub>M concept, together with the “communicating material” paradigm, makes it possible to achieve the highest degree of data synchronization. This degree offers the possibility to perform updates on both the item and the database, even when they are not connected to each other. A scenario in the framework of medical garment lifecycle helps to show that the requirements related to this synchronization degree are met, and the product-information continuum maintained. This has direct impacts on the actors of the PLC, who can obtain new information and knowledge about their products' life.

In the same line, further work shall investigate the feasibility of achieving the four degrees of data synchronization in existing standards that aims at creating standardized interfaces for collecting and exchanging product-related information. The Quantum Lifecycle Management (QLM) work group of the Open Group<sup>9</sup> aims to promote such interface standards, based on the results of the FP6 project PROMISE. The QLM Messaging Interface is one

proposal for such a standard but other proposals exist (see e.g., [26]) and will probably be proposed in the future.

## Acknowledgment

The authors gratefully acknowledge the financial support of the CPER 2007-2013 (Competitiveness Fibre Cluster), through regional (Région Lorraine), national (DRRT and FNADT) and European (FEDER) funds.

## References

- [1] Sudarsan R, and Fenves SJ, Sriram RD, Wang F. A product information modeling framework for product lifecycle management. *Computer-Aided Design* 2005; 37(13):1399–1411.
- [2] Ouertani MZ, Baina S, Gzara L, Morel, G. Traceability and management of dispersed product knowledge during design and manufacturing. *Computer-Aided Design* 2011; 43(5):546–562.
- [3] Chandrasegaran SK, Ramani K, Sriram RD, Horváth I, Bernard A, Harik RF, Gao W. The evolution, challenges, and future of knowledge representation in product design systems. *Computer-Aided Design* 2013; 45(2):204–228.
- [4] Stark J. *Product lifecycle management: 21st century paradigm for product realisation*. Springer-Verlag GmbH; 2011.

<sup>9</sup>www.opengroup.org/qlm/

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

[6] Kiritsis D. Closed-loop PLM for intelligent products in the era of the internet of things. *Computer-Aided Design* 2011; 43(5):479–501.

[7] CL2M. <http://promise.innovation.com/cl2m>

[8] Främling K, Harrison M, Brusey J, Petrow, J. Requirements on unique identifiers for managing product lifecycle information: comparison of alternative approaches. *International Journal of Computer Integrated Manufacturing* 2007; 20(7):715–726.

[9] Männistö T, Peltonen H, Martio A, Sulonen R. Modelling generic product structures in STEP. *Computer-Aided Design* 1998; 30(14):1111–1118.

[10] Meyer GG, Främling K, Holmström J. Intelligent products: A survey. *Computers in Industry* 2009; 60(3):137–148.

[11] McFarlane D, Giannikas V, Wong ACY, Harrison M. Product intelligence in industrial control: Theory and practice. *Annual Reviews in Control* 2013; xx(xx):xx–xx.

[12] Kubler S, Derigent W, Thomas A, Rondeau É. Storage of information on manufactured products using “communicating materials”. In: 1st Workshop on AI Problems and Approaches for Intelligent Environments. Montpellier (France); 2012.

[13] Kubler S, Derigent W, Thomas A, Rondeau É. Embedding data on “communicating materials” from context-sensitive information analysis. *Journal of Intelligent Manufacturing* 2013; 10.1007/s10845-013-0745-y.

[14] Kubler S, Derigent W, Thomas A, Rondeau É. Prototyping of a communicating textile. In: international conference on Industrial Engineering and Systems Management. Metz (France); 2011.

[15] Jover J, Thomas A, Leban JM, Canet D. Interest of new communicating material paradigm: An attempt in wood industry. *Journal of physics: conference series* 2013; 416(1):12–31.

[16] Özsü MT, Valduriez P. Distributed database systems: where are we now?. *Computer* 1991; 24(8):68–78.

[17] Baltrunas L, Ludwig B, Peer S, Ricci F. Context relevance assessment and exploitation in mobile recommender systems. *Personal and Ubiquitous Computing* 2012; 16(5):507–526.

[18] Chan D, Roddick JF. Context-sensitive mobile database summarisation. In: 26th Australasian computer science conference. Adelaide (Australia); 2003.

[19] Ono C, Takishima Y, Motomura Y, Asoh H. Context-aware preference model based on a study of difference between real and supposed situation data. In: 17th international conference on User Modeling, Adaptation, and Personalization. Trento (Italy); 2009.

[20] Khosroanjom D, Ahmadzade M, Niknafs A, Mavi RK. Using fuzzy AHP for evaluating the dimensions of data quality. *International Journal of Business Information Systems* 2011; 8(3):269–285.

[21] Rachuri S, Subrahmanian E, Bouras A, Fenves SJ, Foufou S, Sriram RD. Information sharing and exchange in the context of product lifecycle management: Role of standards. *Computer-Aided Design* 2008; 40(7):789–800.

[22] Suzuki S, Harrison M. Data synchronization specification. Aerospace-ID program report. In: Auto-ID Labs, University of Cambridge. Cambridge (United Kingdom); 2006.

[23] Ranasinghe DC, Harrison M, Främling K, McFarlane D. Enabling through life product-instance management: Solutions and challenges. *Journal of Network and Computer Applications* 2010; 34(3):1015–1031.

[24] Gray J, Helland P, O’Neil P, Shasha D. The dangers of replication and a solution. *ACM SIGMOD Record* 1996; 25(2):173–182.

[25] Petersen K, Spreitzer MJ, Terry DB, Theimer MM, Demers AJ. Flexible update propagation for weakly consistent replication. *ACM SIGOPS Operating Systems Review* 1997; 31(5):288–301.

[26] Främling K, Holmström J, Loukkola J, Nyman J, Kaustell A. Sustainable PLM through Intelligent Products. *Engineering Ap-*

plications of Artificial Intelligence 2013; 26(2):789–799.

## Appendix A. Splitting Protocol

The  $\mu$ -RFID tags that compose the “communicating textile” are Read/Write tags with a memory of 34 bytes. Regarding  $T_{MD}\{3,4\}$ , the character string that must be split is composed of both the *index* (to know in which table, attribute and tuple is located  $T_{MD}\{3,4\}$ )<sup>10</sup> and the content of the data item (i.e. the value “40 °C”). This character string therefore consists of 49 characters as emphasized in Figure 9, knowing that 1 ASCII character is encoded on 1 byte. These 49 bytes are added to the 16 bytes of the header. Accordingly, three tags are needed to split  $T_{MD}\{3,4\}$  as computed in equation A.1.

$$\left[ \frac{\text{index+content (bytes)}}{\text{tag memory (bytes) - header (bytes)}} = 3 \right] \quad (\text{A.1})$$

Figure 9 illustrates the splitting of  $T_{MD}\{3,4\}$  over three RFID tags  $a$ ,  $b$  and  $c$ . These tags have<sup>11</sup>:

- the same *Protocol* value (see tag  $c$  for the name of each header field),  $(255)_{10} = (FF)_{16}$ , because the same header structure is used (i.e. the one defined for the *splitting protocol*) that is identified by the value  $(255)_{10}$ ,
- the same *Size* value,  $(34)_{10} = (22)_{16}$ , because tags  $a$ ,  $b$  and  $c$  are of the same technology with a memory of 34 bytes (see tag  $a$ :  $32 \times 8.5 = 272 \text{ bits} = 34 \text{ bytes}$ ),
- a *Seq\_Num* value which provides the splitting sequence. The first part of the character string is stored in tag  $a$ ,  $Seq\_Num = (1)_{10} = (01)_{16}$ , the second one in tag  $b$ ,  $Seq\_Num = (2)_{10} = (02)_{16}$ , and the last one in tag  $c$ ,  $Seq\_Num = (3)_{10} = (03)_{16}$ ,
- a *Prev\_Num* value that provides the sequence number of the previous tag. For instance, *Prev\_Num* in tag  $b$  is equal to  $(01)_{16}$  (i.e. the *Seq\_Num* of tag  $a$ ). *Seq\_Num* and *Prev\_Num* of tag  $a$  are identical (i.e.  $(01)_{16}$ ) because tag  $a$  is the first tag to be written,
- the same *ID\_Phase* value because the 3 pieces of data resulting from the splitting of  $T_{MD}\{3,4\}$  have respectively been written at the same time in the PLC. The value  $(1344940753)_{10} = (00000000502a2ad1)_{16}$  represents the *timestamp* which translates into the date : 08/14/2012 at 13:05:52,

<sup>10</sup>The index is composed of the table name, the attribute name and the name of the primary key, and is encoded as follows: `tablename.attributename.PrimaryKeyname`

<sup>11</sup>A generic description of each header field is provided in [12].

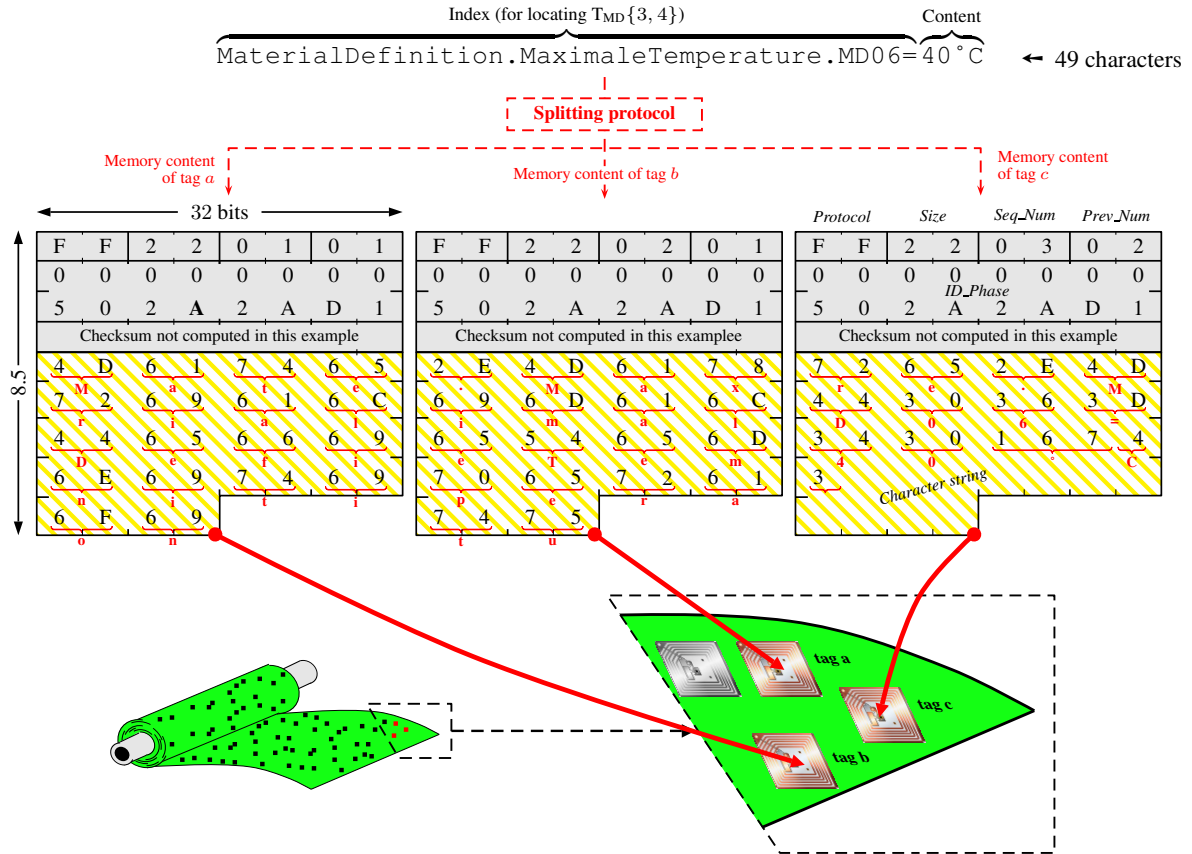


Figure 9: Illustration of how  $T_{MD}\{3,4\}$  is split over a “communicating textile” via the *splitting protocol*

- the checksum value (checksum calculations are not presented in this example),
- a part of the character string related to  $T_{MD}\{3,4\}$  (*index + content*). Each character composing this string is encoded in ASCII as shown in Figure 9 (see for instance tag *a*: letter  $M=(4D)_{16}$ , letter  $a=(61)_{16}$ ). It can be noted that the index requires the full memory of tags *a* and *b* and a piece of tag *c*, and the content of  $T_{MD}\{3,4\}$  is stored at the end of tag *c* (i.e.  $40\text{ °C}$ ).