

Information dissemination process for context-aware products

S. Kubler* W. Derigent* É. Rondeau* A. Thomas*

* Research Centre for Automatic Control of Nancy, Campus sciences,
BP-70239, F-54506 Vandœuvre-lès-Nancy Cedex, France
(e-mail: Firstname.Lastname@cran.uhp-nancy.fr)

Abstract: The amount of data output into our environment is increasing each day, and the development of new technologies constantly redefines how we interact with this information. It is therefore necessary to control the different ways information is diffused. To do so, a four-step data dissemination methodology in the framework of the Supply Chain Management (SCM) is introduced in this paper. On the one hand, the methodology aims to determine which information could be stored on the manufactured product (during its lifecycle) and, on the other hand, to assess what would be their relevance (according to the context).

Keywords: Distributed databases, Data dissemination, Internet of Things, Pervasive computing

1. INTRODUCTION

New challenges and opportunities arise with concepts such as Internet of Things (IoT), Ubiquitous/Pervasive Computing [Weiser (1991)]. Through these concepts, objects of the real world are linked with the virtual world. Thus, connections are not just people to people or people to computers, but people to things and most strikingly, things to things [Sundmaeker et al. (2010)]. Such applications rely on ever more complex information systems combined with ever increasing data volumes, which are stored in a large number of information vectors. These vectors may be either fixed (desktop computers) or mobile (wireless devices, RFID. . .). The IoT based on the RFID usage enables accessing data disseminated on any kind of physical object and developing new smart services [Yan et al. (2008)].

Although widely explored in the Computer Science field, IoT applications in the framework of SCM or Material Flow Management are limited. However, such a concept may turn out to be a good strategy. Indeed, during its lifecycle, a product passes through numerous companies dedicated to specific activities (manufacturing, usage. . .). Technical, semantic and organizational interoperability between these companies is not always ensured, thus, con-
ducing to information loss. If one considers the product as an information vector (i.e. on which information could be stored), it would contribute to improve interoperability all along the product lifecycle. According to Meyer et al. (2009), few research has been conducted on *intelligence at object*, i.e. products carrying their own information and intelligence (one can talk about active products). In fact, most of the time, products are only given an identifier (e.g. RFID) referring to a software agent or a database. This mode of information management is diametrically opposed to works initiated since 2003 by the PDS (Product-Driven Systems) community, which advocates physical information distribution within the product (i.e. information needed for managing its evolution). Meyer et al. concur

with the PDS community by stressing the fact that, in an increasingly interconnected world involving many actors, information should not be stored in a single database but should be distributed throughout the supply chain network. That said, substantial information distribution improves data accessibility/availability compared to centralized architectures, but they are more complicated to design because data consistency/integrity has to be maintained. In short, product information may be allocated both to fixed databases and/or to the product itself. One might then wonder what the optimal data distribution is.

To do so, we define a four-step data dissemination methodology, discussed in section 2. The section 3 focuses on one important step of this methodology which concerns the identification of relevant information that should be stored on the product. The identification method is then implemented on a case study in section 4. This work focuses on the product information management through its lifecycle. According to Meyer et al. (2009), our study considers the product as intelligent and takes a focus on the dimension related to the *location of intelligence* (i.e. at object and/or through the network).

2. PROCESS OF INFORMATION DISSEMINATION

This section provides an overview of the global process to disseminate information between fixed databases and active products. In our process, information is represented through a Logical Data Model¹ (LDM). The process consists of 4 steps as depicted in Fig. 1. This paper is in particular focused on step 2 since steps 1, 3 and 4 have been already dealt with in our previous works [Kubler et al. (2011a,b)]. We resume briefly these works (steps): Process step 1 consists in implementing the database system architecture, which can be either centralized or distributed. Many works in the literature help the designer to choose

¹ we assume that a common data model exists between the actors of the product lifecycle, which once derived gives rise to a LDM.

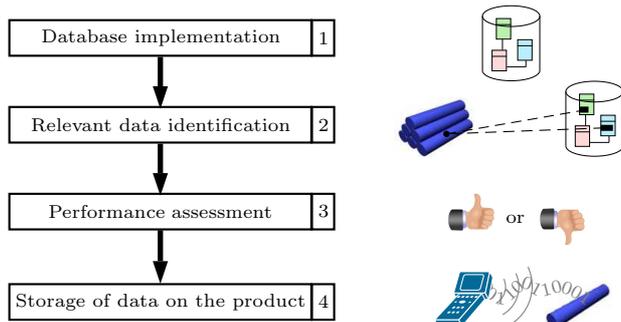


Fig. 1. Proposition of a data dissemination methodology

the more suitable system according to his expectations and the application constraints [Ozsu and Valduriez (1991)]. Step 3 deals with the system performance, which depends on multiple factors as the physical architecture, the data distribution, etc. Kubler et al. (2011a) propose an approach to evaluate each possible data fragment distribution between products and databases in order to identify the best one (regarding a given performance criterion which is the time for achieving a set of manufacturing operations). Once the data allocation is chosen, data fragments are allocated to databases and/or to products. This is performed in step 4. To store data fragments on the product, it is necessary to instrument it. Kubler et al. (2011b) propose a new concept referred to as *communicating material*, which considers the material as intrinsically communicating thanks to a large amount of RFID μ tags scattered into it (a textile prototype was designed). Based on a specific architecture, it is then possible to allocate (i.e. to read/write) data fragments. This last research work has shown that it is technically feasible to set up a communicating product containing its own data. Moreover, the physical performance assessment (step 3) could select adequate data distributions regarding specific performance criteria (e.g. delay), but these may not be adapted to the context. As a result, an additional step (step 2) is required to select relevant data according to the context (subject of the rest of the paper). Let us note that this framework, and especially the steps 2, 3 and 4 will be performed *on-line* (i.e. during a given activity of the product lifecycle: cf. Fig. 3), when the product needs to be written or read.

3. IDENTIFICATION METHOD OF RELEVANT DATA RELATIVE TO THE PRODUCT

During its lifecycle, a product is shared by many actors and undergoes various operations. It is therefore essential to identify information that is relevant to users, but it is not that easy. For this reason, we propose an identification method which is based on the Logical Data Model (LDM). Fig. 2 gives insight into a part of such a LDM (which is considered in the case study proposed in section 4). A given entity of LDM corresponds to a relational table, where the attributes listed in each entity correspond to the table columns and, each table row is referred to as a tuple/instance of the relation. A view of such tables is given in Fig. 5 for 3 entities among the 5 depicted in Fig. 2: *ManufacturingBill*, *MaterialDefinition* and *MaterialLot*. The identification method consists in two stages: (1) identification of all tuples through the entire LDM which are relative to the product instance (i.e. the product to

which we intend to attach relevant data such as production orders). The product instance, denoted i_p in this paper, may be of different types (e.g. product lot, single product) and consequently, will refer to distinct tables in the database; (2) assessment of what is their relevance to be stored on the product according to the user concerns, the product environment, etc. For this purpose, we propose to implement and to adapt the approach developed by Chan and Roddick (2003). This approach is interesting in the sense that they try to match the context with data in order to select *context-sensitive information*. Other procedures through the literature could be used to assess the data relevance. For instance, the *Analytic Hierarchy Process* methods which has the flexibility to be integrated with different techniques like *Fuzzy Logic*, etc. [Vaidya and Kumar (2006)]. It would therefore be interesting to compare such methods with the one proposed by Chan and Roddick. Initially, we opt for their approach because it is developed in the framework of distributed databases, based on criteria which suit to our concerns. Both stages of the identification method are detailed in the next sections.

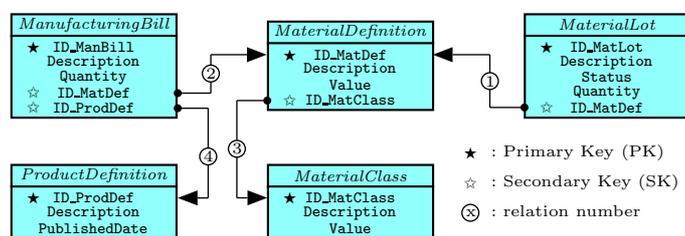


Fig. 2. Example of a Logical Data Model (LDM)

3.1 Adaptation of the method of Chan and Roddick (2003)

As stated above, the method for assessing the data relevance is based in part on the approach developed by Chan and Roddick (2003). Their approach uses the notion of priorities (or weights) to select the appropriate data for the expected situation. In their model these priorities are computed at the level of the data item². The priorities are numerical values either supplied or generated through observation and experimentation and are assigned through a multifaceted evaluation of different criteria (8 in total). In our approach, we implement 3 of the 8 criteria that we consider to be the most appropriate according to our context (i.e. SCM). For each criterion, the calculation of a relative priority, ρ_x (where x represents a criterion), and an assigned priority, Φ_x , are defined and combined in a single formula which gives a prioritization P_d for each data item. Data items will therefore be classified in order of relevancy (according to P_d). The first priority (relative) determines which criteria are more important and the second one (assigned) corresponds to the priority/weight value (between 0 and 1) computed by the criterion. The 3 criteria used in our approach are: (1) *Enumeration*, (2) *Contextual* and (3) *Model-based*. In the first one, an agent enumerates data that he considers as useful (i.e. data that he recommends to attach to the product). Φ_e is 0 or 1 depending on whether the data is enumerated or not (section 3.2 details what is an enumerated data). The second criterion (*Contextual*) can be used to include

² it refers to a table cell (i.e. the intersection of a given tuple/row and a given attribute/column of a relational table).

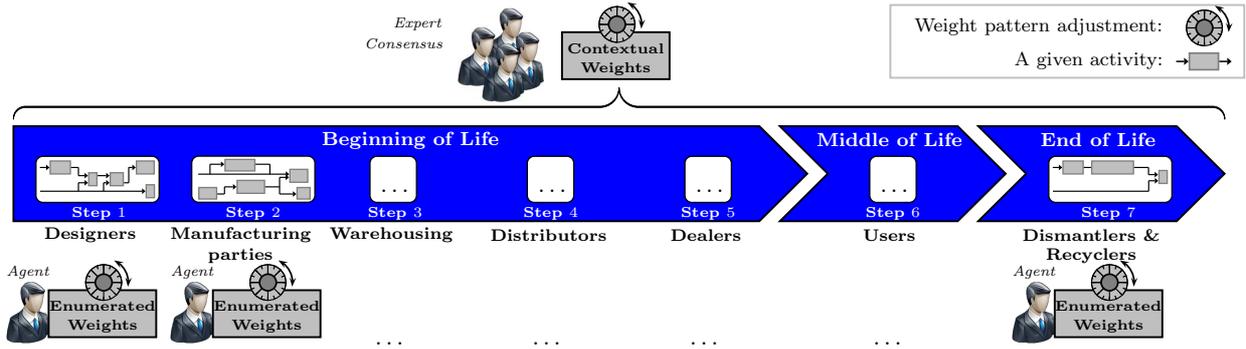


Fig. 3. The various weight adjustments throughout the product lifecycle

data related to the user's details or environment. Indeed, knowledge of the context of use can be useful in inferring the data that may be needed by users. The weight range Φ_c is detailed in section 3.2. The *Model-based* criterion is based on the relationships implied through the LDM. In fact, this criterion needs to compute distances between a reference table A and any other table B . Indeed, the shorter the distance between tables, the higher the data correlation. The distance corresponds to the shortest path to reach B from A (i.e. the number of relation which separate them). For instance, let **MaterialLot** be the reference table A and **MaterialClass** be the table B in Fig. 2. The distance between both tables equals 2 (relations ①-③). Note that the reference table is defined as that which includes i_p . This table is at the centre of our concerns and then, the weight Φ_m would decrease as the modeled distance increases. Chan and Roddick propose the formula: $\Phi_m = k^{-a}$ with $k \in [1; \infty]$ a constant, $a \in \mathbb{N}$ the distance. As mentioned previously, ρ_x and Φ_x (with x the given criterion) are combined in a single formula (equation 1), which gives a prioritization P_d for each data item, where len_d is the length (e.g. in byte) of the data item. The ρ value for each criterion (given the importance of this one) may be accommodated by the system designer.

$$P_d = \frac{\sum_x \rho_x \cdot \Phi_x}{\ln(len_d + 1)} \quad (1)$$

The next section details *when* and *how* to contextualize/enumerate information during the product lifecycle. Moreover, It is necessary to define a protocol to identify which tuples are relative to this product in order to assess only this information (subject of section 3.3).

3.2 Contextual and Enumerated adjustments

This section details *when* and *how* data are contextualized and enumerated during the product lifecycle. ISO (2003b) defines a project as: *a unique process, consisting of a set of coordinated and controlled activities with start and finish dates, undertaken to achieve an objective conforming to specific requirements*. Fig. 3 gives a global view of a product lifecycle consisting of several steps (Step 1 to 7) which in turn, are composed of organized activities. In our method, the contextual weight adjustment (for a given product) is done before the product begins its physical life, through an "Expert consensus". It consists in defining the importance of a group of entities (data) for each step. Indeed, specific entity groups may be identified through the LDM. In fact, information relative to the product is

quite often scattered within organizations: it is a matter of materials adopted, of applications used to manage technical data (e.g. Product Data Management systems - PDM), of applications that manage business and product information (e.g. Enterprise Resource Planning - ERP) and so on. Let us note that models ensuring the interoperability of these systems have emerged like IEC 62264, B2MML, ISA-88 [Baïna et al. (2008)]. Moreover, some works through the literature emphasized *when* information related to such systems has an interest regarding the product and the process (e.g. data from PDM is highly important at the beginning of life for both the product and the process, ERP all along the product lifecycle, etc.). Accordingly, the definition of the entity groups and the contextual weight adjustment could be performed based on such recommendations [Terzi et al. (2010)]. Once all entity groups are defined, the weight adjustment for each one may be performed. Fig. 4 gives insight into what could be the weight pattern for 4 distinct groups of entities regarding each lifecycle step. For instance, the "Expert consensus" strongly recommends to provide to the users, information related to *Entity group 1* at step 1 and 6. In opposition, data related to *Entity group 1* at step 4 will likely not be useful for the actors. The relevance levels defined through the terms *strong*, *high*, *medium* and *weak* correspond, in fact, to values between 0 and 1. Note that the weight related to a given group impacts all entities/attributes involved in that one. The following ones are proposed in an arbitrary way: $\Phi_c(\text{weak})=0.1$, $\Phi_c(\text{medium})=0.3$, $\Phi_c(\text{high})=0.6$ and $\Phi_c(\text{strong})=1$. Let us note that the current method does not allow to deal with several opinions and thus with disagreement among experts. As a result, works subsequent to this paper have been led to design a method for collecting a multitude of opinions and then, to fix the most suitable values. This method is based on a fuzzy-AHP approach.

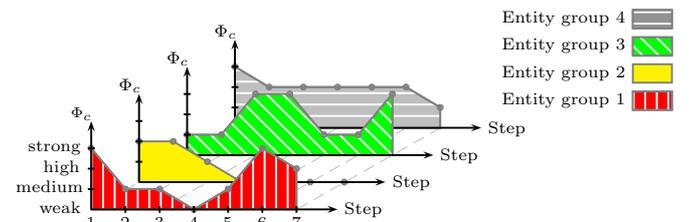


Fig. 4. Contextual Weight adjustment

The enumerated weight adjustment is achieved at each step by an agent expert in his own field. He knows better than anyone which information relative to his product

could be useful for the subsequent actors. However, he has generally a short knowledge regarding the entire product lifecycle and consequently, he could omit information relevant to other actors. As a result, the contextual criterion enables to moderate and to balance the enumerated one. Thus, if the agent moves in the direction of the contextual specifications, the data relevance (i.e. the prioritization P_d) will increase and, if not, will be moderated. Regarding the enumerated weight adjustment, the agent specifies the class attributes he expects to see stored on the product: Φ_e is 1 when an attribute is enumerated and 0 otherwise. With regard to the model-based criterion, nothing needs to be specified by a human since only the LDM structure is taken as input to compute Φ_m .

3.3 Protocol for identifying data relative to the product

The purpose before performing the assessment method (i.e. the prioritization computation via equation 1) is to retrieve all tuples/instances from all relational tables which are related to a given product. After that, computations will be exclusively applied on this set of data. In our approach, the idea is to extract when the time comes (i.e. when the framework is implemented in a given activity to write data on the product), all tables in a matrix format and then, to explore them thanks to an algorithm named **RetrievedData** (see Algorithm 1). It aims to retrieve all tuples from both the initial instance i_p and the set of tables \mathcal{T} . The program output provides the list Lt of all tuples (through the entire LDM) which are related to i_p . Due to a lack of space, the subfunctions **ExplorePK**, **ExploreSK**, **Neighbor** used in Algorithm 1 are not detailed, but we implement **RetrievedData** on a basic example.

Algorithm 1. RetrievedData(\mathcal{T}, i_p)

Variables: \mathcal{T} : the set of tables composing the LDM;
 Te : contains the tables already explored;
 I : contains data items which need to be explored (I_k : k^{th} element);
 I' : contains data items which need to be explored for a given table;
 Lt' : list of tuples (of a given table) related to i_p ;
 N : contains tables (not explored yet) neighboring of tables explored;
Output: Lt : list of tuples (through the entire LDM) related to i_p ;

```

1:  $I \leftarrow I \cup \{i_p\}$ 
2:  $Lt \leftarrow \emptyset$ 
3:  $Te \leftarrow \emptyset$ 
4: while  $I \neq \emptyset$  do
5:   while  $I \neq \emptyset$  do
6:      $[Lt', I'] \leftarrow \text{ExplorePK}(\mathcal{T}, I_1)$ 
7:      $Lt \leftarrow Lt \cup Lt'$ 
8:      $I \leftarrow I \cup I'$ 
9:      $Te \leftarrow Te \cup I_1$ 
10:     $I \leftarrow I - I_1$ 
11:   end while
12:    $N \leftarrow \text{Neighbor}(Te)$ 
13:    $I \leftarrow I \cup \text{ExploreSK}(\mathcal{T}, N)$ 
14: end while

```

The function named **ExplorePK**(\mathcal{T}, I_k) returns both the set of tuples Lt' related to the data item I_k (tuples belonging exclusively to the table containing I_k), and the list I' of data items which remain to be explored. Note that **ExplorePK** may not be sufficient to explore the entire LDM. Accordingly, a function named **ExploreSK** is introduced in order to avoid such a break. It returns new data

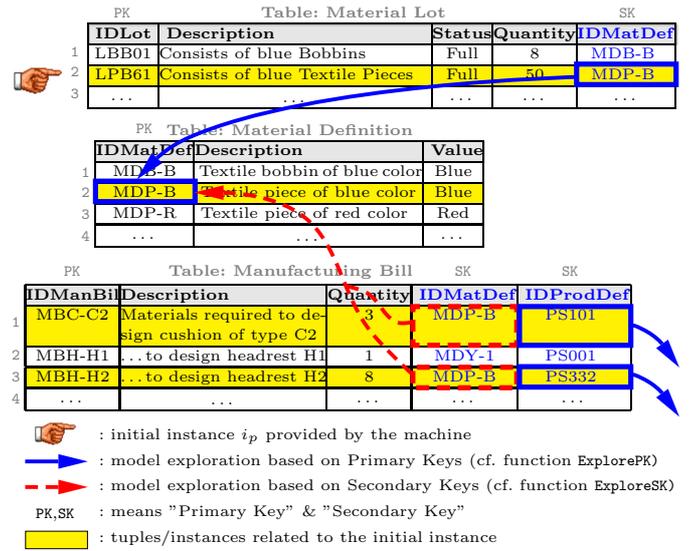


Fig. 5. Tuples retrieving via RetrievedData

items which will be used again as input into **ExplorePK**. **ExploreSK**(\mathcal{T}, N) requires, as input, the set of tables N not explored yet, which are directly linked to tables already explored (Te). Te is provided by **Neighbor**(Te). To better understand Algorithm 1, consider 3 tables in Fig. 5: **MaterialLot**, **MaterialDefinition**, **ManufacturingBill** (respectively noted **MatLot**, **MatDef**, **ManBil** in the algorithm steps given below). **RetrievedData**(\mathcal{T}, i_p) is then implemented where i_p is defined as the data item LBP61 which refers to the primary key (PK) of **MatLot** (noted {**MatLot:row2**} in the algorithm steps). A tuple is noted by the row number (e.g. {**MatLot:row2**}) and an algorithm step (line x) refers to the algorithm line number:

In line 1: $I = \{\text{MatLot:row2, coll}\}$;
In line 7: $Lt' = \{\text{MatLot:row2}\}$, $I' = \{\text{MatDef:row2, coll}\}$;
 First, the entire tuple related to i_p is returned (i.e. **MatLot:row2**). Secondly, the function checks if there are secondary keys (SK) in the current table. If so (there is **IDMatDef**), the function select in tables related to SK (i.e. the table **MatDef**) the data item for which PK is equal to that one (i.e. **MatDef:row2, coll** which has for value **MDP-B**).
In line 8: $Lt = \{\text{MatLot:row2}\}$;
In line 9-10: basic operations (memory effect);
In line 11: $I = \{\text{MatDef:row2, coll}\}$;
In line 7: $Lt' = \{\text{MatDef:row2}\}$, $I' = \emptyset$;
In line 8: $Lt = \{\text{MatLot:row2; MatDef:row2}\}$;
In line 11: $I = \emptyset$;
In line 13: $N = \{\text{ManBil}\}$;
 ManBil is a neighbor of **MatDef** and has not been explored yet.
In line 14: $I = \{\text{ManBil:row1, coll1; ManBil:row3, coll1}\}$;
 At this step, the algorithm is blocked since we cannot explore any other table. As a result, **ExploreSK** aims at returning new data items (if possible). It checks first in tables belonging to N if there are SK. If so (2 SK are identified in **ManBil**), it matches the identified SK with the PK of tuples belonging to Lt in order to determine if there are similar keys. If so (it is the case with **MDP-B**), the data item (PK) relative to the new tuples (i.e. **MBC-C2** and **MBH-H2**) are returned.
In line 7: $Lt' = \{\text{ManBil:row3}\}$, $I' = \emptyset$; and so on

In resume, the method to determine relevant product data consists at each step of the product lifecycle, in identifying all tuples which are somehow related to the product (via Algorithm 1) and then, in computing the data relevance (via equation 1). For that, two weight patterns need to be

specified by human experts (enumeration/contextual) and one (Model-based) is based on the LDM structure.

4. CASE STUDY

4.1 Scenario definition and specification

In our case study, a specific LDM is implemented which is based in part on the B2MML standard³ ISO (2003a). The model used in this paper consists of 19 entities (extracted from B2MML) which are clustered into 4 groups, namely:

- **Product data group:** MaterialLot, MaterialDefinition, MaterialClass, ManufacturingBill, ActualMaterialLot, MaterialSegmentSpecification,
- **Personal data group:** Person, ActualPersonSegment, PersonClass, PersonSegmentSpecification,
- **Production data group:** ProductionOrder, SegmentRequirement, SegmentResponse, ProductSegment, ProductDefinition,
- **Equipment data group:** Equipment, EquipmentClass, ActualEquipmentSegment, EquipmentSegmentSpecification,

The *Equipment* and *Personal data groups* report information about equipments and persons which/who are somehow related to the product (e.g. equipments used for manufacturing it). The *Product* and *Production data groups* relate respectively information about the product composition (e.g. raw materials) and operations (e.g. production rules). Fig. 5 illustrates 3 of the 19 entities, which belong to *Product data group*.

Let us assume that the "Expert consensus" has defined the contextual weight pattern as shown in Fig. 4, where "Entity group 1, 2, 3, 4" correspond respectively to the *Product*, *Personal*, *Production* and *Equipment data groups*. Moreover, the agent in step 7 enumerates the table attributes listed in Table 1. These patterns are defined for any lot instance of blue textile pieces. The LDM is not shown in this paper but the distance values will be specified when needed for calculations.

Table 1. Table attributes enumerated in step 7

Table Name	Enumerated attributes
MaterialLot	{IDLot, Quantity}
MaterialDefinition	{IDMatDef, Value}
ProductSegment	{IDProdSeg, Duration, UnitDurat.}
ProductionOrder	{IDProdOrder, StartTime, EndTime}

Henceforth, the identification method can be applied. Assume now that the product is being treated in a given activity of step 7 (see Fig. 3) and that the machine performing the operation needs to write information on that one. For that, the machine gets the product lot instance $i_p = \text{LPB61}$ and runs the identification method: first, all tuples (Lt) are retrieved via Algorithm 1, as explained in section 3.3 (cf. Fig. 5). The weight patterns of each criterion is now matched with this set of tuples. Fig. 6 depicts for *MaterialDefinition* the matching result "data/pattern" for the 3 criteria. All tuples without any relation with i_p are set to 0. Regarding the enumeration matching, the attributes *IDMatDef* and *Value* are set to 1 because they are

³ B2MML (Business To Manufacturing Markup Language) is meant to be a common data format to link business enterprise applications (e.g. ERP) with manufacturing enterprise applications (e.g. MES).

enumerated (cf. Table 1), in opposition to *Description*. Regarding the contextual matching, all attributes are set to 0.8 since the "Expert consensus" has specified a *high* level at Step 7 for the group 1 (i.e. *Product data group*), which contains the table *MaterialDefinition*. Regarding now the last matching (Model-based), the weight is equal to k^{-1} because the distance between *MaterialDefinition* and *MaterialLot*⁴ equals 1 (see Fig. 2). Calculations are then performed for each data item of each table via equation 1 (see the example $P_{2,1}$ in Fig. 6). In our case study, k is fixed to 2 and the criterion importance ρ_e, ρ_c, ρ_m respectively to 100, 75, 50 (as defined by Chan et al.).

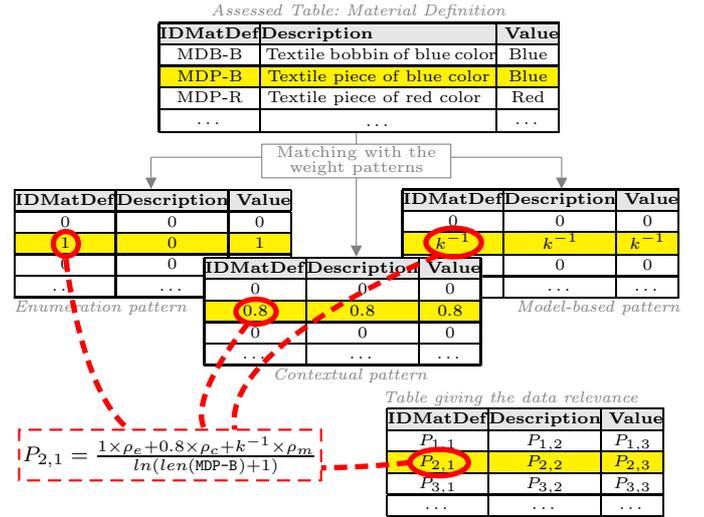


Fig. 6. Matching between tuples and weight patterns

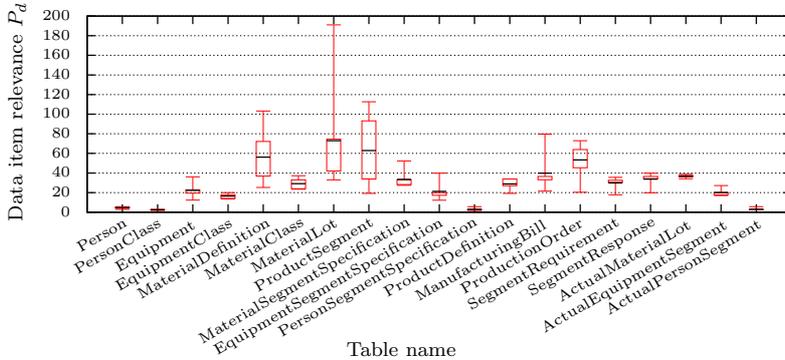
4.2 Results about data relevance

In this section, we comment the results of the data item relevance obtained after calculations. The term *list* is used in what follows to refer to all data items having a prioritization $P_d > 0$. Fig. 7(a) provides the resulting *list* ordered from the highest P_d to the smallest. The most relevant data item is the one located at row 2, column 4 in *MaterialLot* (cf. Fig. 5). Due to the large amount of data items included in the *list* (around 230), we present the results in the form of diagrams (pie chart and wisker diagram). First, let us look at the wisker diagram Fig. 7(b). For each table is given the min, the 1st/3th quartile, the mean and the max P_d of the set of data items included in the *list* and belonging to the table. We can see that the highest prioritizations (in average) are relative to *MaterialLot* (which includes the 1st data item of the *list*: $P_d = 191.15$), but also to *MaterialDefinition*, *ProductSegment*, *ProductionOrder*. This is due to the fact that some attributes of these tables are enumerated (cf. Table 1) in addition to a *high* and *strong* recommendation by the "Expert consensus" regarding respectively the *Product* and *Production data group* (which contain these four tables). Let us focus now on the pie chart Fig. 7(c) which is another representation of results. The equation Per_{gr} provided in Fig. 7(c) allows to compute the percentage relative to an entity group gr , where $len(list)$ is the number of data items into the *list* and P_{d_k} is the

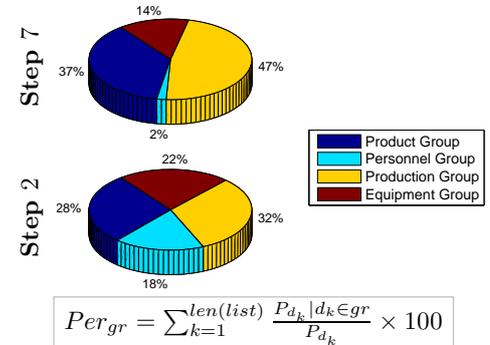
⁴ *MaterialLot* is the reference table because $i_p \in \text{MaterialLot}$

Rang	Table Name	Attribute Name	Primary Key Value	Cell Value	Relevance (P_d)
1 st	MaterialLot	Quantity (col4)	LPB61 (row2)	50	191.1502
2 th	ManufacturingBill	Quantity (col3)	MBH-H2 (row3)	8	112.6170
3 th	ManufacturingBill	Quantity (col3)	MBC-C2 (row1)	3	112.6170
...
224 th	ActualPersonSegment	Unit of Measure (col3)	MBH-H2 (row3)	8	1.7441
225 th	PersonClass	Description	PCPLW01	Worker Line...	0.9313

(a) List of data items ordered from the highest P_d to the smallest



(b) Result of Step 7 detailed for each table in a wisker diagram



(c) Result of Step 2 and 7 in form of a pie chart

$$Per_{gr} = \sum_{k=1}^{\text{len}(list)} \frac{P_{d_k} | d_k \in gr}{P_{d_k}} \times 100$$

Fig. 7. Results of the data item relevance

prioritization value of the data item k . A high group percentage might mean two things: (1) P_d values relative to data items belonging to the group are relatively high compared to the rest of the *list*; (2) a large majority of data items belong to the group. The pie chart (Step 7) clearly reflects the previous explanations, where the *product & production data groups* are more relevant (37% and 47%) than the other due to the enumeration choices and the "Expert consensus" decisions. In opposition, *Personal data group* is not relevant (with 2%) since there is no enumeration and no high recommendation from the "Expert consensus" ($\Phi_c = weak$). The identification method is now applied by taking into account the contextual specifications of Step 2 (cf. Fig. 4), with the same enumeration pattern (cf. Table 1). The *Personal data group* is now relatively important compared to Step 7. Indeed, there is still no enumeration but now, the "Expert consensus" recommends highly to provide information relative to this group. In resume, if the agent moves in the direction of the contextual specifications, the data relevance (P_d) will significantly increase and, if not, will be balanced. Finally, it can be concluded, **regardless** the physical system performance (i.e. the step 3), that data items at the top of the *list* have a high probability of being stored on the product in opposition to the ones at the bottom (see Fig. 7(a)).

5. CONCLUSION

Concepts such as Internet of Things, Ubiquitous Computing redefine how we interact with the information. It is therefore necessary to control the different ways information is diffused. To do so, a four-step data dissemination methodology in the framework of the Supply Chain Management is introduced in this paper, and one of these steps is then detailed. It aims at determining which information could be stored on the manufactured product and, to assess what would be their relevance (according to the context). This is achieved via an identification method which is applied on a case study. Current works are lead in order to achieve the four-step integration.

REFERENCES

- Baïna, S., Panetto, H., and Benali, K. (2008). Product oriented modelling and interoperability issues. *Enterprise Information Systems*, 293–308.
- Chan, D. and Roddick, J. (2003). Context-sensitive mobile database summarisation. In *26th Australasian computer science conference*, volume 16, 139–149.
- ISO (2003a). Enterprise - control systems integration. IEC/ISO 62264-1.
- ISO (2003b). Quality management systems - guidelines for quality management in projects. ISO 10006:2003.
- Kubler, S., Derigent, W., Thomas, A., and Rondeau, E. (2011a). Key factors for information dissemination on communicating products and fixed databases. In *Service Orientation in Holonic and Multi Agent Manufacturing Control*.
- Kubler, S., Derigent, W., Thomas, A., and Rondeau, E. (2011b). Prototyping of a communicating textile. In *International Conference on Industrial Engineering and Systems Management*, 1333–1342.
- Meyer, G., Främling, K., and Holmström, J. (2009). Intelligent products: A survey. *Computers in Industry*, 60(3), 137–148.
- Ozsu, M. and Valduriez, P. (1991). Principles of Distributed Database Systems. *Prentice Hall*.
- Sundmaeker, H., Guillemin, P., Friess, P., and Woelfflé, S. (2010). Vision & challenges for realising the Internet of Things. *Cluster of European Research Projects on IoT*.
- Terzi, S., Bouras, A., Dutta, D., and Galletti, M. (2010). Product lifecycle management—from its history to its new role. *International Journal of Product Lifecycle Management*, 4(4), 360–389.
- Vaidya, O. and Kumar, S. (2006). Analytic hierarchy process: An overview of applications. *European Journal of operational research*, 169(1), 1–29.
- Weiser, M. (1991). The computer for the 21st century. *Scientific American*, 265(3), 94–104.
- Yan, L., Zhang, Y., Yang, L., and Ning, H. (2008). The Internet of Things: From RFID to the Next-Generation Pervasive Networked Systems. *Auerbach Publications*.