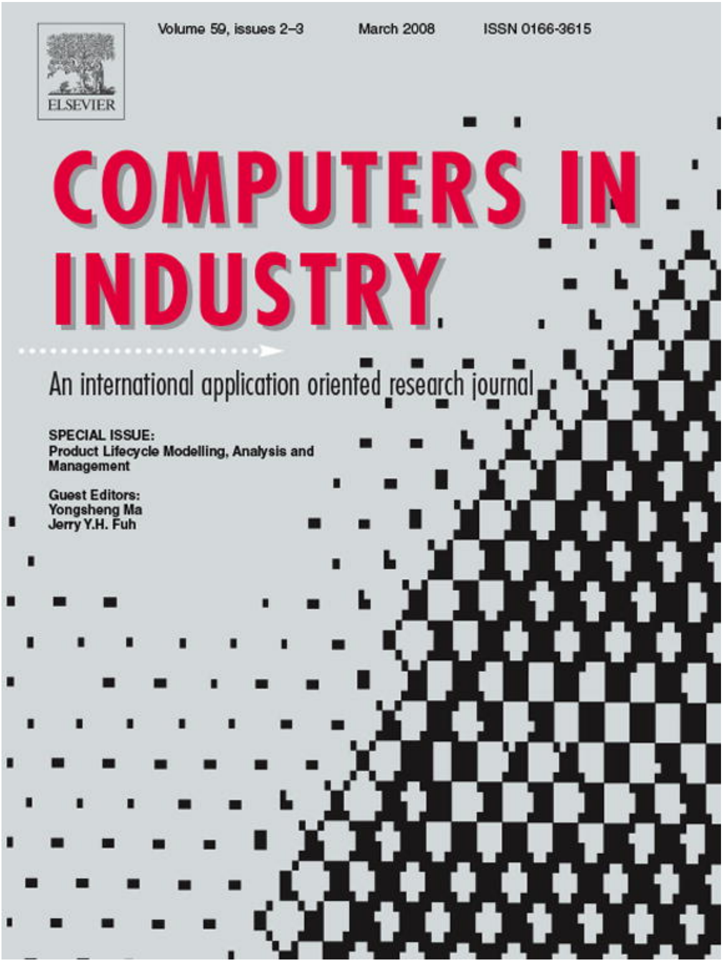
Volume 59, issues 2–3
March 2008
ISSN 0166-3615
ELSEVIER
COMPUTERS IN INDUSTRY
An international application oriented research journal
SPECIAL ISSUE:
Product Lifecycle Modelling, Analysis and Management
Guest Editors:
Yongsheng Ma
Jerry Y.H. Fuh

Available online at www.sciencedirect.com
ScienceDirect
ELSEVIER
Computers in Industry 59 (2008) 231–241
COMPUTERS IN INDUSTRY
www.elsevier.com/locate/compind

PRoduct ONTOlogy: Defining product-related concepts for logistics planning activities

Diego M. Giménez [a], Marcela Vegetti [b,c], Horacio P. Leone [b,c], Gabriela P. Henning [a,*]

[a] INTEC (UNL-CONICET), Güemes 3450, Santa Fe S3000GLN, Argentina
[b] INGAR (UTN-CONICET), Avellaneda 3657, Santa Fe S3002GJC, Argentina
[c] CIDISI (UTN), Lavaise 610, Santa Fe S3004EWB, Argentina

Available online 17 August 2007

Abstract

Current Internet-based technologies enable the operation of extended supply chains (ESCs) and introduce new requirements on managing and sharing product-related information in such ESCs, where product models are the fundamental information source. This paper describes an extension of the product data framework originally introduced by PRoduct ONTOlogy (PRONTO). The extended model provides the foundations for a distributed product data management (DPDM) system and is fully consistent with the idea of managing product information according to two hierarchies: the abstraction hierarchy (AS) and the structural hierarchy (SH). They formalize the data aggregation and disaggregation processes required by logistics planning activities. In this work, the *Property* and *PropertyValue* concepts were incorporated into the ontology to handle different types of data.

Keywords: Product model; Structural hierarchy; Abstraction hierarchy; Logistics planning activities

1. Introduction

The advances in information technology and the Internet boom provide a great opportunity for a new era of supply chain (SC) integration [1]. Within an ESC, the manufacturing logistics (referring to all planning, coordination and support functions required to carry out manufacturing and associated logistic activities) demands accurate and reliable information of different granularity levels about products in order to be efficient and to optimize the logistics-manufacturing interface [2]. Traditionally, product information is spread among several intra-organizational systems, especially ERP, PDM, and, more recently, Product Lifecycle Management (PLM) systems, with many possibilities of data redundancies and inconsistencies. Moreover, these systems do not provide support for handling the non-direct relationships that exist among product data defined at different granularity levels. These levels are usually linked to the temporal horizon of the associated decision problems.

To overcome some of these difficulties common product models to be shared within an organization have been introduced. However, the centralized approach is usually not feasible in the ESCs context due to the lack of support for inter-organizational business integration. One major obstacle is the low degree of automation in the exchange and integration of product-related data among business partners, mainly due to the use of different representations.

Product life cycles are shortening while, at the same time, new products must be delivered to market quicker than before. This leads companies to form ESCs in which information concerning common products must flow rapidly, faultlessly, and automatically so that they can compete in global markets. This requires product data to be transferred between companies in electronic form, with a high level of common representation [3].

Clearly, product data is created by and used in different organizational areas by all the ESC participants. Nevertheless, these areas are often characterized by heterogeneous environments in which product data may be represented in different ways. Without a standard definition of the product-related data, the semantic integration among the many logistics activities involved in the ESC is impossible.

* Corresponding author. Tel.: +54 342 4559175; fax: +54 342 4550944.
E-mail address: ghenning@intec.unl.edu.ar (G.P. Henning).

0166-3615/$ – see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.compind.2007.06.019

To face the challenges of a more demanding global competition, Web-based PDM systems arose [4]. Although these systems are technically and syntactically integrated, they do not allow a common understanding yet. The reason is that ''integration'' implies technical, syntactical and semantic integration. Internet and Web technology support the first two aspects while the last may be solved through the definition of ontologies.

An ontology is an explicit and formal specification of a shared conceptualization and provides a conceptual framework for communicating in a given application domain [5]. Consequently, ontologies for product data models provide a framework for sharing a precise meaning of symbols exchanged during communication among the many stakeholders involved in the ESC and allow the definition of agile and flexible DPDM systems. Thus, for example, Yoo and Kim [6] have presented a Web-based knowledge management system for facilitating seamless sharing of product data among application systems in virtual enterprises. Current research activities in this area are oriented towards the use of ontologies as a foundation for the ''Semantic Web'' [7].

Recently, Vegetti et al. [8] have made a contribution, proposing an ontology called PRoduct ONTOlogy (PRONTO). This work extends PRONTO with new concepts related to the specification of mechanisms for aggregating and disaggregating different kinds of product-related data needed for ESC manufacturing logistics, as well as representing such data along the product abstraction hierarchy. Section 2 justifies the need for extended product data models, Section 3 introduces the model itself and illustrates it by means of a few examples. Finally, Section 4 presents the most important conclusions.

2. Need for an extended model

Due to computational limitations and forecasting uncertainties, aggregated information (e.g. about product lines, product families, aggregate units of production) rather than detailed product information is used at the planning level. For instance, aggregate planning develops tactical plans for total sales, total production, targeted inventory and customer backlog for substitute or fictitious products representing the aggregate information of a set of similar items. In contrast, coarse information is disaggregated (when it is necessary) to feed data for solving material or distribution requirements planning problems (e.g. when aggregate plans should be converted into detailed master schedules).

This requires increased capability to organize data and knowledge at different abstraction levels during the whole product life cycle. It is possible to propose at least two product-related concept hierarchies to manage the product information complexities. One of them, referred as the *structural hierarchy* (SH), organizes the knowledge related to product structural information. The SH is a tool to manage the information associated with the multiple recipes and/or processes available to manufacture a given product or a group of similar products. The material requirements planning (MRP) system is a classical example of an application managing data along the SH. Within this hierarchy a typical information handled is the Bill Of Material (BOM) representation, which specifies the subordinate components (as well as their required quantities) that are physically needed to make each final product or assembly.

The other hierarchy, referred as *abstraction hierarchy* (AH), organizes products according to different levels of specification (from substitute products to individual items). It is oriented towards managing the complexity originated by the huge number of products that are nowadays manufactured in industrial facilities. The AH also employs knowledge structures and mechanisms for keeping consistency among the product-related data at different abstraction levels. Many examples taken from the specialized literature reveal how ''forward'' and ''backward'' links (associated with aggregation and disaggregation tasks) along the AH should be employed to coordinate those planning functions that are executed at different time horizons. For example, at strategic and tactical levels the forecast of an aggregate demand is more appropriate (since it is more accurate) than the forecast of an individual item, allowing a bullwhip effect reduction in the inventory and back-orders levels [9]. Similarly, average costs (production, inventory holding and backordering) and production rates are required by top-level production planning decisions while detailed data is used at the shop floor [10].

Many contributions published in the last 10 years have proposed knowledge representations for product SHs (based on different BOM models) and AHs (based on the construction of product families) [11–17]. Nevertheless, both hierarchies exhibit almost no integration. Moreover, these approaches only support the handling of structural information and do not provide a knowledge framework to manage other types of information through the AH, such as data associated with production and logistic costs, demand, inventory policies and levels, labour requirements, lead-times, product dimensions (logistic cube, weight), storage and transportation needs (temperature, humidity conditions), etc.

3. Extended product data model

3.1. Handling of product abstractions

PRONTO [8] formalizes a product knowledge representation that integrates the SH and AH hierarchies, mainly focusing on the treatment of product structural information. An object-oriented (OO) representation was used to model the concepts included in the ontology as well as the existing relations among them. OO technology was initially adopted by many authors [18,19] to propose structural product models. Recently, its use has been extended towards the modeling, specification and implementation of PDM and PLM systems [20–22].

Specifically, the ontology resorts to three abstraction levels for representing product-related concepts: *Family*, *VariantSet* and *Product*. They define an AH that allows to manage today's sudden increase in the number of products, which gives rise to multiple variants or alternatives.

As it is shown in the UML class diagram [23] of Fig. 1, each group of alike products receives the name of *Family*. This

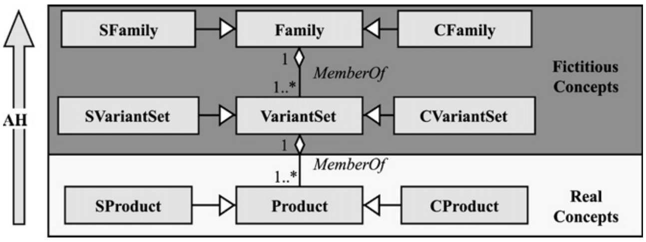

Fig. 1. Abstraction hierarchy adopted in PRONTO.

concept can include simple products (*SFamily*) or compound ones (*CFamily*). In the first case, a simple product is an atomic raw material or an acquired component that is not composed of other components or it is not decomposable into other products. On the other hand, a compound product is one that results from the assembly or processing of components or one that can be decomposed into other products (non-atomic raw material). Thus, a *Family* may group raw materials, acquired components, intermediate products (as assemblies) or final products.

In turn, a subset of members of a given family, having similar characteristics (structure and logistics features), is classified under the concept of *VariantSet*, which also can be simple (*SVariantSet*) or compound (*CVariantSet*).

The previous concepts refer to fictitious products. In other words, they correspond to products that do not have physical existence but that can be interpreted as substitute products having average characteristics.

Finally, the specific products are represented by the *Product* concept, modeling the most detailed level of the AH. This real concept can represent a simple or compound product (*SProduct* or *CProduct*) in aggreement with the previous classification.

The abstraction levels are related among themselves by means of *MemberOf* associations, indicating that a set of product entities included at a lower level are always members of a product abstraction instance in an upper level.

With the purpose of exemplifying the expressiveness of the proposed ontology, a small case-study of a candy industry is addressed. The best way to interpret the extended product model is by instantiating the generic concepts for the specific case-study. In the following UML diagrams, the stereotype *IO Class* appearing above a given object name will denote that such object is an instance of the *Class* referenced in the label. For example, in Fig. 2(a), *WrappedCandy* is an instance of the *CFamily* class, named <<IO *CFamily*>>.

To illustrate the proposed concepts, Fig. 2(a) shows the AH corresponding to the wrapped candy family. This family includes two variant sets: wrapped fruit candy and wrapped milk candy. In turn, each variant set has two members: wrapped strawberry and orange candies for the first case, and wrapped nut and coffee candies for the second one. Besides, Fig. 2(b) and (c) show the AHs associated with the two families of components required in the elaboration of wrapped candies: unwrapped candy and wrapping paper, respectively.

3.2. Handling of product structures

To represent structural characteristics, the model also incorporates a SH that allows to manage composition and decomposition structures. Traditionally, the modeling of data

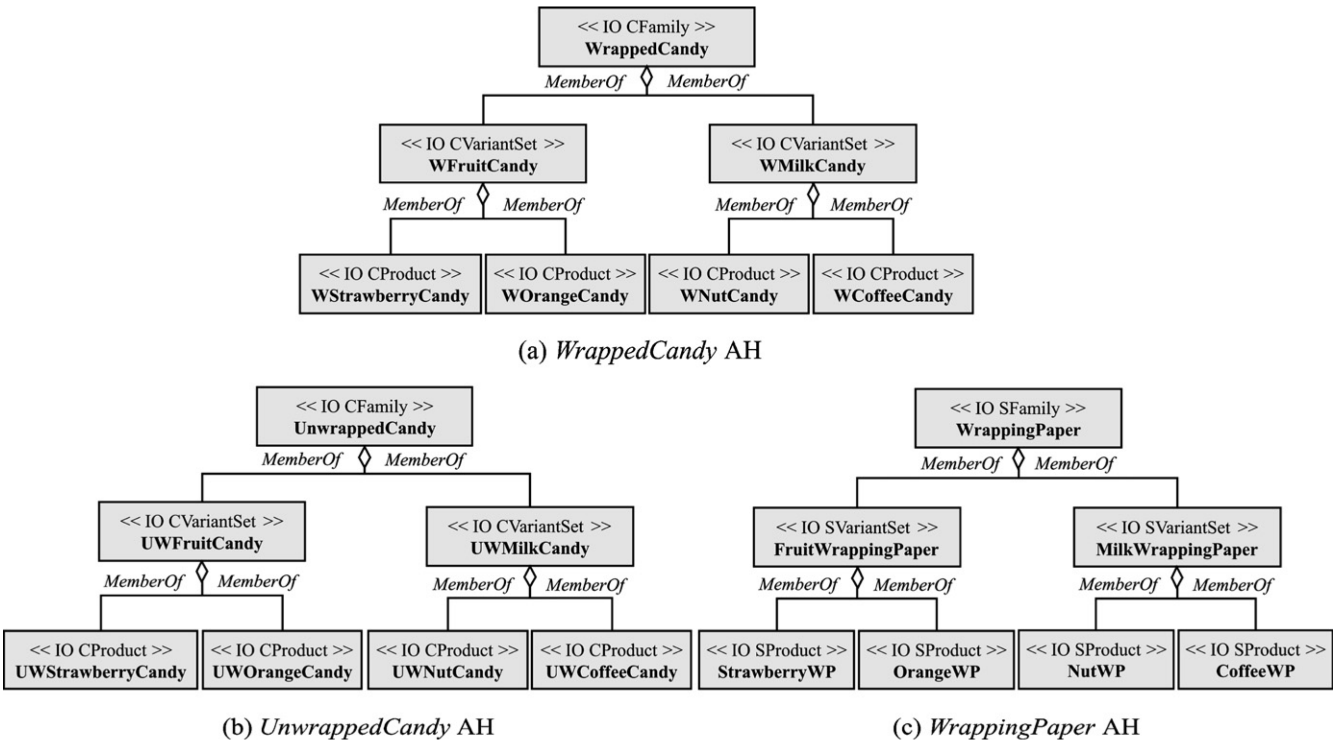

Fig. 2. Different AHs associated with the case-study.

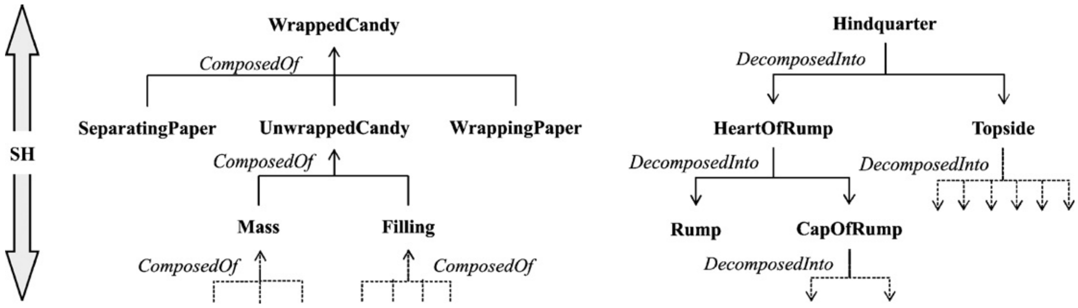

Fig. 3. Examples of composition and decomposition BOMs.

concerning product structures has been done through the BOM representation [24]. However, conventional BOM processing systems do not efficiently support the maintenance of the huge amounts of data demanded by production planning activities. Furthermore, they do not handle decomposition-based production strategies where products are obtained by decomposing raw materials, like in some food industries (milk and meat ones) and in the petrochemical business, where hybrid structures (combining composition and decomposition operations) may be associated with end products. In these industries, the raw materials are non-atomic, that is, a succession of decomposition operations is needed to obtain intermediate products that are later transformed into final ones. Fig. 3 presents typical composition and decomposition structures to be managed by the SH. The multi-level BOM representation on the left side is associated with the case-study that was taken to describe the product model and the representation on the right corresponds to a meat industry product.

The proposed SH adheres to the generative BOM philosophy [25], where a specific BOM is derived from a common product structure. As depicted in Fig. 4, the *Family* concept can be associated or not with a *Structure* (or a set of them) depending on whether it is a compound family or a simple one. In the first case, the structures define different ways of combining component parts and raw materials (*CStructure*) or decomposing non-atomic raw materials (*DStructure*) to make the products included in the family by means of composition or decomposition relations (*CRelation* and *DRelation*) established with the components' families.

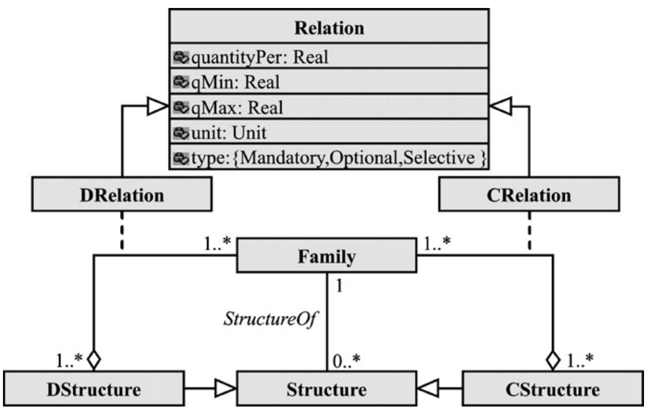

Fig. 4. Structural hierarchy adopted in PRONTO.

Each relation contains at least information about: (i) the quantity/number of units of a given component required to manufacture a unit of aggregate product (composition relation) or the quantity/number of units of intermediate products obtained from the decomposition of a unit of a non-atomic raw material (decomposition relation); (ii) minimum and maximum allowed quantities and the quantities' unit of measure; and (iii) the relation type. Three kinds of relations can be identified according to the exigencies for incorporating a certain component in a particular product structure: (i) *Mandatory*, when the component must be present in all the BOMs as part of the structure shared by the family members; (ii) *Optional*, when the component may be present or not in a specific BOM; and (iii) *Selective*, when a given number of components of those associated with the structure by means of selective relations must be included in the particular BOMs.

Fig. 5 illustrates the unwrapping candy family SH extracted from the case-study. As it is depicted, the composition structure of this family is always composed of mass, to which it is possible to add filling.

As mentioned above, in recent years, many industries have been forced to drastically increase their product variety and adopt mass customization production strategies. This phenomenon, called "product flexibility" [25], has turned into a new complexity factor for the product design, production processes, control systems and, consequently, for the information systems supporting them. This fact motivates research focused on the BOM representation of a great number of variants. To address this problem, all real products having a similar structure are grouped within the compound variant set concept. In such a way just one structure is stored for all the variant set members, which is derived from one of the structures associated with the family.

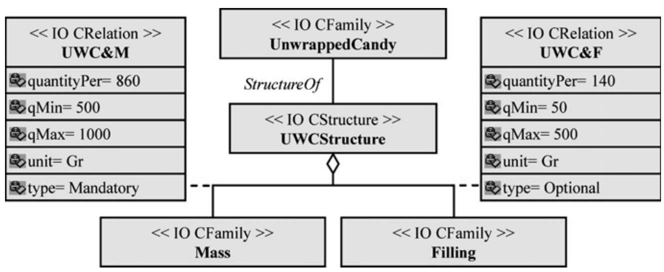

Fig. 5. *UnwrappedCandy* structural hierarchy.

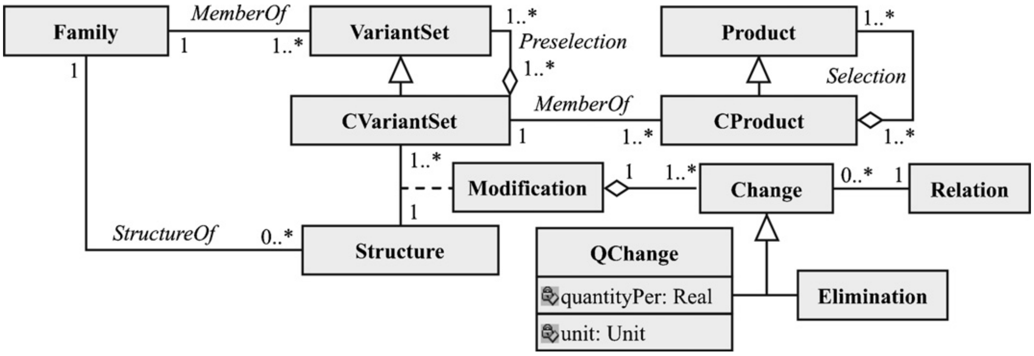

Fig. 6. *VariantSet* structure management.

In Fig. 6, it is shown that each compound variant set specifies the variant sets (simple and compound) considered (*Preselection*) into the structure from which it is derived, as well as the rules (*Change*) to adapt (*Modification*) such structure to the one that the variant set members specifically share.

Particularly, by means of the association of one specific variant set for each component of the structure, the real BOMs (corresponding to the members of a given compound variant set) are restricted to be composed of one member of each variant set previously associated. In turn, the allowed structural changes are: (i) component elimination (*Elimination* subclass), and (ii) change into the composition/decomposition quantity (*QChange* subclass). The first type of change is only permitted for those components linked to the structure by optional relations or selective ones. On the other hand, the second change requires the specification of a new *quantityPer* value and the unit of measure in which it is expressed. This value cannot violate the allowed minimum and maximum quantities. Quantity restrictions reduce those mistakes that may happen during composition/decomposition data entry into the PDM system.

Fig. 7 illustrates the particular structure of the *WFruitCandy* variant set. It can be seen that this substitute product (1000 Gr) is composed of *UnwrappedCandy* (970 Gr) and *WrappingPaper* (30 Gr) but does not contain a *SeparatingPaper*.

Finally, Fig. 8 shows that the single-level BOM of a given compound product (*CProduct*) is entirely defined by choosing (*Selection*) the specific simple and compound products to be combined during its production; just one member of each variant set included in the respective structure is chosen. In this example, the *WStrawberryCandy* product is composed of the real components *UWStrawberryCandy* and *StrawberryWP*. The composition quantities are already specified into the structure of the *WFruitCandy* variant set, as it was described in Fig. 7.

Continuing with the definition of single-level BOMs related to compound components, multi-level BOMs, such as the one depicted in Fig. 9 for *WStrawberryCandy*, can be obtained.

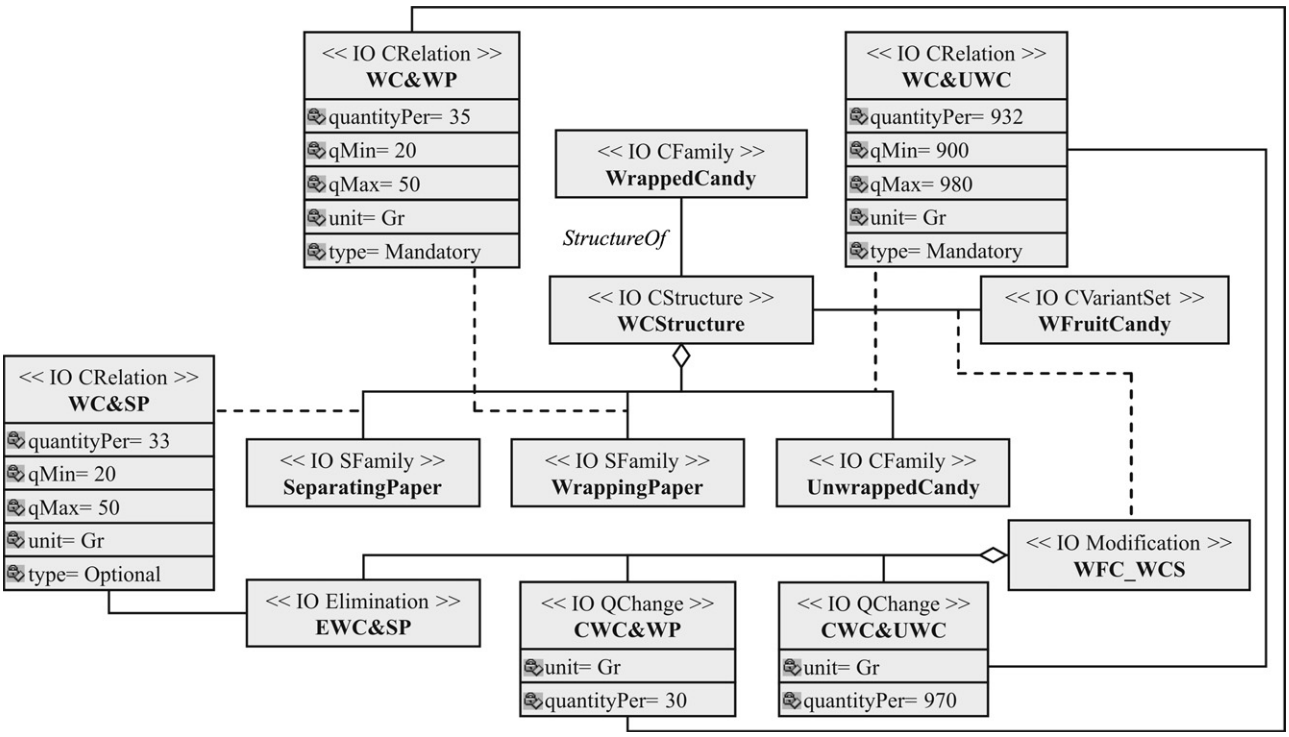

Fig. 7. Example of the structure definition of a specific variant set.

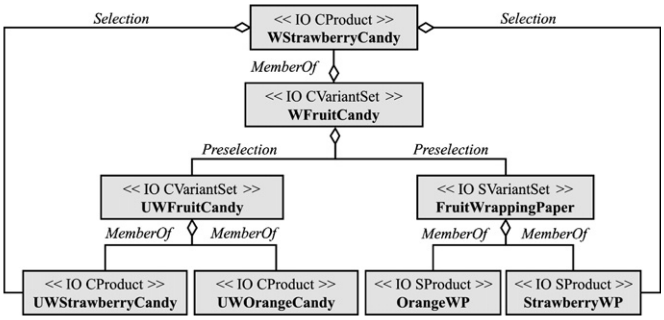

Fig. 8. BOM definition of a given product.

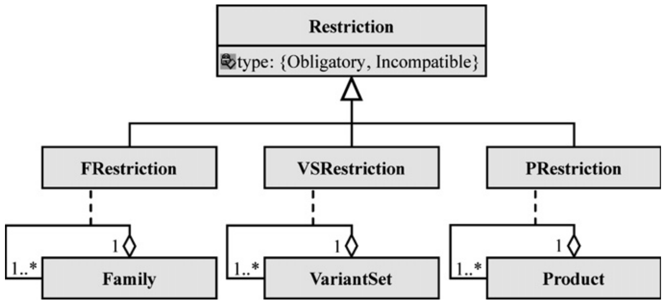

Fig. 10. Restrictions represented in the model.

An important aspect is that companies vary the product portfolio along the time. So, many obsolete structures relative to real products would be kept in the PDM systems, even when they are not included in production plans or their manufacture has been discontinued (with a dramatic increase of physical storage requirements). Under these circumstances, this compact model, that stores common aspects once and just keeps track of the distinctive features at the *VariantSet* and *Product* levels, seems to be very efficient in terms of data storage needs. Moreover, the uncontrolled storage of BOMs may also originate logical data consistency problems if changes in the data associated with a certain component are not simultaneously made in all the BOMs in which it participates. The proposed model allows to carry out changes in the common structures without having to deal with the affected BOMs individually.

Additionally, it is necessary to consider that for a given structure not all the components combinations are valid due to technological or commercial reasons. Therefore, the model includes a mechanism for expressing restrictions that actuate at the moment of defining generic structures or real products' BOMs. These constraints complement those established by the composition/decomposition relation types. Fig. 10 depicts the different kinds of restrictions among components that can be applied at each abstraction level. They are classified in agreement with the abstraction level in which they occur: *FRestriction*, *VSRestriction* and *PRestriction*. In turn, it is possible to identify two main types of restrictions among components: (i) *Obligatory* and (ii) *Incompatible*; both must be defined to obtain valid structures. The first type forces a given set of components (families, variant sets or products) to take part in the structure (at any level) in which certain product concept (which establishes the *Obligatory* restrictions with the above mentioned components) has been included.

In contrast, *Incompatible* restrictions impose a given set of components (families, variant sets or products) to be not present in the structure (at any level) in which certain product concept (which establishes the restriction relations with the above mentioned components) has been included.

In Fig. 11, a few restrictions that appear in the case-study are presented. Its analysis reveals that, at the *Family* level, a *WrappingPaper* must always be included in those structures in which a *SeparatingPaper* participates as a component. However, a restriction of the same type would not be valid in the opposite direction. In turn, at the *VariantSet* level, an *UnwrappedMilkCandy* must never be accompanied by a *FruitWrappingPaper*. Finally, at the *Product* level, an *UnwrappedStrawberryCandy* must always be accompanied by an *StrawberryWrappingPaper*.

To summarize, a partial view of PRONTO, including the concepts defined to support the structural information handling, is shown in Fig. 12.

3.3. Property and property value concepts

In order to enlarge the semantics associated with the AH, the *Property* concept has been incorporated into the model

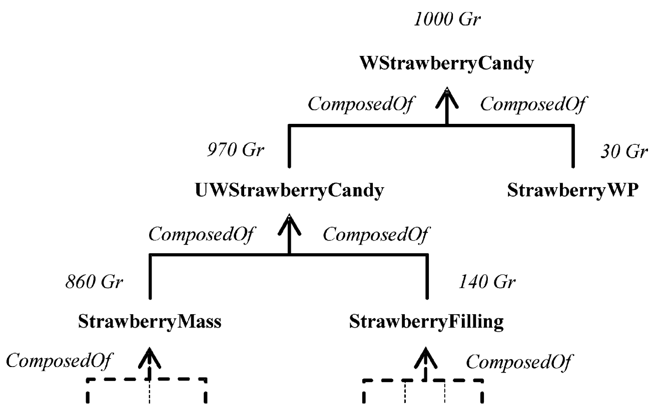

Fig. 9. Partial view of the *WrappedStrawberryCandy* multi-level BOM.

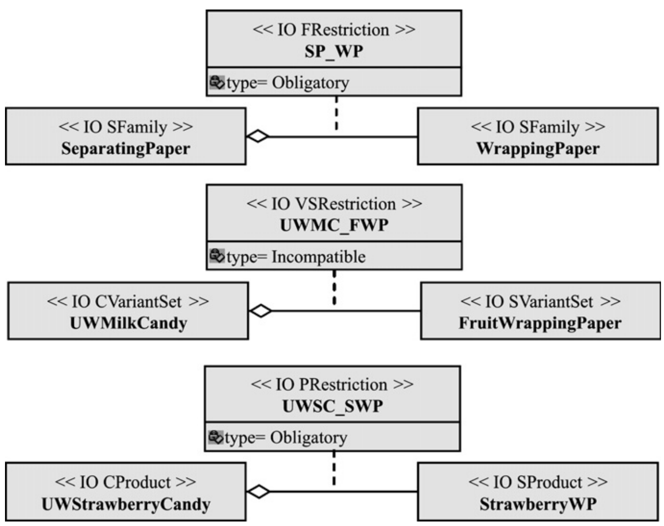

Fig. 11. Some restrictions that appear in the case-study.

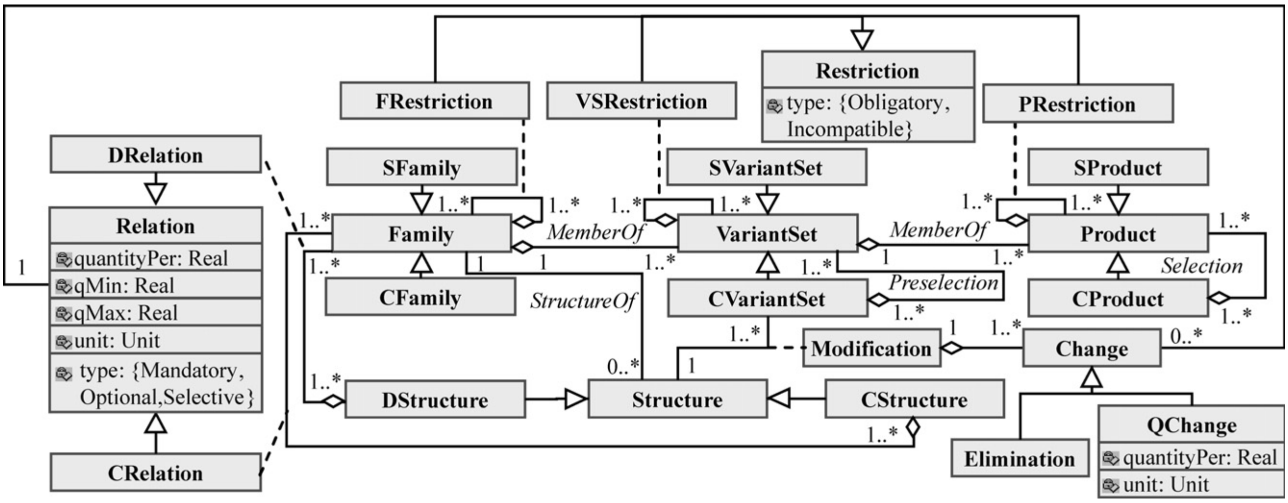

Fig. 12. PRONTO's partial view: handling of structural information.

allowing it to manage all kinds of information (structural and non-structural). Besides, it formalizes the data vertical integration along the AH levels (*Family*, *VariantSet* and *Product*) that occurs during information aggregation and disaggregation processes, which is not supported by other product models. In turn, the value (or values) that a property assumes for a certain product representation, at a given level of the AH, is specified by means of the *PropertyValue* concept. The ontology's extension achieved by incorporating these new concepts is depicted in the class diagram of Fig. 13. The abstraction levels already defined in PRONTO are represented as a specialization of the *ProductAbstraction* class. Additionally, this class is linked to the *Property* one by means of the *PropertyValue* association class.

All the properties include a detailed description of their meaning; moreover, they can be quantitative or qualitative, depending on the associated value type. For qualitative properties, the allowed set of values generally is of *String*, *Symbol* or *Boolean* type. On the other hand, quantitative property values could be *Real*, *Integer* or other numerical types. In this case, the set of possible units of measure must also be defined. Fig. 14 shows a few examples of these two property types.

The proposed model decouples the *Property* concept from the respective values that a given property can assume at the various product abstractions in the same or different levels of the AH. The *PropertyValue* concept includes the range of allowed values in the context of a given association, and the unit of measure whenever it is necessary to specify it. With regards to the value itself, it can be a single one or a set of them.

3.4. Property value classification

The proposed extension takes into account the fact that property values can be obtained from different sources and by resorting to different calculation or retrieval mechanisms. A value classification based on the maintainability and availability

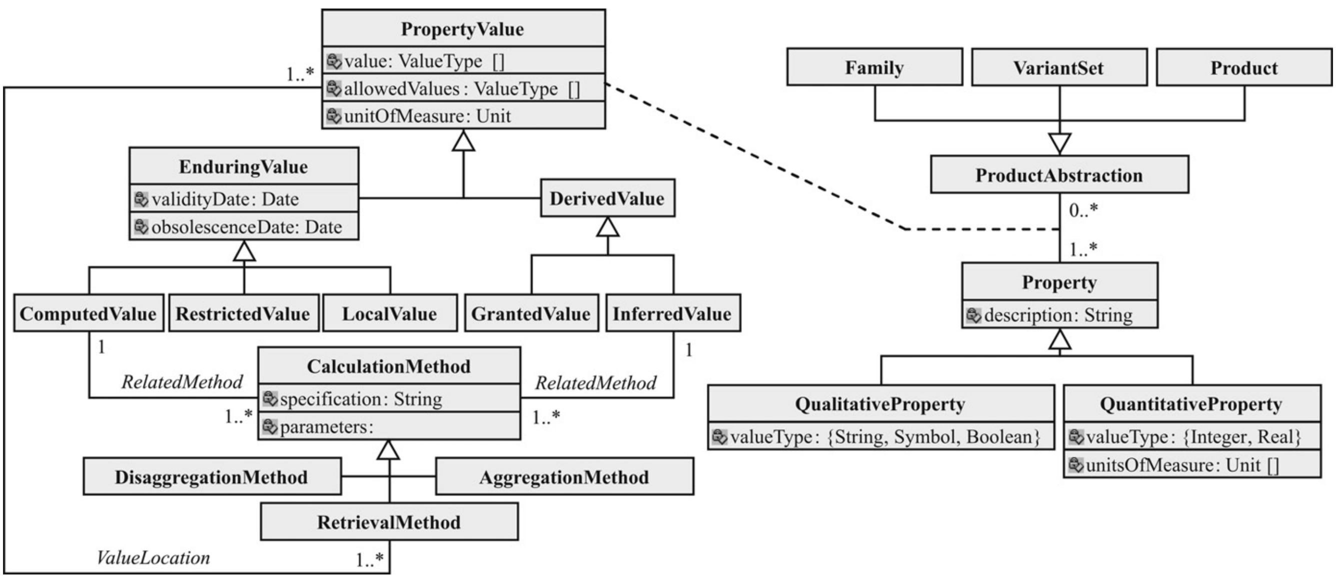

Fig. 13. PRONTO's extension: handling of non-structural information.

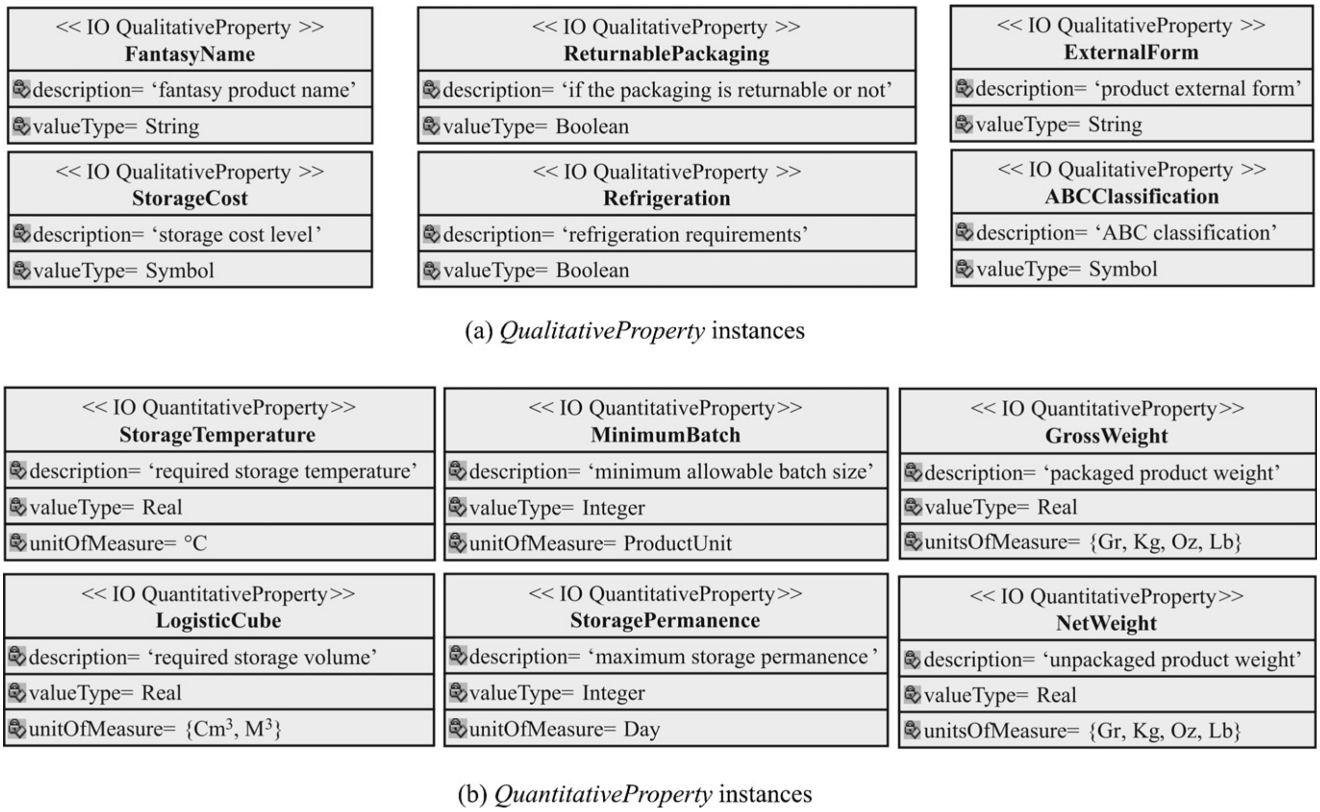

Fig. 14. Some example instances of the *QualitativeProperty* and *QuantitativeProperty* subclasses.

of these data has been proposed. This allows to represent those property values that either exhibit a high frequency of change (thus, information is always out of date and the stored values must be disregarded) or are available in other product abstractions (so it is not necessary to store them twice).

Then, this conceptualization considers two specializations of the *PropertyValue* class, in agreement with the physical existence or not of the property value at the product abstraction level where it is demanded. Thus, the *EnduringValue* class represents the situation in which the value resides physically as

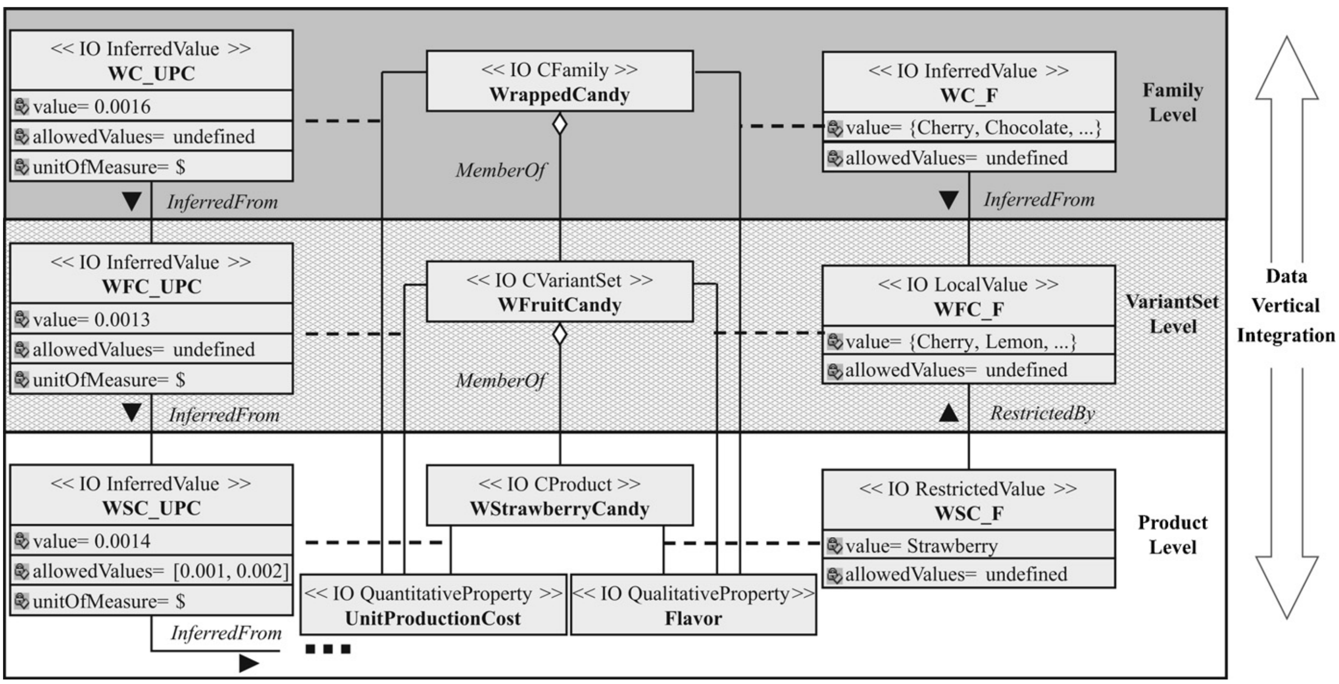

Fig. 15. Some instances of the *PropertyValue* class pertaining to the *InferredValue*, *LocalValue* and *RestrictedValue* categories.

an object related to the *value* attribute of the product abstraction being considered, and *DerivedValue* models the circumstance in which the value is obtained at the moment it is required. In turn, the applicability period of the enduring values is delimited by the validity and obsolescence dates.

Moreover, the two previous categories are specialized according to the nature of the property valuation process. In the case of *EnduringValue*, the following subcategories were identified: (a) *ComputedValue*: the value is calculated by using one or more *CalculationMethods* (see Fig. 13); (b) *RestrictedValue*: the value that a given property assumes in a certain product abstraction is constrained by the value set that the same property assumes in another product representation included at an upper abstraction level, which is related to the original one by means of a *MemberOf* association. In this situation, a range of possible values will not be defined because it will be determined by the condition expressed above; (c) *LocalValue*: the valuation process does not use any information stored in any other product abstraction.

Regarding the *ComputedValue* category, when the value of a given property of a certain substitute product (a family or a variant set) is obtained by means of an aggregation process, the calculation is based on the values that the same property presents in *ProductAbstraction* instances, defined at a lower level and related to the original one by means of *MemberOf* associations (e.g. the value of the annual production of a specific family may be obtained from the aggregation of the values corresponding to its variant sets, which in turn can be obtained by aggregating the annual production values of their associated product instances). In this case, the *CalculationMethod* will use an aggregation mechanism.

In contrast, an information disaggregation process has the purpose of generating detailed data, related to the members of a given product abstraction from data stored in such substitute product. For example, the demand forecasted for a specific variant set can be used to estimate the demand of a given member of this product abstraction by disaggregating the information on the basis of the product market share. In this case, a disaggregation method will be used.

Finally, when neither an aggregation nor a disaggregation process is involved, a retrieval method is utilized. This allows a data horizontal integration among product abstractions defined at the same level of the AH, which could be related inside the SH (e.g. the lead time of a final product is obtained from the components' lead time, the gross weight is calculated from the respective net weight and the packaging weight).

On the other hand, the *DerivedValue* class is specialized into two subcategories: (a) *GrantedValue*: the value is taken from the value of the same property in another product abstraction. Within this context, it might be interpreted as if the value were inherited; (b) *InferredValue*: the valuation process is carried out via a calculation method. As for the *ComputedValue* subcategory, an aggregation, disaggregation or retrieval method can be employed.

With the intention of illustrating these concepts, Figs. 15 and 16 include some instances of the different property value categories related to the candy industry case-study. The attributes (*validityDate* and *obsolescenceDate*), that establish the validity period of the data defined in the *EnduringValue* subclass, were omitted in order to preserve the clarity of the class diagrams.

Fig. 15 depicts a partial view of the abstraction hierarchy in which *WrappedCandy*, *WFruitCandy* and *WStrawberryCandy* participate. At each level, the values of the *UnitProductionCost* and *Flavor* properties are shown. For the first property, the values adopted at the *Family* and *VariantSet* levels (*WC_UPC*

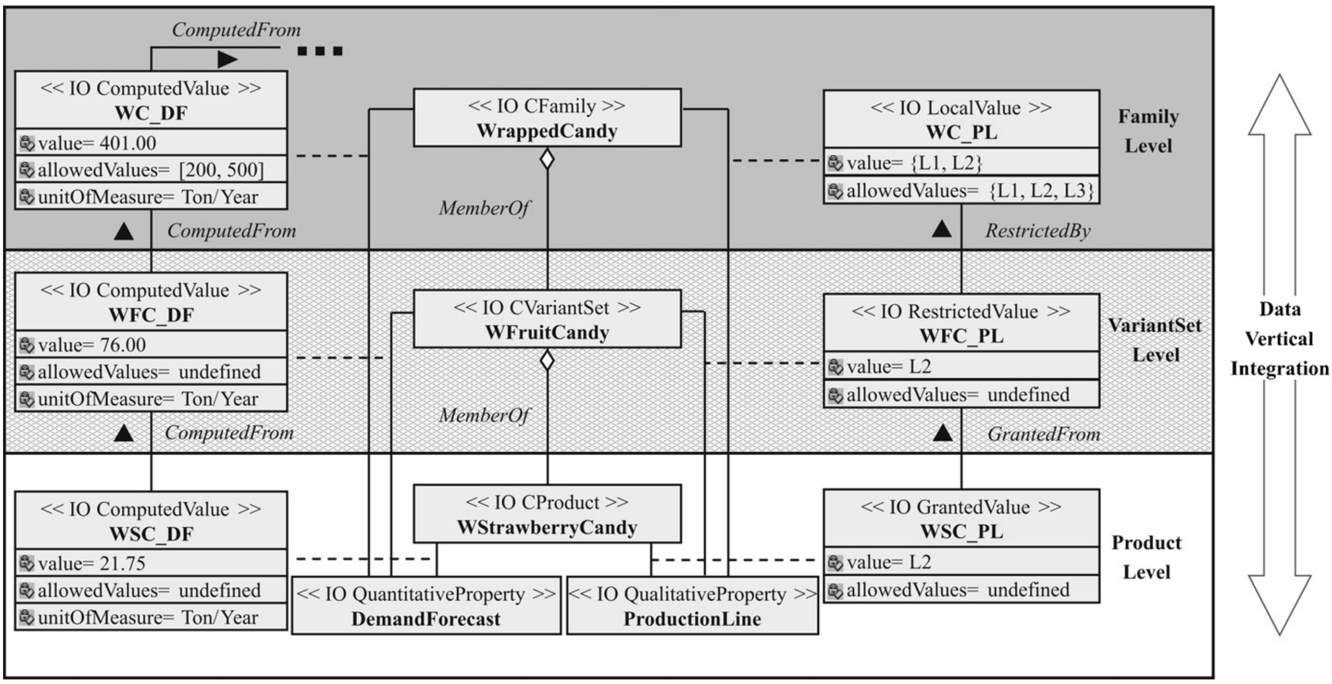

Fig. 16. Some instances of the *PropertyValue* class pertaining to the *ComputedValue*, *LocalValue*, *RestrictedValue* and *GrantedValue* categories.

and *WFC_UPC* instances) are inferred from the values of such property at the product level (*WSC_UPC* instance) by means of aggregation processes. In turn, the unit production cost of the wrapped strawberry candy is inferred from the components' cost (not shown) by means of a retrieval method.

On the other hand, the values of the *Flavor* property at the *VariantSet* level (*WFC_F* instance) impose a restriction on the values that this property could take at the product level (*WSC_F* instance). Besides, the value of this property at the highest level (*WC_F* instance) is inferred from the variant sets' flavors.

The second example (Fig. 16) shows that the value of the *DemandForecast* property at the *Family* level (*WC_DF* instance) is used to compute the values that this property takes at the lower levels (*WFC_DF* and *WSC_DF* instances) by means of disaggregation processes. Likewise, the first one is computed from historical information and sale prospects (not shown).

Furthermore, the production line where the wrapped strawberry candies are manufactured (*WSC_PL* instance) is determined by the value that the *ProductionLine* property assumes at the *VariantSet* level (*WFC_PL* instance). This value is restricted by the values that the above-mentioned property presents at the top level (*WC_PL* instance).

This formalization of the data vertical and horizontal integration would allow automating information aggregation and disaggregation processes required at the time of making strategic, tactical and operational decisions, related with the logistics activities involved in ESCs. Moreover, by means of automatic update mechanisms, that can be executed in predefined periods, product data associated with real and substitute products can always be available and updated in the DPDM system. This task becomes complex and hand-crafted without the explicit representation introduced in this section.

At present, a prototype of a DPDM system that supports the horizontal integration of structural data across the SH is being developed. Actually, OWL [26] was employed to create the system's knowledge base, which is a distributed one. The generic PRONTO concepts are centralized while instances are distributed in the different nodes. Each of them has a JOSEKI Server [27]; thus, all nodes behave as peers. Moreover, they have a local interface, implemented with Java and the Jena framework [28], that supports the underlying distributed query system.

4. Conclusions

Product-related data constitutes a fundamental information source for many industrial activities. There are many applications, from design to sales, that require product information, which many times is exchanged between enterprises. Therefore, the product model should maintain data for every product life cycle stage. The definition of a product ontology establishes a common formal vocabulary to be used for each stakeholder of an ESC, who commits to a given product ontology, accepting the terminology that it prescribes. Particularly, PRONTO considers different abstraction levels in relation to the product concept: *Family*, *VariantSet* and *Product*. Across them it is possible to handle different information types with diverse aggregation degrees.

The separation of the *Property* concept from its associated value (*PropertyValue*) allows products defined at different abstraction levels to share the concept of a certain property and to assign it distinct values at each product abstraction. An extra advantage of such separation is that the definition of properties (their meanings and the permitted value types) can also be shared by the many participants in a SC. Thus, all the instances of the *Property* class could be seen as a repository of ''attribute definitions'' used by several organizations during integration processes.

Besides, the ontology's extension presented in this article formalizes both processes of information aggregation and disaggregation that occur during logistics planning activities. The representation of the calculation method for the *InferredValue* and *ComputedValue* permits to document and register the knowledge associated with such computations.

Finally, it must be remarked that product information is not static. ESCs are subject to continuous internal restructurings due to changes in the offer-demand relations or due to the inclusion of new actors (suppliers, manufacturers, logistic providers, customers). Thus, future work will focus on the development of contextual ontologies to make compatible the product representations introduced by these new actors with the current product model shared by the previous participants of an ESC.

Acknowledgements

This work has been supported by CONICET, UNL, UTN and ANPCyT (PICT 12628).

References

[1] H.L. Lee, S. Whang, e-Business and supply chain integration, in: T.P. Harrison, H.L. Lee, J.J. Neale (Eds.), Practice of Supply Chain Management, Kluwer Academic Publishing, Boston, 2003.
[2] R.B. Chase, N.J. Aquilano, F.R. Jacobs, Production and Operations Management, 8th ed., Irwin/McGraw-Hill, New York, 1998.
[3] A. Saaksvuori, A. Immonen, Product Lifecycle Management, 2nd ed., Springer, Berlin, 2005.
[4] S. Zhang, W. Shen, H. Ghenniwa, A review of internet-based product information sharing and visualization, Computers in Industry 54 (2004) 1–15.
[5] T.R. Gruber, A translation approach to portable ontology specifications, Knowledge Acquisition 5 (1993) 199–220.
[6] S.B. Yoo, Y. Kim, Web-based knowledge management for sharing product data in virtual enterprises, International Journal of Production Economics 75 (2002) 173–183.
[7] T. Berners-Lee, J. Hendler, O. Lassila, The Semantic Web, Scientific American, May 2001.
[8] M. Vegetti, G.P. Henning, H.P. Leone, PRoduct ONTOlogy. An ontology for complex product modeling domain, in: Proceedings of the ENPROMER, Rio de Janeiro, 2005.
[9] D. Simchi-Levi, P. Kaminsky, E. Simchi-Levi, Managing the Supply Chain, McGraw-Hill, New York, 2004.
[10] S. Nahmias, Production and Operations Analysis, 4th ed., McGraw-Hill/Irwin, New York, 2001.
[11] H.M.H. Hegge, Intelligent product family descriptions for business applications, Ph.D. Thesis, Eindhoven University of Technology, 1995.
[12] F.J. O'Donnell, K.J. MacCallum, T.D. Hogg, B. Yu, Product structuring in a small manufacturing enterprise, Computers in Industry 31 (1996) 281–292.
[13] F. Erens, K. Verhulst, Architectures for product families, Computers in Industry 33 (1997) 165–178.

[14] K.A. Olsen, P. Sætre, A. Thorstenson, A procedure-oriented generic bill of materials, Computers & Industrial Engineering 32 (1997) 29–45.
[15] J. Jiao, M.M. Tseng, V.G. Duffy, F. Lin, Product family modeling for mass customization, Computers & Industrial Engineering 33 (1998) 495–498.
[16] P. De Lit, J. Danloy, A. Delchambre, J.M. Henrioud, An assembly-oriented product family representation for integrated design, IEEE Transactions on Robotics and Automation 19 (2003) 75–88.
[17] S.W. Hsiao, E. Liu, A structural component-based approach for designing product family, Computers in Industry 56 (2005) 13–28.
[18] J.M. Usher, An object-oriented approach to product modeling for manufacturing systems, Computers & Industrial Engineering 25 (1993) 557–560.
[19] Y. Chung, G. Fischer, A conceptual structure and issues for an object-oriented bill of materials (BOM) data model, Computers & Industrial Engineering 26 (1994) 321–339.
[20] B. Eynard, T. Gallet, P. Nowak, L. Roucoules, UML based specifications of PDM product structure and workflow, Computers in Industry 55 (2004) 301–316.
[21] R. Sudarsan, S.J. Fenves, R.D. Sriram, F. Wang, A product information modeling framework for product lifecycle management, Computer-Aided Design 37 (2005) 1399–1411.
[22] G. Thimm, S.G. Lee, Y.-S. Ma, Towards unified modelling of product life-cycles, Computers in Industry 57 (2006) 331–341.
[23] G. Booch, J. Rumbaugh, I. Jacobson, The Unified Modeling Language User Guide. Object Technology Series, Addison-Wesley, 1999.
[24] J. Clement, A. Coldrick, J. Sari, Manufacturing Data Structures, John Wiley & Sons, New York, 1992.
[25] E.A. Van Veen, J.C. Wortmann, Generative bill of material processing systems, Production Planning & Control 3 (1992) 314–326.
[26] Web Ontology Language, http://www.w3.org/2004/OWL/.
[27] Joseki, A SPARQL Server for Jena, http://www.joseki.org/.
[28] Jena, A Semantic Web Framework for Java, http://jena.sourceforge.net/documentation.html.

Diego M. Giménez received his Industrial Engineering degree from "Universidad Nacional del Litoral" (UNL), Santa Fe, Argentina, in 2005. He is currently a Ph.D. student in Information Systems Engineering at "Universidad Tecnológica Nacional" (UTN). He also has a Research Fellowship from the National Council for Scientific and Technical Research of Argentina (CONICET), to work at "Instituto de Desarrollo Tecnológico para la Industria Química" (INTEC). His research interests focus on supply chain management (SCM) problems and the representation of product knowledge in the SCM context, topic on which he is preparing his Ph.D. thesis.

Marcela Vegetti received an Engineering degree in Information Systems from "Universidad Tecnológica Nacional" (UTN), Santa Fe, Argentina, in 1999, and also obtained her Ph.D. degree in Information Systems Engineering from UTN in 2007. She currently has a Research Fellowship from the National Council for Scientific and Technical Research of Argentina (CONICET), to work at "Instituto de Desarrollo y Diseño" (INGAR). Her research interests are in the conceptual and formal modeling areas and in the representation of product knowledge.

Horacio P. Leone is a Professor at the Systems Department of "Facultad Regional Santa Fe, Universidad Tecnológica Nacional" (Santa Fe, Argentina), where he currently is the Department Head. He also holds a Researcher position at the National Council for Scientific and Technical Research of Argentina (CONICET), working at "Instituto de Desarrollo y Diseño", where is currently the Institute Director. He obtained his Ph.D. degree in Chemical Engineering from "Universidad Nacional del Litoral" (Santa Fe, Argentina) in 1986 and was a Postdoctoral Fellow at the Massachusetts Institute of Technology (1986–1989). His current research activities focus on semantic web applications to supply chain information systems and enterprise modeling. He has supervised several Ph.D. students.

Gabriela P. Henning is a Professor at "Universidad Nacional del Litoral" (Santa Fe, Argentina) and holds a Researcher position at the National Council for Scientific and Technological Research of Argentina (CONICET), working at INTEC (CONICET-UNL). She received a Ph.D. degree in Chemical Engineering from UNL in 1986 and was a Postdoctoral Fellow at the Massachusetts Institute of Technology (1986–1989), specializing in applications of Artificial Intelligence in Process Engineering. Her current research interests include the development of integrated information systems for production organizations as well as the solution of industrial scheduling and supply chain management problems by means of knowledge-based techniques. She has supervised several Ph.D. students.