Formal Grammars for Product Data Management on Distributed Manufacturing Systems

Rui M. Sousa[1], Paulo J. Martins[1], Rui M. Lima[1],

[1] Department of Production and Systems, School of Engineering, University of Minho,
Campus de Azurém, 4800 058 Guimarães, Portugal
{rms, pmartins, rml}@dps.uminho.pt

Abstract. This work shows how formal grammars with attributes can be advantageously used to deal with two fundamental aspects of product data management - product diversity management and generation of specific product data based on clients' specification – in the context of distributed manufacturing systems, while networks of geographically distant collaborative entities. This contribution will constitute a new component for an existing model, developed by the authors, for dynamic production planning and control which includes the interoperability with industrial equipment. The proposed approach is centered on attributed formal grammars, allowing the formalization of the data representation for each family of products and also of some of the inherent processing (e.g. generation of specific products' bill-of-materials).

Keywords: formal grammars, product data management, bill-of-materials, distributed manufacturing systems.

1 Introduction

Over the last years, the markets' tendency to frequently demand differentiated products with high quality and low prices has become even more accentuated. This tendency has direct implications over two aspects: the products themselves and the correspondent production systems.

Products are no longer strictly defined by the companies, being, instead, defined in some extent by the customers (e.g. Nike's customers can specify online their own sport shoes). Therefore, products' diversity is experiencing a dramatic increase, causing serious problems to the traditional product data management (PDM) systems (where, frequently, each product has its own bill-of-materials) due to the huge volume of information involved. From the PDM perspective, this paper does not intend to develop new models to overcome the problem of products' high diversity. Among other approaches, namely MBOM - modular bill-of-materials [1] and BOMO – bill-of-materials and operations [2], the GBOM - generic bill-of-materials [3] is a relatively recent concept, with recognized effectiveness, specifically designed to deal with that problem. Based precisely on the GBOM concept, the main objective of this paper is the formalization, using attributed formal grammars, of generic product data

representation and of some of the inherent processing, namely the generation of specific product data. Reference [4] use graph grammars to model families of products accordingly to the modular product architecture [5] which also addresses the problem of product family design itself. A detailed state-of-the art review on product family design can be found in [6]. The present paper does not intend to cope with this design problem but rather to deal with existent families of products. Besides the rigour and absence of ambiguities that characterizes the formal approaches, the use of formal grammars reduces the gap between specification and implementation due to particular equivalences between formal grammars and automata [7].

Traditional production systems are also facing serious problems as they are not adequate to produce small quantities of a large diversity of products and, besides, to do that very quickly, with high quality and low price. A significant number of traditional production systems achieve reduced cycle times and low costs only for large quantities of an extremely reduced diversity of products (eventually unitary), i.e. they are oriented to mass production. To overcome this problem, paradigms like mass customization production [8] and customer oriented production [9] have been introduced. When based on these paradigms, systems are expected to respond to customers' specific demand (low quantities and high diversity) keeping the advantages of mass production (reduced cycle times and low costs). In structural/organizational terms, and to overcome the traditional "monolithic" companies, the so-called distributed manufacturing systems (while networks of separate collaborative entities) are expected to dynamically identify and select the resources (which can be geographically distant from each other) that can better respond to a given market opportunity. From the production systems perspective, it is expected that the previously referred formalization, will contribute to the improvement of a concrete distributed production planning and control (PPC) model [10, 11]. In this model, the distributed system for production of a specific product is composed of a network of autonomous processing elements directly related with the bill of materials of that product. It is assumed that each of these elements has the capacity and ability to deliver a product's component. Furthermore, the proposed mechanism for selection of processing elements could dynamically originate different networks for identical products. Therefore it is expected that the ability to cope with high diversity of products will contribute to the improvement of the referred distributed production planning and control (PPC) model.

The previous paragraphs show that the present work is included in an embracing project which involves a number of research areas, namely: formal approaches in manufacturing systems design, product data management, production planning and control, and, industrial automation. This diversity constitutes an additional challenge to the investigation team.

The paper is structured in five sections. After this initial section, a very brief introduction to PDM is provided in section 2, emphasizing the importance of the generic bill-of-materials concept. Formal grammars and correspondent equivalent automata are described in section 3 which also includes the description of the attributed formal grammar concept. On section 4 a specific attributed formal grammar for generic product data representation and processing is developed and an example of its application is provided. Finally, on section 5, some conclusions are outlined, including perspectives of future work.

2 Product Data Management

Product data management (PDM) is one of the most important functional areas of production planning and control (PPC) systems. PDM should provide mechanisms not only to represent product data, but also to make that information available to other functional areas (e.g. commercial management, master scheduling planning, materials requirement planning and products' budgeting).

One of the most important mechanisms involved in PDM is the bill-of-materials (BOM). In general terms the basic BOM represents the structure of components of a specific product. Typically the observed increase on the products' diversity happens because clients have the possibility to select values for some characteristics of the products (e.g. the colour of the sport shoes). Hence, instead of specifying a BOM for each specific product (implying thus huge volumes of information with high levels of redundancy), the specification of a single BOM for each family of products is much more effective from the information management point of view. That is precisely the main purpose of the generic bill-of-materials (GBOM) [3]. The specification of a GBOM for a given family of products includes the definition of a set of parameters which represent relevant characteristics of that family. Later, the instantiation of those parameters with specific values allows the transformation of the GBOM into a BOM of a specific member of the family. Inspired in an example from [3], Fig. 1 represents a basic chair and the correspondent GBOM.

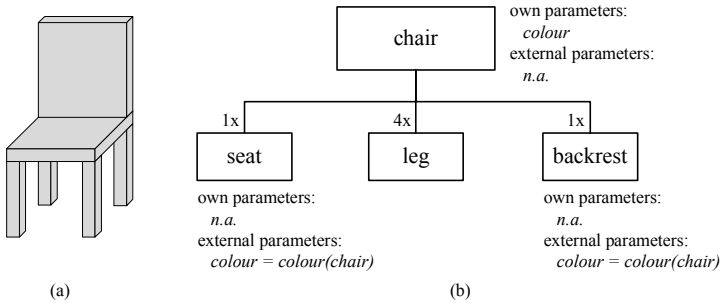

Fig. 1. (a) Chair, (b) Chair's GBOM (adapted from [3]).

In this simple example the customer can choose the colour of the chair and that choice directly determines the colour of both seat and backrest, but has no effect on the legs. For both seat and backrest, colour is an external parameter whose value is inherited from the chair. For the chair, colour is an own parameter and thus should be directly instantiated.

The formalization of this kind of structures brings rigour and absence of ambiguities, but those are not the only reasons to formalize. The formal concept used in this work – the formal attributed grammar – provides mechanisms to model not only the tree structure with correspondent parameters and inheritance relations, but also synthesis relations (sections 3 and 4). The synthesis relations here proposed are not identifiable in the GBOM model [3]. Additionally, as already referred in the previous section, this particular kind of formalization reduces the gap between specification and implementation.

3 Formal Grammars

Basically a formal grammar uses an alphabet (with two types of symbols: terminal and non-terminal), and a set of rewriting rules (productions), to generate words (constituted by terminal symbols) that are subsequently used to represent aspects of a given area (e.g. manufacturing systems general area). The formal grammar concept is defined by several authors [12, 13, 14], but all those definitions are similar and based on Chomsky's definition [15]. Due to notation's adequacy the following definition, presented in [16], is here used: A formal grammar G is a four-tuple $G=(V_T,V_N,S,R)$ where V_T is a finite set of terminal symbols, V_N a finite set of non-terminal symbols ($V_T \cap V_N=\phi$), S is the initial symbol ($S \in V_N$) and R is a finite set of productions.

The existence of a production $\alpha \rightarrow \beta$ in R means that grammar G allows the substitution of the string α by the string β. To illustrate the process of derivation of a word consider the grammar $G=(V_T,V_N,S,R)$ where $V_T=\{m_1,m_2,m_3, \mapsto\}$, $V_N=\{S\}$ and $R=\{S \rightarrow m_i,\ S \rightarrow m_i \mapsto S\}$. To avoid the repeated use of the same index i both rules are subjected to an application condition $1 \leq i \leq 3 \wedge o_i=0$ where $o=(o_1, o_2, o_3)$ is the so called occurrence vector. A possible derivation is $S \Rightarrow m_1 \mapsto S \Rightarrow m_1 \mapsto m_2 \mapsto S \Rightarrow m_1 \mapsto m_2 \mapsto m_3$. A derivation ends when the word only contains terminal symbols (i.e. no more productions can be applied).This particular derivation has three steps and the sequence of applied productions is $(2,2,1)$. The word $m_1 \mapsto m_2 \mapsto m_3$ can be graphically interpreted (Fig. 2).

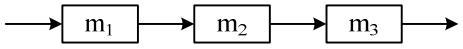

Fig. 2. Interpretation of word $m_1 \mapsto m_2 \mapsto m_3$.

Based on the productions' type, the Chomsky's hierarchy identifies four classes of formal grammars: unrestricted, context-sensitive, context-free and regular. Moreover these grammars are equivalent, respectively, to the following four types of automata: Turing machine (TM), linear bounded (LBA), pushdown (PDA) and finite state (FSA). A detailed analysis on this subject can be found in [7]. The importance of these equivalences resides mainly on implementation (contrarily to formal grammars, automata are easy to implement). These grammars are syntactic mechanisms and to attain semantic aspects, attributed grammars were introduced.

An attributed formal grammar G_a is a triple $G_a=(G,A,P)$ where G is a context-free grammar, A is a finite set of attributes and P is a finite set of assertions. Basically in an attributed grammar any symbol (terminal or non-terminal) may have a set of attributes and every production may have a set of assertions which represent the relations between attributes. With these features, formal grammars become suitable for the PDM area (section 4). In fact each element of a BOM will be represented by a terminal symbol, those elements' parameters are represented by attributes associated to the correspondent terminal symbols and the relations between parameters of different elements are represented by assertions. Assertions allow the representation of not only the inheritance relations referred in section 3, but also of synthesis relations (e.g. the value of a given parameter of a given element is obtained from the values of parameters of other elements below in the hierarchy).

4 Formal Grammar for PDM

This section introduces an attributed formal grammar, denoted as G_1, able to represent the GBOM for families of products and also to process the involved parameters leading thus to the generation of the BOM for each specific product.

The attributed formal grammar $G_1=(G,A,P)$ includes: (i) a context-free grammar $G=(V_T,V_N,S,R)$ where $V_T = \{c_1, \ldots, c_n, \downarrow_1, \ldots, \downarrow_n,],[\}$ with $n \in N$, $V_N = \{S, A\}$ and $R=\{S \rightarrow c_i,\ S \rightarrow c_i[A],\ A \rightarrow \downarrow_i c_i[A],\ A \rightarrow AA,\ A \rightarrow \downarrow_i c_i\}$, (ii) a finite set of attributes A, and, (iii) a finite set of assertions P. While G is a syntactic mechanism, A and P are already associated to semantics and thus their definition is dependent of each family of products. Productions 1, 2, 3 and 5 must have an application condition to avoid the repeated use of the same index i. That condition is $1 \leq i \leq n \wedge o_i = 0$ where $o=(o_1, \ldots, o_n)$ is the occurrence vector. Two possible derivations performed by G are: $S \Rightarrow c_1[A] \Rightarrow c_1[AA] \Rightarrow c_1[AAA] \Rightarrow c_1[\downarrow_2 c_2 AA] \Rightarrow c_1[\downarrow_2 c_2 \downarrow_3 c_3 A] \Rightarrow c_1[\downarrow_2 c_2 \downarrow_3 c_3 \downarrow_4 c_4]$ and $S \Rightarrow c_1[A] \Rightarrow c_1[AA] \Rightarrow c_1[AAA] \Rightarrow c_1[\downarrow_2 c_2[A]AA] \Rightarrow c_1[\downarrow_2 c_2[\downarrow_3 c_3]AA] \Rightarrow c_1[\downarrow_2 c_2[\downarrow_3 c_3] \downarrow_4 c_4 A] \Rightarrow c_1[\downarrow_2 c_2[\downarrow_3 c_3] \downarrow_4 c_4 \downarrow_5 c_5]$. The derivation sequences are *(2,4,4,5,5,5)* and *(2,4,4,3,5,5,5)*, respectively. The generated words, i.e. $c_1[\downarrow_2 c_2 \downarrow_3 c_3 \downarrow_4 c_4]$ and $c_1[\downarrow_2 c_2[\downarrow_3 c_3] \downarrow_4 c_4 \downarrow_5 c_5]$, may have the graphical interpretation represented on Fig. 3.

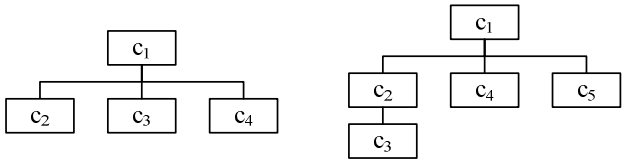

Fig. 3. Interpretation of words $c_1[\downarrow_2 c_2 \downarrow_3 c_3 \downarrow_4 c_4]$ and $c_1[\downarrow_2 c_2[\downarrow_3 c_3] \downarrow_4 c_4 \downarrow_5 c_5]$.

Symbol c_i represents the element i of the structure and $\downarrow_i$ represents the relation between c_i and its parent. Thus, using this interpretation, grammar G is able to represent any structure similar to those represented on Fig. 3 and, obviously, this class of structures can be used to represent BOM. The pushdown automaton (PDA) T equivalent to G can be specified by the state diagram represented on Fig. 4.

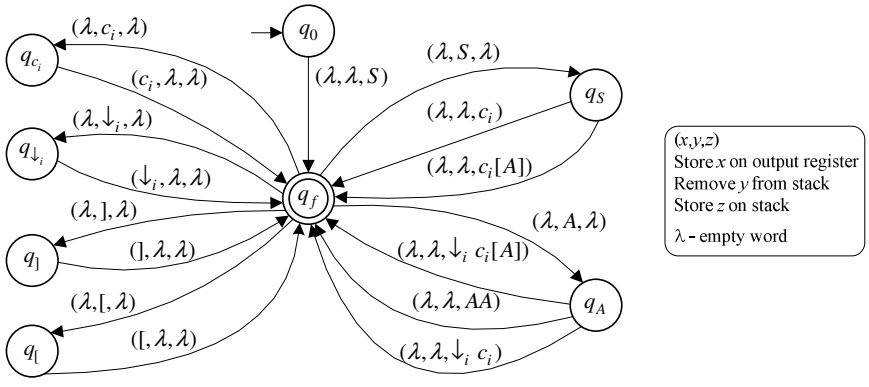

Fig. 4. State diagram of the PDA T equivalent to grammar G.

The GBOM example (chair's family) presented on section 2 (Fig. 1) will now be extended with more parameters in order to demonstrate the application of G_1. The first step is the definition of the structure. The analyst will conduct the process, which is based on the PDA T (Fig. 4), indicating the product and its components (and correspondent quantities), until the desired structure is achieved (Fig. 5).

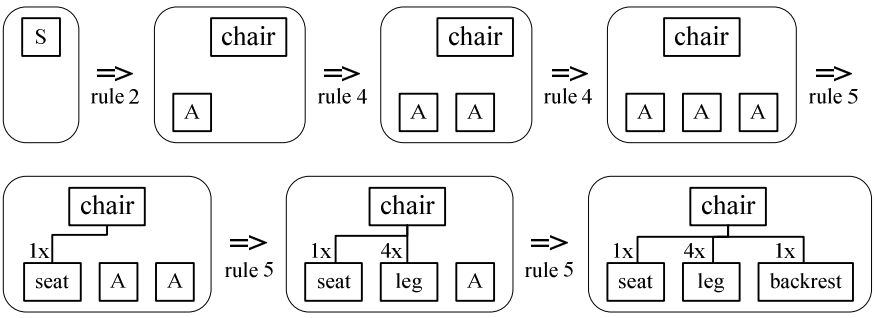

Fig. 5. Derivation process for chair's family GBOM.

In this particular case the derivation sequence recorded at the end of the structure definition is *(2,4,4,5,5,5)*. Note that the analyst is not aware of this sequence – it is just an internal representation used by the PDA *T*.

In the next step the analyst indicates the parameters (Table 1) for each element (i.e. he indicates the set *A* of attributes for each symbol c_i and $\downarrow_i$ of G_1).

Table 1. Parameters for the chair's family GBOM.

Symbol	Description	EIP	ESP	OP
c_1	chair		cost	totalHeight, width, depth, seatHeight
c_2	seat	width, depth		cost
c_3	leg	height		cost
c_4	backrest	height, width		cost
$\downarrow_2$	c_2-c_1 rel.			quantity
$\downarrow_3$	c_3-c_1 rel.			quantity
$\downarrow_4$	c_4-c_1 rel.			quantity

Parameters can be: external inherited (EIP), external synthesized (ESP) or own parameters (OP). Finally the analyst should indicate how the parameters are related to each other (i.e. he indicates the set P of assertions for each production of G1 derivation sequence). Thus, on the first derivation step (rule $S \rightarrow c_1[A]$) the OP assertions are *totalHeight(chair)="user input"*, *width(chair)="user input"*, *depth(chair)= "user input"* and *seatHeight(chair)= "user input"*, meaning that the referred parameters' values should be defined by the user. The only ESP assertion is *cost(chair)=quantity(seat)*cost(seat)+quantity(leg)*cost(leg)+quantity(backrest)* cost(backrest)*. On the second and third derivation steps (rule $A \rightarrow AA$) no assertions are necessary as the symbol involved has no parameters. On the fourth derivation step

(rule $A \rightarrow \downarrow_2 c_2$) the EIP assertions are *width(seat)=width(chair)* and *depth(seat)=depth(chair)*, meaning that both EIP inherit their value from the chair. The OP assertions are *cost(seat)=seatCostTable[width(seat)*depth(seat)]*, and thus the cost of a seat is defined to be dependent of its dimensions, and *quantity(seat)=1*. For next step (rule $A \rightarrow \downarrow_3 c_3$) the only EIP assertion is *height(leg)=seatHeight(chair)* and the OP assertions are *cost(leg)=legCostTable[height(leg)]*, and *quantity(leg)=4*. For the last derivation step (rule $A \rightarrow \downarrow_4 c_4$) the EIP assertions are *height(backrest)=totalHeight(chair) - seatHeight(chair)* and *width(backrest)= width (chair)*, and the OP assertions are *cost(backrest)=backrestCostTable[height (backrest)*width(backrest)]*.

At this point, if the cost's tables are instantiated, G_1 is able to generate the BOM for any particular product of the chair's family. After the "user input" parameters have been instantiated (OP parameters of the chair), the automaton T (Fig. 4) starts the derivation sequence *(2,4,4,5,5,5)*, and, as the sequence proceeds, calculates all the EIP and OP (using the correspondent assertions). Then it calculates all the ESP (only one in this case: *cost(chair)*) going through the derivation sequence in the opposite direction *(5,5,5,4,4,2)* and using the correspondent assertions (note that this mechanism of going through the sequence in one and in another direction is one of the features of the attributed formal grammars, here effectively used). In other words, the client chooses the dimensions he wants for the chair (totalHeight, width, depth and seatHeight) and the system calculates the dimensions of each component (seat, leg and backrest) and the cost of the chair.

The previous example has shown that the developed grammar G_1 has achieved the intended purposes: capability to represent generic product data and to generate specific product data (only at the moment it becomes necessary).

5 Conclusions

The main objective of this work was achieved: the formalization, based on the attributed formal grammar concept, of the generic bill-of-materials (GBOM) model. The developed G_1 grammar formally describes the generic structure of the product's family, the parameters of each element of that structure (product/subassembly /component) and the relations that exist between parameters of different elements (inheritance and synthesis). The inclusion of synthesis relations is a contribution of this work as the original GBOM model does not mention them. Synthesis relations allow the modeling of situations where the value for a given parameter of a given element must be obtained from the values of parameters of other elements from lower hierarchical levels. Thus, G_1 is able to represent the GBOM for almost any family of products and to generate the BOM for any specific product of that family. Additionally, the equivalent pushdown automaton T (Fig. 4) provides a specification of the G_1 core, more close to the implementation stage. Note that neither the analysts nor the users have to know the formal grammar concept.

The developed formalization contributes to the improvement of the distributed production planning and control (PPC) model referred in the paper's introduction,

because in that model there is, for each product, a direct relation between autonomous processing elements and BOM's elements.

In terms of future work the next task will be the update of the distributed PPC system's prototype (the automaton T will provide the core for the software module). However the developed formalization may also be applied to other manufacturing paradigms. Another possible improvement involves G_1 itself - for now only one parameter (quantity) was associated to symbol $\downarrow_i$ (which represents the relation between component c_i and its parent). Other parameters can be added allowing G_1 to deal with, for example, information about the operations necessary to assemble c_i.

References

1. Orlicky, J.A., Plossl, G.W., Wight, O.W.: Structuring the Bill-Of-Materials for MRP. Product Inventory Management 13(4), (1972)
2. Jiao, J., Tseng, M.M., Ma, Q., Zou, Y.: Generic Bill-of-Materials-and-Operations for High-Variety Production Management. Concurrent Engineering: Research and Applications 8(4), 297--322 (2000)
3. Hegge, H.M.H., Wortmann, J.C.: Generic Bill-Of-Material: a New Product Model. International Journal of Production Economics 23(1-3), 117--128 (1991)
4. Du, X., Jiao, J., Tseng, M.M.: Graph Grammar Based Product Family Modeling. Concurrent Engineering: Research and Application 10(2), 113--128 (2002)
5. Ulrich, K.: The Role of Product Architecture in the Manufacturing Firm. Research Policy 24(3), 419--440 (1995)
6. Jiao, J., Simpson, T.W., Siddique, Z.: Product Family Design and Platform-based Product Development: a State-of-the-art Review. Journal of Intelligent Manufacturing 18, 5--29 (2006)
7. Sousa, R.: Contribution to a Formal Theory of Production Systems. PhD thesis (portuguese language), Production and Systems Department, University of Minho (2003)
8. Gilmore, J.H., Pine, B.J.: The Four Faces of Mass Customization. Harvard Business Review 75(1), 32--49 (1997)
9. Mousavi, A., Adl, P., Rakowski, R.T., Gunasekaran, A.: Design of a Production Planning System Using Customer Oriented Design and Resource Utilisation (CODARU). The International Journal of Advanced Manufacturing Technology 17(11), 805--809 (2001)
10. Lima, R., Sousa, R., Martins, P.: Distributed Production Planning and Control Agent-based System. International Journal of Production Research 44, 3693--3709 (2006)
11. Lima, R., Sousa, R.: Agent-based Prototype for Interoperation of Production Planning and Control and Manufacturing Automation. In: 12th IEEE Conference on Emerging Technologies and Factory Automation, (2007)
12. Denning, P.J., Dennis, J.B., Qualitz, J.E.: Machines Languages and Computation. Prentice-Hall Inc. (1978)
13. Mikolajczak, B.: Algebraic and Structural Automata Theory. North-Holland (1991)
14. Révész, G.E.: Introduction to Formal Languages. Dover Publications Inc. (1991)
15. Chomsky, N.: On Certain Properties of Grammars. Information and Control 2, 137--167 (1959)
16. Sousa, R., Putnik, G.: A Formal Theory of BM Virtual Enterprises Structures. In: Camarinha-Matos, L. (eds) Emerging Solutions for Future Manufacturing Systems, Springer, pp. 315--322, ISBN: 0387228284, (2004)