Applying Description Logic to Product Behavioral Design within Advanced CAD Systems

François de Bertrand de Beuvron
Amadou Coulibaly
LICIA,
INSA Strasbourg, FRANCE
beuvron@liia.u-strasbg.fr
amadou.coulibaly@insa-strasbourg.fr

Abstract

In this paper, we investigate the use of Description Logic (DL) for representing Product Behavioral constraints in Computer Aided Design (CAD) Systems. In an integrated design approach, the topological and geometrical model provided by traditional CAD-systems should be completed by semantic data. These additional descriptions and constraints are often expressed by feature structures, or by object-centered representations such as the EXPRESS language included in the STEP standard. Before using this EXPRESS Model for the Behavioral analysis, a coherence control is required. We propose to use DL as a formal tool in order to test this coherence. For this purpose, we first define a mapping from EXPRESS-G into the $\mathcal{ALCNI}$ description logic. An example is then given for product maintainability analysis.

keywords: CAD, Integrated Design, Product Modelling, Description Logic Application

1 Introduction

Traditional CAD systems represent mechanical objects by their topological and geometrical features. These properties however represent only a small part of the information needed during the product life cycle. Many proposals for integrating these constraints in the early stages of the design process have been developed [10]. Most of them deal with manufacturing and assembling constraints [11]. In a general framework described in [5], the product is represented within a three dimensional (FSB) space with the product functions, structures, and behavior (see figure 1A). In the next section, we will present the FSB design approach, and give a short introduction to the EXPRESS language that is used for the product analysis. Finally, in section 3, we consider the accessibility of components as a specific criterion of maintainability, and we show that expressive description logic can nicely and declaratively describe such criterion.
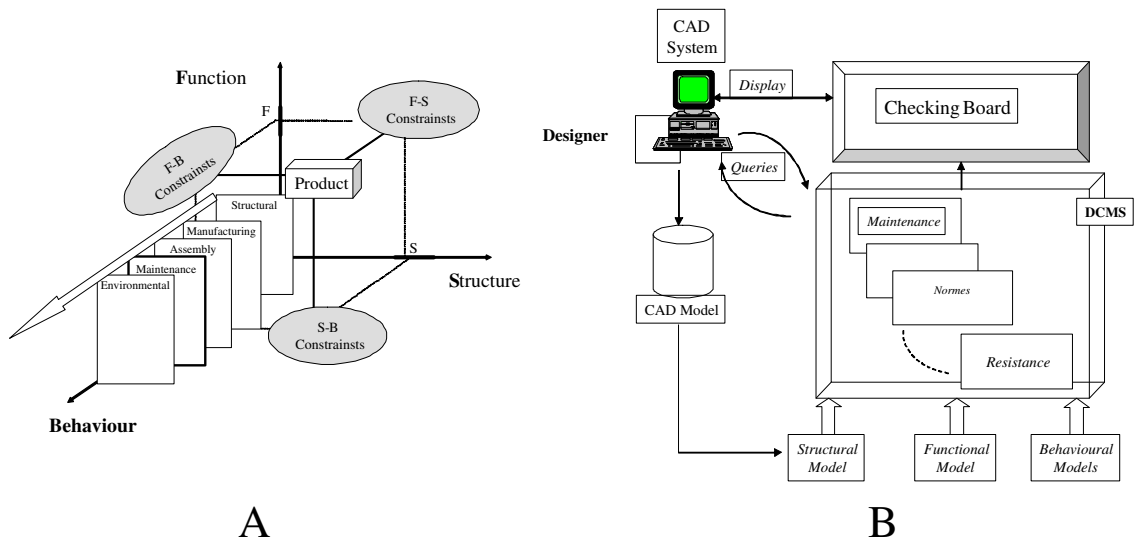
Figure 1: A : FSB Product representation ; B : Integration into CAD System

2 Behavioural Design Approach

Figure 1B shows the integration of the FSB approach in a CAD System: The structure of the product is first modelled within a CAD system. Then, this model is completed with semantic data. The EXPRESS language described below is used to represent the product overall properties. Before using this EXPRESS representation for behavioral analysis, a coherence control of the semantic data is required. Description Logic is used to perform this control.

The EXPRESS language is an expressive object oriented (OO) knowledge representation formalism. It has been defined within the STEP standard [1]. Subsequent parts of the STEP ISO standard define a large knowledge base of EXPRESS schemata related with CAD. EXPRESS is a very expressive representation language that includes:

- The definition of functions and procedures in a Pascal-like, imperative programming language.

- Attributes may be mono-valued (such as partID or suppliedBy) or multi-valued (such as hasPart). In case of multi-valued attributes, you can precise the lower and upper cardinality, and the type of the collection (i.e. set, bag, list, or array).

- Derived attributes may be defined using some composition of other attributes (such as $fabName = suppliedBy \circ name$) or more generally by using any function (as $maxIntensity = calIntensity(SELF.voltage, SELF.power)$).

This section tries to give some flavor of this OO language using an example. EXPRESS-G is a graphical representation of EXPRESS that only includes part of the overall language expressivity. EXPRESS-G can only describe static components of EXPRESS like entities, attributes, hierarchies, but cannot give the definition of
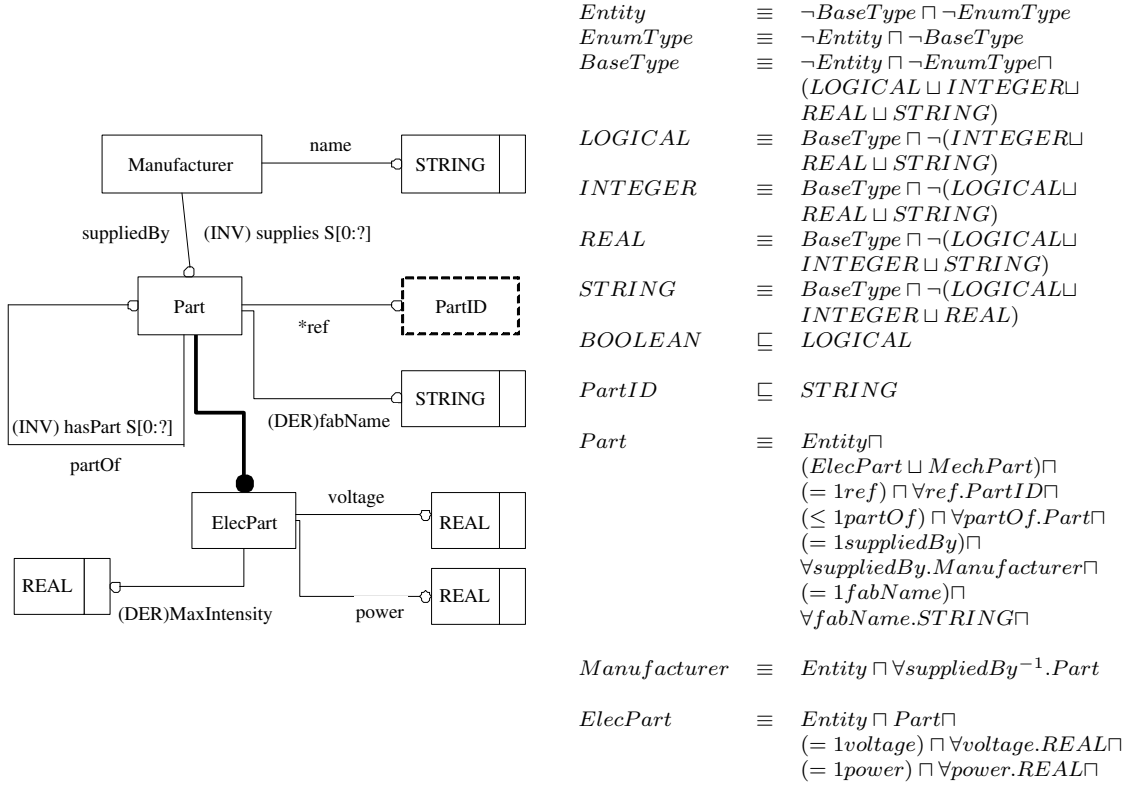
$$
\begin{aligned}
Entity &\equiv \neg BaseType \sqcap \neg EnumType \\
EnumType &\equiv \neg Entity \sqcap \neg BaseType \\
BaseType &\equiv \neg Entity \sqcap \neg EnumType \sqcap \\
&\quad (LOGICAL \sqcup INTEGER \sqcup \\
&\quad REAL \sqcup STRING) \\
LOGICAL &\equiv BaseType \sqcap \neg(INTEGER \sqcup \\
&\quad REAL \sqcup STRING) \\
INTEGER &\equiv BaseType \sqcap \neg(LOGICAL \sqcup \\
&\quad REAL \sqcup STRING) \\
REAL &\equiv BaseType \sqcap \neg(LOGICAL \sqcup \\
&\quad INTEGER \sqcup STRING) \\
STRING &\equiv BaseType \sqcap \neg(LOGICAL \sqcup \\
&\quad INTEGER \sqcup REAL) \\
BOOLEAN &\sqsubseteq LOGICAL \\
PartID &\sqsubseteq STRING \\
Part &\equiv Entity \sqcap \\
&\quad (ElecPart \sqcup MechPart) \sqcap \\
&\quad (= 1ref) \sqcap \forall ref.PartID \sqcap \\
&\quad (\leq 1partOf) \sqcap \forall partOf.Part \sqcap \\
&\quad (= 1suppliedBy) \sqcap \\
&\quad \forall suppliedBy.Manufacturer \sqcap \\
&\quad (= 1fabName) \sqcap \\
&\quad \forall fabName.STRING \sqcap \\
Manufacturer &\equiv Entity \sqcap \forall suppliedBy^{-1}.Part \\
ElecPart &\equiv Entity \sqcap Part \sqcap \\
&\quad (= 1voltage) \sqcap \forall voltage.REAL \sqcap \\
&\quad (= 1power) \sqcap \forall power.REAL \sqcap
\end{aligned}
$$

Figure 2: an EXPRESS-G shema and corresponding $\mathcal{ALCNI}$ TBox

derived attributes, and of course no definition of the algorithmic part of EXPRESS contained in function nor the constraints stated in the rules. In this graphical representation, (see figure 2) the proximity between EXPRESS-G and the class diagrams of UML clearly appear as has been studied in [2]. Using DL as a formal representation of UML diagrams has been presented in [4]. As noted for UML, the DL translation provide a clear semantic for EXPRESS-G schemata. Furthermore, the consistency of the schemata can be checked.

By lack of space in this short paper, we will not give the translation rules from EXPRESS-G to $\mathcal{ALCNI}$. $\mathcal{ALCNI}$ is the standard $\mathcal{ALC}$ description logic extended with (non-qualified) number restriction and inverse roles (see [8]). An example of such translation is given in figure 2, that will be further extended in the next section.

3 An application in design for maintainability.

We present an illustration of our approach in the scope of design for maintainability. In this context, the case of the accessibility of critical components have been considered. In a product some components, considered as critical have to be extractable. For simplicity, we will consider only two kind of links between components : welded or screwed. We want to ensure that critical parts are extractable.

Part-Whole relationship has already been studied within DL in [3] [9]. In the $\mathcal{ALCNIH}_{\mathcal{R}+}$ DL, that include transitive roles, and role hierarchies, it is possible to refine the definition of the TBox of the previous section by defining the set $\mathcal{R}^+ = \{partOfT, accessiblePartOfT\}$ of transitive roles, and the partial order $\subseteq$ on roles as the reflexive and transitive closure of $\prec$:

$$
\begin{array}{lcl}
partOf & \prec & partOfT \\
accessiblePartOfT & \prec & partOfT \\
weldedIn & \prec & partOf \\
screwedIn & \prec & partOf \\
screwedIn & \prec & accessiblePartOfT
\end{array}
$$
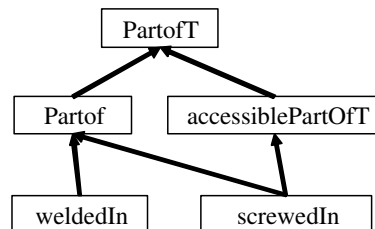
The additional information that weldedIn and screwedIn are pairwise disjoint roles $((= 0\ weldedIn) \sqcup (= 0\ screwedIn))$ must be inserted either as a general TBox axiom, or included in the definition of the Part Concept. It is now possible to define for example extractable component or assembly that can be totaly dismantled.

$$
\begin{array}{lcl}
Extractable & \equiv & Part \sqcap \forall weldedIn.\bot \sqcap \forall partofT.\forall weldedIn.\bot \\
CanBeDismantled & \equiv & Part \sqcap \forall partOfT^-.\forall weldedIn.\bot
\end{array}
$$

The example sketched in this section shows that expressive description logic such as $\mathcal{ALCNIH}_{\mathcal{R}+}$ allow for a declarative representation of the product structure. But the previous definitions have no explicit representation in the EXPRESS schema (they may be hard-coded in ad-hoc functions). General knowledge about mechanical products can therefore be expressed in DL by adding $\mathcal{ALCNIH}_{\mathcal{R}+}$ axioms to the TBox created by the translation process.

4   Conclusion and Perspectives

In this paper, we tried to show the place where DL can be effectively used in the general framework of an integrated CAD system. In this perspective, we define a translation from the widely used EXPRESS-G schemata to the $\mathcal{ALCNI}$ DL. We also have shown that expressive DL including role hierarchies and transitive roles may be successful in expressing a specific maintenance criterion.

Many points need to be clarified in this on-going research. On the DL side, the proposed mapping, while covering almost all the features of the EXPRESS-G schemata is still a strict subset of the complete EXPRESS language. Extensions of the proposed mapping should integrate more specific user types (with so-called concrete domains [7]), some limited kind of derived attributes (role composition [6]), and some of the constraint rules (using general axioms in the TBox)

References

[1] ISO 10303-1:1994. Industrial automation systems and integration - product data representation and exchange - part 1: Overview and fundamental principles.

[2] Florian Arnold and Gerd Podehl. Best of both worlds - a mapping from EXPRESS-G to UML. In Jean Bézivin and Pierre-Alain Muller, editors, *The Unified Modeling Language, UML'98 - Beyond the Notation. First International Workshop, Mulhouse, France, June 1998, Selected Papers*, volume 1618 of *LNCS*, pages 49–63. Springer, 1999.

[3] Alessandro Artale, Enrico Franconi, Nicola Guarino, and Luca Pazzi. Part-whole relations in object-centered systems: An overview. *Data and Knowledge Engineering (DKE) journal*, 20:347–383, 1996.

[4] Daniela Berardi, Diego Calvanese, and Giuseppe De Giacomo. Reasoning on uml class diagrams using description logic based systems. In *Proc. of the KI'2001 Workshop on Applications of Description Logics*. CEUR Electronic Workshop Proceedings, http://ceur-ws.org/Vol-44/, 2001.

[5] Amadou Coulibaly, Egon Ostrosi, and Bernard Mutel. A mechanical integrated design approach based on the product functional, structural and technological specification models. In *proceeding of IDME98, Compiègne, May 27-29*, 1998.

[6] F. Grandi. On expressive description logics with composition of roles in number restrictions. In *Proc of. 9th Intl' Conf. on Logic for Programming Artificial Intelligence and Reasoning (LPAR 2002)*, pages 202–215, Tbilisi (Georgia), 2002. Springer-Verlag.

[7] Volker Haarslev, Ralf Mller, and Michael Wessel. The description logic $\mathcal{ALCNH}_{\mathcal{R}^+}$ extended with concrete domains: A practically motivated approach. In *Proceedings of International Joint Conference on Automated Reasoning, IJCAR'2001*, pages 29–44, Siena, Italy, 2001. Springer-Verlag.

[8] I. Horrocks, U. Sattler, and S. Tobies. A PSpace-algorithm for deciding $\mathcal{ALCNI}_{R^+}$-satisfiability. LTCS-Report 98-08, LuFg Theoretical Computer Science, RWTH Aachen, Germany, 1998.

[9] Patrick Lambrix. *Part-Whole Reasoning in Description Logics*. PhD thesis, Linkoping, 1996.

[10] Pierre-alain Yva. Le contrôle de cohérence des modèles en conception routinière de produits industriels. In *PRIMECA*. Nancy, Université d'automne, 1999.

[11] Gilles Zwingelstein. *La maintenance basée sur la fiabilié/diagnostique et maintance"*. Hermes, 1996.