Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computers in Industry 57 (2006) 331–341
COMPUTERS IN INDUSTRY
www.elsevier.com/locate/compind

Towards unified modelling of product life-cycles

G. Thimm *, S.G. Lee, Y.-S. Ma

School of Mechanical & Aerospace Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798, Singapore

Received 11 November 2004; accepted 12 September 2005
Available online 28 November 2005

Abstract

This paper presents the potential of modelling a product's life-cycle using the Unified Modelling Language (UML). The potential benefits and limitations are discussed. An example of a vacuum cleaner is cited in support of this approach. Model consistency across the various life cycle stages of the product is of major concern and an algorithm for constraint management is proposed and prospective research directions highlighted.

Keywords: Unified Modelling Language; Product life-cycle; Business process; Product model

1. Introduction

Product Life-Cycle Management (PLM) is a strategic business approach that consistently manages all life-cycle stages of a product, commencing with market requirements through to disposal and recycling (see Fig. 1). PLM involves a multitude of stake holders (e.g., customers, suppliers, and regulators), who require various levels of detail and representations of information. For example, the cost accountant may wish to track the costs incurred at certain life cycle stages; regulatory bodies are concerned with data on quality levels and end-of-life disposal options. The type and quantum of data modelled and how much of that data should be visible depends on the desired granularity.

At the February 2004 Georgia Tech-Industry Symposium in Atlanta, USA, a majority of the participants expressed their desire for an ontology for PLM under-pinned by a formal modelling process [private communication]. Twenty-eight percent of the 75 participants (of which 35 were from industry) identified "*single semantic PLM language*; *ontologies*; *data dictionaries*" as priority research thrusts. UML, a graphical modelling language used for computer soft- and hard-ware development [1,2] , offers just this, although it was conceived for object-oriented programming, and therefore, has limitations if applied to other disciplines. Not surprisingly, "*explicit PLM use—cases followed by a formal process modelling UML for PLM*" was voted in third place by 23% of the Georgia Tech-Symposium participants. It can be concluded that industry is in need of a formal modelling technique for PLM embedded in a computer-supported framework, towards which the authors consider this publication a first step.

Nevertheless, to the best knowledge of the authors, no modelling framework using a high-level, top-down modelling technique exists that captures all the aspects of a product's life-cycle stages and translates or connects them seamlessly. The modelling process is not a major concern and conventional approaches as described in [3,4] are suitable. Past experience shows that a translation of models (described in terms proper to each stage) is flawed by information losses and one-way data transfer. Consequently, a PLM modelling framework has to use a unique language for all stages, raising the question of the ideal candidate language. The authors are of the opinion that UML is the most promising candidate, for reasons detailed in Section 1.1. It is not intended here to provide a fine-grained, bottom-up modelling technique, as for example, in feature-based modelling [5,6] , but rather a complementary approach. On the other hand, the authors intend to implement a tool for the creation of such models and to allow them to interact with, for example, CAE tools.

1.1. Why UML is a good PLM modelling language

A PLM-oriented derivative of UML (PLMUL) has many advantages over other approaches and existing modelling

* Corresponding author. Tel.: +65 67904415; fax: +65 67911859.
E-mail address: mgeorg@ntu.edu.sg (G. Thimm).
URL: http://www.drc.ntu.edu.sg/users/mgeorg

0166-3615/$ – see front matter
doi:10.1016/j.compind.2005.09.003

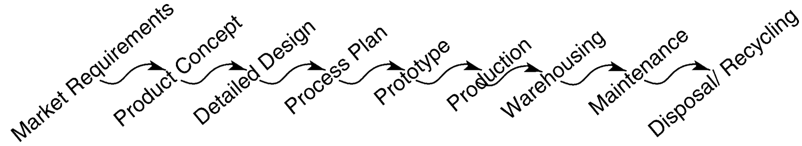

Fig. 1. Stages in product life-cycle management.

techniques (for example, IDEF0 [7] , work-flow modelling [8]). Some of these are:

- Industry has widely accepted UML as a modelling language:
  (1) UML can model business processes to some extent [9,10] and underpins various commercial business process planning tools. It is compelling that some business processes are regarded as a part of a product life-cycle and *vice versa* (see Sections 2.4 and 2.5).
  (2) UML is presently the most versatile modelling technique in industry.
  (3) The Object Management Group [11](an international, not-for-profit consortium including several international companies) endorses UML and uses it in or with other highly industry relevant specifications. Examples of such specifications are the Model Driven Architecture (MDA) or the Common Warehouse Meta-model (CWM).
- The same syntax, that is, the same graphical symbols, can be used across product life-cycle stages. Although a developer only visualises or modifies a very limited number of views at a time, changes are reflected throughout the entire model, which fosters the consistency of PLM models across life-cycle stages. This also assures scalability: rather coarse models can be subsequently refined until a level of detail is reached that allows for example, prototyping or production.
- Modern machines often rely on control by complex software systems modelled using UML. Therefore, if the UML approach is extended to non-software systems, the same supporting tools can be build upon.
- Product models show, depending on the stage of modelling and the role of the persons involved, various levels of detail. UML attends to this and allows purpose-oriented views, favouring communication between designer, project manager, process planner, client, etc.
- UML is consistent with state-of-the-art concepts like functional design [12,13] and end-of-life disposal [14–16].
- UML is an information-rich representation; models can be tested for consistency, analysed, or translated into other representations (Gantt charts, bills-of-material, and so on; see Section 2.5).

1.2. Data modelling for PLM

Although the points stated in Section 1.1 show that UML is a good candidate for a PLM modelling language, its applicability to detailed product design of, for example, mechanical parts is unproven. Although UML is unable to directly describe geometry at this moment, the authors are confident that it is still applicable as it has the same foundation as certain feature-based modelling approaches (all use object-oriented techniques including inheritance, composition, association, and so on). Examples for such feature-based approaches are standard component libraries [17] , assembly feature templates [18,19] , multiple-view feature modelling [20,21] , and unified features [5,22].

These approaches differ mainly from the approach suggested in this paper in that they are very much focused on product details and on developing more universal modelling techniques (that is, a bottom-up approach, [20] has some top-down aspects). For example, in [22] , the STEP standard is extended to include *unified features*in order to rationalise process planning and design for instance, as well as to foster consistent modelling. A similar approach covering four different life-cycles from the conceptual design stage to product assembly is discussed in [20,21].

Opposed to this, the proposed *Product Life-Cycle Unified Modelling Language*(PLUML) is an attempt to model top-down, starting with general, macro models and then working down towards more detailed models. Certainly, either approach has its advantages and draw-backs and it is the hope of the authors that, as in software development, both approaches complement each other.

1.3. Possible issues in the UML modelling of PLM

As the examples in Section 2 show, the symbols and notions of UML are rather easily interpreted in a more general engineering setting. However, a closer examination reveals some potential problems. The most prominent one is probably the consistency of models: the propagation of changes in UML software-models as well as consistency checks among models. The need for consistency checks within and across engineering models is well recognised and researched on (see for example, [20,22–24]), but no product life-cycle encompassing approach exists. These approaches to maintain consistency are very much limited to part geometry and (low-level) feature compatibility.

Furthermore, even if the semantics of UML is expected to be powerful enough to fulfill PLM-needs, the UML symbols are visually not explicit and the existing associations not quite appropriate. In practice, an engineer probably would like to easily distinguish between electrical, mechanical, and entities of other categories. The consistency of PLUML models is further addressed in Section 3 , based on the case example of a vacuum cleaner. Other issues are the foci of future research as given in Section 4.

2. A PLUML case study: a vacuum cleaner

The purpose of the case example is to demonstrate the feasibility of using UML for PLM including business