Mechanical Systems and Signal Processing (2001) **15**(3), 513–527
doi:10.1006/mssp.2000.1380, available online at http://www.idealibrary.com on IDEAL®

LIFE-CYCLE MANAGEMENT OF INFORMATION AND DECISIONS FOR SYSTEM ANALYSES

YORAM REICH

Department of Solid Mechanics, Materials, and Systems, Faculty of Engineering, Tel Aviv University, Ramat Aviv 69978, Israel. E-mail: yoram@eng.tau.ac.il

(Received 26 June 2000, accepted 30 November 2000)

Quality products require quality analysis; analysis that delivers the necessary behavioural information using the assigned resources in the prescribed time frame. Using analysis tools effectively requires deep understanding of theory and practice. Practical analysis could be supported by managing analysis tools information and their past uses. If managed properly, this information would be beneficial for present and future tool reuse. Following an examination of analysis activities, a model of analysis activities is developed including various aspects that influence it. The model includes all contextual information of analysis. With this information the model could improve analysis as well as prevent serious analysis errors. The model called AML—analysis modelling language—is described precisely with the object-oriented formalism unified modelling language (UML). The most important part of AML, the static structure diagram, is discussed in this paper. AML could be transformed with reasonable effort into an analysis process management system.

1. INTRODUCTION

Product life cycle refers to whatever happens to the product throughout its life starting before inception in *market analysis*, through its *design*, *manufacture*, *usage* (including *maintenance*), and *recycling*. Any system and its associated information evolve through these stages from abstract to concrete. To illustrate, at the end of design, the product definition becomes more precise than initially envisioned, for example by specifying dimensional tolerances; while the information about the product manufacture is still incomplete, for example, dimensional variations are unknown and manufacturing flaws are not yet detected. Similarly, the information about the actual product behaviour is uncertain. For instance, uncertain material structure and properties, imprecise dimensions, uncertain environment, and unknown or uncertain interactions between physical phenomena could contribute to variability in anticipated system behaviour.

When the system is being realised in physical objects, some of its associated information may become more precise such as the final product dimensions as built, while others, such as material structure and properties, remain uncertain. Some parameters, such as friction between components, also change in time.

The evolution of information as product development stages unfold mandates special attention from systems that support design processes. This paper concentrates on systems analysis: the process of estimating systems behaviour. The behaviour of a system deals with its performance under various conditions including handling and operation. Behaviour is determined using analysis methods applied to conceptual models or measurements on scaled physical products or prototypes. Analysis could be performed with simple equations for deriving some values or complex ones such as 3D time-dependent multidisciplinary analysis. The system behaviour as estimated by analysis determines the system fulfilment of its desired functionality which is the initial driver of the product design. Consequently, the importance of product behaviour and functionality makes analysis information and processes prime targets for life-cycle management.

Analysis is also a much less formal concept. When designers interact in product design, their exchange involves a particular aspect of the design that is modelled in their discussion. A question posed by one designer constitutes modelling and the response an analysis. Often, the focus of the discussion or negotiation drifts marking the use of several models which, while possibly loosely connected, are nevertheless invaluable for the negotiation. Therefore, to benefit from past models arising in collaborative processes, the information derived from previous negotiations between designers needs to be maintained [1].

Information not only evolves through development stages but also any piece of information related to product behaviour evolves throughout the product life cycle. Initially, market analysis determines whether a product with the expected functions is viable. Subsequently, in product design, various analyses determine conceptually whether the product meets its intended specifications. In manufacturing, quality assurance determines through measurements whether the manufactured system meets its physical requirements (e.g. tolerances) and how they impact the intended behaviour. Finally, during system operation, sensors can provide data regarding actual system behaviour. All these calculations or measurements are *different evolving views of the same system behaviour*. They must be integrated and managed together.

Not only does the system evolve through its life cycle stages, but so also does its entire context including analyses methods and computer tools, engineers performing analyses, customer requirements, design standards (e.g. environmental), parts availability, and technology. Thus, life-cycle management of analyses information means the management of all these aspects.

The importance of life-cycle management of analyses information cannot be taken lightly; simply excelling in analytical skills does not guarantee that an analysis will be reliable. For example, analysis could fail or be unreliable if the software used is incorrect due to improper testing. Roache [2] describes an example of his own work where he introduced a gross error of a factor 2 into an analysis code. The error was not detected due to partial testing. Following proper testing, the error was corrected and the code produced reliable results without failing on a previously failed problem.

Even if analysis tools are verified and validated, there is still ample room for failures. There could be serious consequences to misuse of analysis or simulation tools. A product could fail completely due to erroneous analysis or simulation arising from mismanagement of the analysis process. As it turns out, organisations that lead technology progress are not immune to such failures. Consider the following examples of space-related system failures.

Misuse of analysis input. The failure (and waste of $193M) of NASA Mars Lander in December 1999 was due to a wrong use of units—the most basic aspect of doing any engineering calculations. The unit mismatch led to erroneous maneuver required to place the Lander in correct Mars orbit. With proper management and tools, such a failure could be easily avoided.

Mismanagement of product development including analysis. The failure of the Ariane 5 rocket in June 96 was caused by a conversion of a 64-bit floating point number to a 16-bit signed integer that failed because the number was greater than could be represented by a 16-bit signed integer. The code in the particular location was not protected from such failure although it was protected in other similar places in the code. The decision not to protect it probably came from using analysis scenarios data from the Ariane 4 model that