Uluslararası İleri Doğa Bilimleri ve Mühendislik Araştırmaları Dergisi
Sayı 7, S. 109-114, 4, 2023
© Telif hakkı IJANSER'e aittir
Araştırma Makalesi

https://as-proceeding.com/index.php/ijanser
ISSN: 2980-0811

International Journal of Advanced Natural Sciences and Engineering Researches
Volume 7, pp. 109-114, 4, 2023
Copyright © 2023 IJANSER
Research Article

Considerations on Utilizing the Object-Oriented Paradigm in the Development of an MRP-based Production Planning System

Mihály Szilárd Avornicului, József Udvaros* and Norbert Forman

Faculty of Finance and Accountancy, Budapest Business School, Hungary

*udvaros.jozsef@uni-bge.hu *Email of the corresponding author*

(Received: 20 April 2023, Accepted: 04 May 2023)

(DOI: 10.59287/ijanser.2023.7.4.564)

(1st International Conference on Scientific and Innovative Studies ICSIS 2023, April 18-20, 2023)

ATIF/REFERENCE: Avornicului, M. S., Udvaros, J. & Forman, N. (2023). Considerations on Utilizing the Object-Oriented Paradigm in the Development of an MRP-based Production Planning System. *International Journal of Advanced Natural Sciences and Engineering Researches*, 7(4), 109-114.

Abstract – Manufacturing companies today face an increasingly dynamic environment. Development of production planning systems requires approaches that offer flexibility in development of solutions. Applying the UML (Unified Modeling Language) and MDA (Model Driven Architecture) paradigm can ensure compatibility between applications developed on different platforms. The MDA approach to software development becomes an obvious choice. In this approach, the models drive the process of software development. These models are defined at different levels of abstraction to represent various aspects of the system. The transformation of models from one level of abstraction to another, or the transformation of models to executable code is performed by using automated transformation tools. The strength of MDA lies in the fact that it is based on widely used industry standards for visualizing, storing and exchanging software designs and models. The models in MDA are abstracted at three different levels – the Computation Independent Model (CIM), the Platform Independent Model (PIM) and the Platform Specific Model (PSM). The key to the success of MDA lies in automated or semi-automated model-to-model and model-to-code transformations. In this paper, we examine how a platform-independent model of an MRP-based production planning system can be created using MDA approach, from which several platform-specific systems can be achieved, as required.

Keywords – UML, MDA, MRP, Production Planning System, System Development, Flexibility

I. Introduction

Production planning lies at the heart of every manufacturing firm, since it provides the information needed to satisfy current and future market demand, facilitating to maintain the balance between supply and demand. Companies today use large integrated systems for production planning, called Enterprise Resource Planning (ERP, web-integrated ERP) systems, but the majority of them is based on the original Material Requirements Planning (MRP) philosophy [1].

MRP-based systems have been widely accepted and used by manufacturing companies in the last three or four decades [2], [3], however, implementation and integration difficulties have been frequently reported, mostly due to the fact that it has to be implemented as a total system which covers most of the business processes of the company [4]. In this paper we attempt to offer a basic modeling tool that can be used for developing a platform-independent MRP-based production planning system.

Since in the past ten years most firms acquired serious software assets, if we want to develop a new system, we have to consider the preservation of existing data. Therefore, we have to seek for a solution that enables data transfer to the newly developed system. The Model Driven Architecture (MDA) aggregates into one framework the previously accepted modeling standards, and creates the possibility to embed subsequent standards. The aim of the MDA standard is to ensure the collaboration of various components at the model level, independently from the implementation technology or platform [5]. The essence of MDA is the definition of a technology-independent model for distributed enterprise IT infrastructure; therefore, it can be successfully applied in the development of an MRP-based system.

II. Materials and Method

A. Material Requirements Planning (MRP)

Material Requirements Planning (MRP) is a computerized information system developed specifically to help manufacturing companies manage market demand, plan inventory and schedule production [6]. As technology and information solution evolved, MRP systems were first transformed into MRP II systems which include greater integration with information in other parts of the organization. MRP/MRP II systems are today incorporated into larger ERP and web-integrated ERP systems [1]. The most important function of MRP-based systems is that it translates market demand into requirements for all subassemblies, components and raw materials needed to produce the required products [7]. Thereby, MRP-based systems facilitate an efficient scheduling of production tasks and assure that production resources are provided in the right amount and at the right time [8].

The central elements of an MRP-based system are (1) the Master Production Schedule (MPS) which converts market demand into product types and quantities that have to be produced and (2) the Bill-of-Materials (BOM), which contains the structure of each product indicating the amount of each type of raw material, component and subassembly needed to produce the product.

B. UML and MDA modeling

An MRP system is complex and sophisticated, thus it is a serious challenge for the developers. Using UML for developing an MRP-based system the following aspects are opportune:

- UML powerful instruments for modelling aspects of behaviour. This way, classes contain both data and the associated processes.
- UML offers a unified vision above grouping different components in the form of packets and also the physical aspect of these packets.
- MRP-based systems are complex and present a dynamic evolution of a higher level than other types of IT systems. It is necessary to use an instrument of analyzing and designing that permits the future extension of the system.
- UML includes mechanisms that permit the easy reuse of the systems components (capsulation, heritage and polymorphism).
- UML permits an integrating point of view for both the data and the process.
- Analyzing and designing UML permits a continuous development of the model.
- Methods of development that call UML are iterative methods guided diagrams of the usage cases. This strategy permits the elimination of programming errors before the application gets to the client. This facts points to a series of advantages like: respecting deadlines, reducing costs of exploit and maintenance and the chances that the system does not satisfy the users needs.
- The user is in contact with the team during the whole period of developing. In case of other methods it was only present at the beginning end at the end of the process.
- It is an expressive language that permits the designer to express shades of the model that would be hard to express in other cases.
- It is an extendible language that permits the continuous adaptation to demands of the domains that it makes models for.

- Despite its flexibility, UML is a precise language with well defined forms.
- Once a usage case is finalised, because the paradigm of object oriented programming the respective subsystem can be launched.

The Model Driven Architecture (MDA) prescribes certain kinds of models to be used, how those models may be prepared and the relationships between different kinds of models. The Model Driven Architecture specifies three viewpoints on a system, a computation independent viewpoint, a platform independent viewpoint and a platform specific viewpoint. The three primary goals of MDA are portability, interoperability and reusability through architectural separation of concerns [9].

In a typical MRP-based solution, in case of traditional design, the PSM model includes conceptual model-based platform-specific details and appears in the form of various descriptions. The system development may be performed starting from the existing environment, so the implementations can be decoded using certain tools.

MDA follows and includes most of the existing industry standard, leading to long-term flexibility of implementations, upgradeability and easy adaptability to subsequent developments [9].

C. Modeling the Production planning System

In the MDA approach, the model contains information about components, structure, behavior and functionality of a particular system.

An efficient MRP-based production planning system needs to fulfill following requirements:

- It has to include a unit that supports the sales and marketing function of the company. This unit next to order entry and billing is also responsible for forecasting product demand based on historical data and order records [10]. This information regarding future market demand is needed to build the Master Production Schedule (MPS).
- It has to include a technological list or list of operations regarding machine types and machine times needed to produce each type of product.
- It has to include a data regarding actual capacity constraints: number of machines, types of capacities and the amount of labor force (working hours) available to produce each type of product [11].
- Evolved MRP II systems also include data regarding the company’s financial system [6]. This enables to include into the system the exact costs of production, including the cost of labor force, cost of operating production systems, cost of capital and other incidental expenses.

The research has been made through a case study, in which we design an MRP-based production planning system, taking in consideration the next steps (Fig. 1):

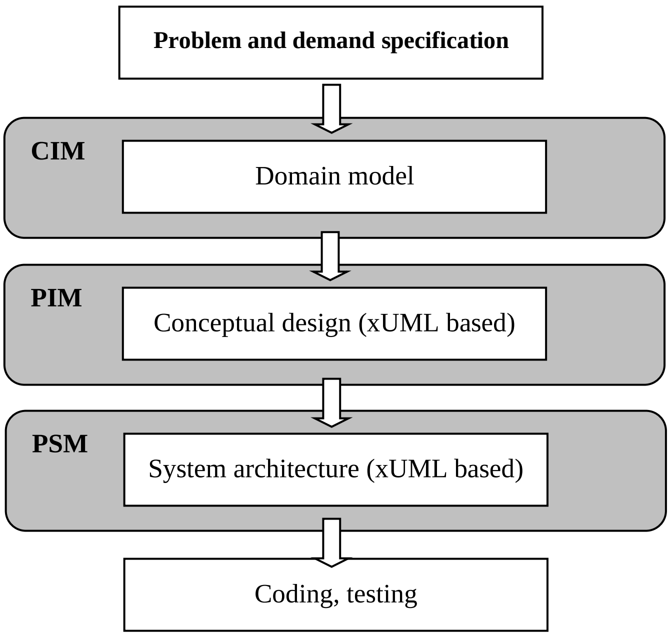
Problem and demand specification
CIM
Domain model
PIM
Conceptual design (xUML based)
PSM
System architecture (xUML based)
Coding, testing

Fig. 1 The concept of the modeling production planning system

As the first step, we create the CIM model of the system, containing the system environment and the requirements. The next step will be the completion of PIM model, which in our case will be xUML-based, and the last step is the transformation of PIM to PSM.

III. Results

A. The CIM model of the production planning system

Like we mentioned before, the first step is to create the CIM model of the system. A CIM is sometimes called domain model, or business model. The CIM plays an important role in bridging the gap between domain experts (or business experts) and software developers. This will be achieved using use-case diagrams.

The UML-based CIM model of the production planning system is shown in the following figure (Fig. 2):

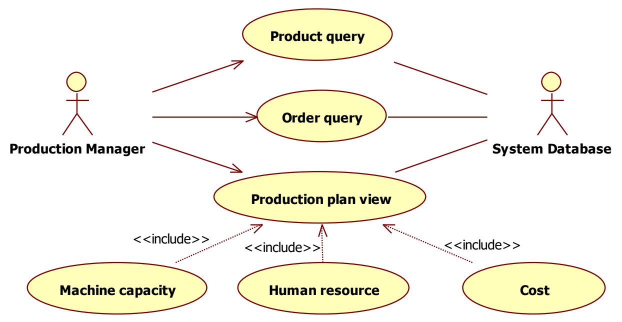

Fig. 2 The CIM model of the production planning system

The specifications of this model are mostly text-based, and are based on user consulted diagrams. The model identifies user requirements, expectations regarding the new system. The CIM and his refinements are a critical part of the development. It is important to be properly completed in order to the developer clearly understand the task. In the model the primary actor is the production manager, the secondary actor is the system database. The production manager should be able to query the products, the orders, and to view the production plan.

B. *The PIM model of the production planning system*

The platform independent model is the first model which actually is created. A PIM describes the construction of a system on an ontological level, meaning that the construction of the system is specified without implementation details. In case of our system, the PIM will describe the actual system. We define the PIM of the system using xUML.

Executable UML (xUML) is a higher level of abstraction layer, which allows us to deploy and to run systems in different environments without any change [12].

In conformity with the concept of MDA, each xUML model is platform independent, therefore the transformation between these models using an operation-specific language denotes a PIM-PIM mapping [5].

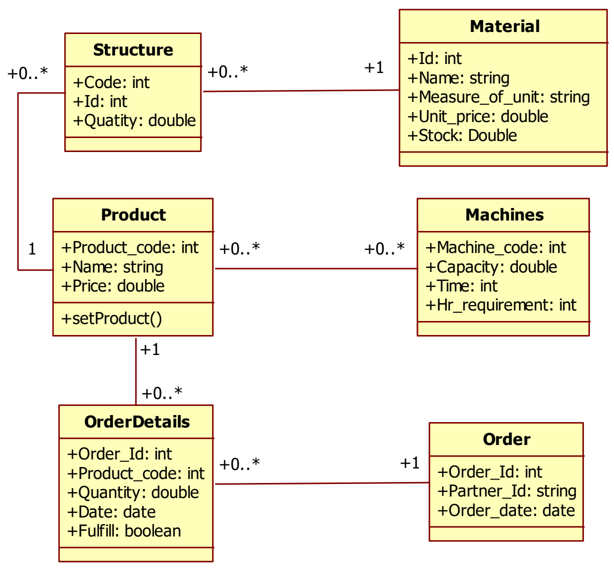

Fig. 3 The xUML-based PIM model of the system

The PIM is a platform independent model and includes the class diagram, which presents the analyzed system in an abstract way. In this model the detailed classes can be seen, where the attributes and methods are specified, as well as the relationships between classes. Based on CIM, we obtain the xUML-based platform-independent model (Fig. 3).

In addition, the multiplicity of the relationship between products and machines, for example, is many to many, as a product can be produced with several machines, and a machine makes several different products. The visibility of the attributes is Public, therefore is indicated with “+”.

C. *The PSM model of the production planning system*

The transformation of PIM to PSM is an important step of MDA paradigm. The number of PSM models depends on the characteristics of architecture. PSM combines the specifications in the PIM with the details that specify how that system uses a particular type of platform [12]. Platform specific elements are added.

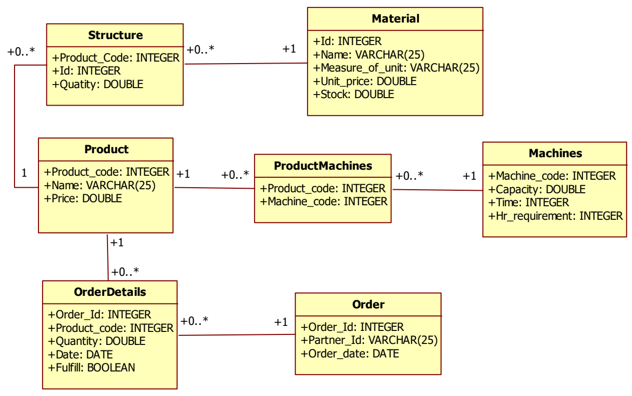

Fig. 4 The PSM model of the system in Oracle

In our case, the classes are converted into tables. The classes' attributes will be the columns of the tables. The types will also be transformed depending on the architecture. For example, the "string" from xUML will be transformed to "VARCHAR" in Oracle. The visibility will also change in the course of PIM-PSM transformation: public will be converted into private. We implement the database in Oracle. As we can see in this figure, the relationship between Products and Machines is decomposed.

IV. DISCUSSION

On the one hand, the MDA standard declares the technology independence of model specifications describing domain-specific functionalities and behaviors, and on the other hand, it ensures interoperability between implementations running on different platforms. When considering the benefits of MDA, it is important to note that:

- architectures following MDA principles achieve interoperability between past, present and future systems,
- MDA's middle-layer scale allows the integration of applications,
- domain-specific applications defined according to MDA can interoperate more widely than before, both on enterprise and cross-enterprise platforms [13].

In fact, MDA is a response to the challenges that network environments (such as MRP systems) with different architectures and systems on different platforms, and constantly changing circumstances pose for users of computing support. The need for interoperability of heterogeneous systems places a number of demands on the technology used, and these must be ensured:

- portability, which increases the complexity of the systems,
- platform independence,
- interoperability between different platforms, and
- efficiency.

The MDA standard can be successfully implemented if the majority of developers start work by designing a technology-independent PIM model rather than writing the source code. This is a key message of MDA [14].

V. CONCLUSION

In this paper we discussed how a flexible MRP-based production planning system can be developed using UML and MDA. First, we created the CIM model of the system, then the platform independent model expressed in xUML, a platform-independent modeling language.

The platform-independent model is subsequently translated to a particular platform-specific model (PSM) by mapping the PIM to Oracle. Following the above-described steps, we can develop systems that more accurately satisfy customers' needs, and that offer more flexibility in system evolution. From the platform-independent model we can produce several platform-specific models.

REFERENCES

[1] N. Slack, S. Chambers, and R. Johnston. (2007) Operations Management, *Prentice Hall*, Pearson Education.

[2] R. Gumaer. (1996) Beyond ERP and MRP II. Optimized planning and synchronized manufacturing, *IIE Solutions*, pp. 32-36.

[3] D. Steele, P. Philipoom, M.Malhotra, and T. Fry. (2005) Comparisons between drum-buffer-rope and material requirements planning: a case study, *International Journal of Production Research*, vol. 43, no. 15, pp. 3181-3208.

[4] R. E. Giachetti. (2009) Design for the entire business: Enterprise architecture has arrived, *IE Industrial Engineer Magazine*, pp. 39-43

[5] M. Raffai. (2005) Az UML 2 modellező nyelv, Szakkönyv Kiadás alatt, *Palatia Kiadó*, Győr

[6] L. Krajewski, L. Ritzman, and M. Malhotra. (2007) Operations management. Processes and value chains, *Prentice Hall*, Pearson Education.

[7] J. Udvaros, N. Forman and S. M. Avornicului. 2023. Agile Storyboard and Software Development Leveraging Smart Contract Technology in Order to Increase Stakeholder Confidence, *Electronics* 12, no. 2: 426. https://doi.org/10.3390/electronics12020426

[8] B. Ram, M. R. Naghshineh-Pour and X. Yu. (2006) Material requirements planning with flexible bills-of-material, *International Journal of Production Research,* vol. 44, no. 2, pp. 399-415.

[9] J. S. Her, H. Yuan and S. D. Kim. (2010) Traceability-centric model-driven object-oriented engineering, *Information & Software Technology*, pp: 845-870

[10] P. Duchessi, C. M. Schaninger, and D. R. Hobbs. (1989) Implementing a Manufacturing Planning and Control Information System, *California Management Review*, vol. 31, pp. 75-90.

[11] R. Stegerean. (2002) Sisteme modern de conducere a producţiei, *Editura Dacia*, Cluj-Napoca

[12] W. Xiening. (2006) Research on MDA Model Drive Architecture based on xUML, *Computer Development & Application*, pp: 46-48

[13] C. Avornicului and M. Avornicului. (2006) Aspects of using MDA paradigm in system development, *Proceeding of International Conference on Business Information Systems*, Iaşi, pp: 378-385

[14] C. Avornicului and M. Avornicului. (2010) Managementul şi proiectarea sistemelor informatice de gestiune, *Editura Risoprint*, Cluj-Napoca