TECHNICAL TRANSACTIONS | **CZASOPISMO TECHNICZNE**
MECHANICS | MECHANIKA

4-M/2016

DARIUSZ KARPISZ*

THE QUALITY OF IMPLEMENTATION OF MANUFACTURING ICT – SELECTED PROBLEMS

JAKOŚĆ IMPLEMENTACJI PRZEMYSŁOWYCH SYSTEMÓW INFORMACYJNYCH – WYBRANE PROBLEMY

A b s t r a c t

The paper presents models for designing of information systems for manufacturing. Chosen model of system implementation has a significant impact on the quality and performance of the target solution. Special focus is on the data model layer and the possibility of changing the standard three-tiered model of information systems through increased utilisation of resources of database server.

Keywords: manufacturing ICT, databases, n-tier models, database-centric architecture

S t r e s z c z e n i e

W artykule przedstawiono modele projektowania systemów informatycznych do zastosowań przemysłowych. Wybrany model implementacyjny ma istotny wpływ na jakość i wydajność docelowego rozwiązania. Szczególnie skupiono się na warstwie modelu danych i możliwości zmiany standardowego modelu trójwarstwowego systemów informacyjnych przez większe wykorzystanie zasobów serwera baz danych.

Słowa kluczowe: przemysłowe systemy informacyjne, bazy danych, model wielowarstwowy, architektura zorientowana na dane

DOI: 10.4467/2353737XCT.16.236.5985

* Ph.D. Eng. Dariusz Karpisz, Institute of Applied Informatics, Faculty of Mechanical Engineering, Cracow University of Technology.

1. Introduction

The specific conditions for using information systems in industrial applications should generate the appropriate design and implementation patterns. The variety of software such as CAD/CAM/CAE causes many difficulties in cooperation among themselves and with other systems such as ERP or CRM. It would seem that over the years of progress of computer science, in the domain such as ICT (Information and Communication Technologies) [1, 2] there were made the right choices in the areas of hardware, software and its integration. The fact is that IT corporations use the standardized models to deliver products to market as soon as possible in order to maximize the ROI (Return of Investment) regardless of different industry requirements. These standard models are correct, however, maximization of performance for targeted ICT solutions is not applicable to them. Presented in article multilayered models of systems are used for rapid software developing for IT companies. This can be achieved due to the possibility of segregating of work and partial code generation, mainly resulting from the application templates and data models without consideration of the specific needs in industry.

2. Methods of implementation ICT in Manufacturing

The main aspect in the ICT systems designing, both single application and composed of separate cooperating programs are multi-layer models.

2.1. *n*-Tier Architecture

Due to solutions for logical separation of information systems there are set single-, two-, three- and multi-tiered models known as *n*-Tier Architecture [3]. All of these models consist of three types of functional modules:

– *presentation module* – translation, presentation of tasks and results to the user (called also as User Interface – UI),
– *logic module* – control of an application functionality and process flow (called as Business Logic),
– *data module* – data persistence mechanisms and data access layer.

Within the single-tier model all tasks associated with storing data and data manipulation, business logic, data presentation and user interface generating are supported on a single computer.

Within the two-tier model, presentation layer and business logic are executed on the client's workstation, and the data is stored in a separate, dedicated database server. In this mode data can be shared by many customers as opposite to single-tier model.

Within the three-tier model there is performed only presentation layer at the Client Machine (usually in the form of access via a Web browser or Java application) while the main functionality is supported by the Application Server. In this mid-range layer there is implemented full business logic along with application layer access to data stored on a dedicated server – Database Server with Database Management System (DBMS).

Progress in Object-Oriented Programming (OOP) drove a significant acceleration of the development of applications as well as expansion of business logic and design of application based on functional model. An integral part in this approach implemented in ntiers model is Object-Relational Mapping (ORM) mechanism to connect classic relational data model [4] (or other data models [5]) with class structure of OOP application. Using ORM and OOP moved data oriented design to the business logic design.

2.2. Architecture of Manufacturing Systems

Manufacturing Systems are most expanded domain of ICT. In fact, most of the software and hardware is part of a Three-tiered model however much more advanced. On Fig. 1 is a model example of ICT used in the preparation of production. Classic CAM software can be treated as One-tired model excluding license servers. In more advanced systems there is often used Product Lifecycle Management software (PLM) which acts as an intermediate layer in access to detailed data that is available on the Database Server. There are also used specialized databases such as materials databases used in Simulation-Based Engineering [6]. Additional software to support the design and production planning [7] and quality management [8] are used in specific areas of manufacturing. Such software use normally the same shared database server to support thematic databases.

Local or central databases are used in all of the modules of complex industrial systems. Despite this fact, the designing of systems following the bottom-up concept from data model to UI is not popular.

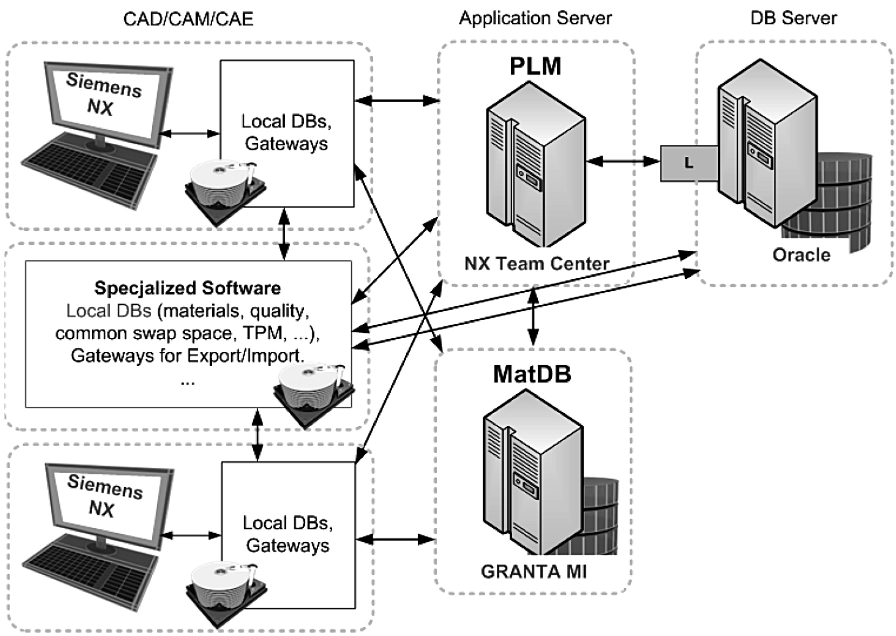

Fig. 1. Example application model using in CAM/CAE systems

2.3. The Problem: Fast Implementation or Quality/Performance

Figure 2 provides an overview on the design, implementation and application of information systems. In the three-tier model the highest investments are incurred for database servers and high-speed dedicated storage. It is therefore always a layer with the highest efficiency. The presentation layer in the form of hardware and software supporting e.g. WWW interface is the cheapest part of the investment and potentially of the smallest performance.

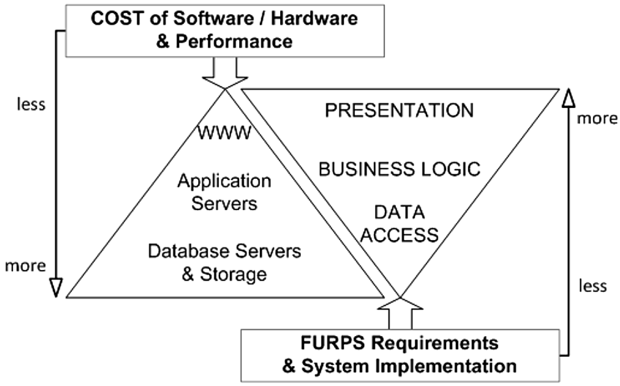

Fig. 2. Dependencies between Architecture of ICT and Implementation Process

Fast alignment of the product compliance with client's expectations, usually defined as a set of functional requirements (Functionality in FURPS Method) is delivered due to enhanced UI and advanced business logic. The extensive use of OOP and ORM techniques results in insufficiency within the definition of the data model. The vast majority of requirements is then moved to the presentation services with low performance hardware instead of using User Defined Functions (UDFs) which are executed in more powerful servers and storage systems. It seems obvious that such an approach is incorrect.

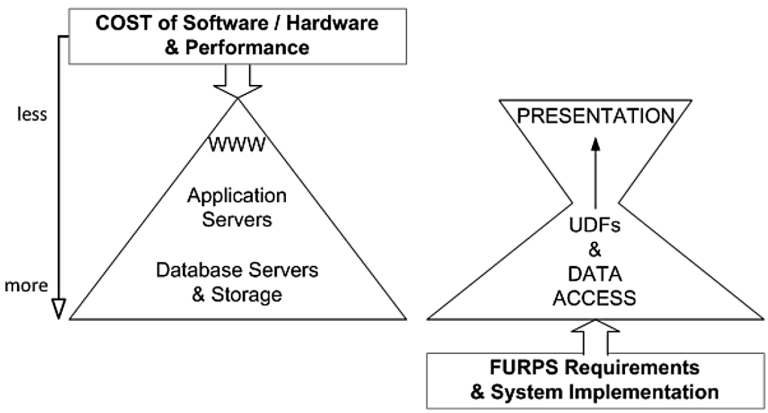

Fig. 3. Database-centric Architecture used for enhanced utilization of resources

Due to use of solutions such as trigerrs and UDFs it is possible to move a number of mechanisms implemented in the presentation layer and the business logic directly to database servers for execution (Fig. 3). Given approach is known as Database-centric Architecture [9] allowing better utilization of resources.

3. Solutions and Results

Based on materials database example (Fig. 4) consisting of a standard set of tables Materials and Factors remaining in the multi-value relationships self-organizing database can be built. As previously mentioned, software such as CAD/CAM/CAE use a local databases e.g. tools, tooling systems, materials, configurations. Below is sample description of material in the local database for Siemens NX:

```
DATA|MAT0_00266|7050|ALUMINUM|75-150 HB|ALUMINUM ALLOYS, WROUGHT -
```

Normally, in order to supply the system CAM with real data, it is required to manually edit the database files, use the GUI or use PLM. Additional software layer is also used within the application server to synchronize data. On the other angle, having ability to use specialised systems for production support [5] it may be worth considering adding to the data model procedures, which will automatically generate code row for CAM.

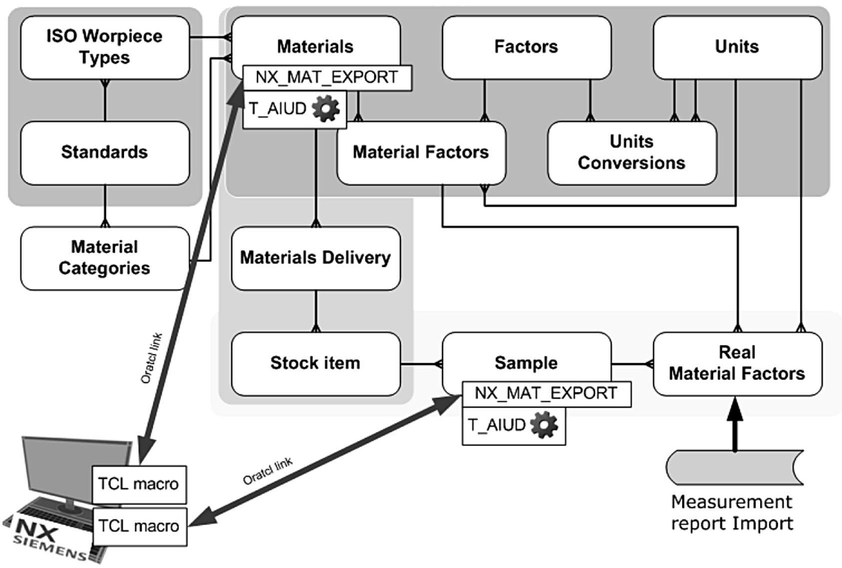

Fig. 4. Example of materials database using to auto-generate CAM material database row

Fig. 4 presents a table with *Materials* and *Samples* (of materials) with added column *NX_MAT_EXPORT* to store Siemens NX row with material description. The *T_AIUD* is a trigger of After Insert or Update, or Delete type which automatically responds to data editing and inserts text for *NX_MAT_EXPORT* column. It is the most advanced model for data preparation which is to be exported to other systems outside the database server. Data prepared in this way does not require any additional software to its translation, but only macroprocedures within CAM workstation (e.g. as a script in Tool Command Language – TCL). In this case, the database connection can be implemented directly by the Oratcl library and is invisible to the user.

In this paragraph there will be presentation of transition from the classical approach to Database-centric Architecture base on advanced triggers and UDFs mechanisms. The base query is a request from Application Server for a preparation of specific material row for the NX System:

```
select * form MATERIAL where MAT_NX_MATCODE = 'MAT9_00500';
```

The most powerful solutions of this problem is the use of UDF:

```
select NX_MAT_EXPORT('MAT9_00500') from dual;
```

or the specific trigger *T_AIUD* executed in *NX_MAT_EXPORT* column:

```
select NX_MAT_EXPORT from NX_MAT_EXPORT
where MAT_NX_MATCODE = 'MAT9_00500';
```

The final result:

```
DATA|MAT9_00500|500T|ARMOX STEEL|480-540|High Hardness Armor Alloing Steel
```

The last solution requires only reading of the final row for export to the CAM workstation. Data preparation can be several times faster than in the case of translating raw data on the application server. In all cases presented applications use standard request of data – SQL query. There is no need for any additional techniques for calling functions type of UDFs from the applications connecting to the database. In fact, every procedure or function can be requested in form of query in advanced database servers like Oracle so the transfer of part of the functionality associated with parsing, checking whether the translation of data for export to a variety of systems that do not require changes in the way the programming and data retrieval.

References

[1] U.S. Access Board, *Information and Communication Technology (ICT) Standards and Guidelines*, Notice of Proposed Rulemaking, 36 CFR Parts 1193&1194, United States Access Board, 2015.

[2] Tongia R., Subrahmanian E., Arunachalam V. S., *Information and Communications Technology for Sustainable Development, Defining a Global Research Agenda*, Allied Publishers, Bangalore, 2005, 19-41.

[3] Fowler M., *Patterns of Enterprise Application Architecture*, Addison Wesley, 2002.

[4] Codd E.F., *A Relational Model of Data for Large Shared Data Banks*, Communications of the ACM, Vol. 13, No. 6, 1970.

[5] Karpisz D., *Relational Data Model is Not Only One*, Technical Transactions, Vol. 4-M1, 2011, 187-194.

[6] Stojek M., Pietraszek J., *Simulation-Based Engineering Science Challenges of the 21st Century*, Applied Mechanics and Materials, Vol. 712, 2015, 3-8.

[7] Gawlik J., Kiełbus A., Karpisz D., *Application of an Integrated Database System for Processing Difficult Materials*, Solid State Phenomena, Vol. 223, 2015, 35-45.

[8] Gawlik J., Kiełbus A., *Application of artificial intelligence in supervising the technological equipments and quality of products*, The Practice of Quality Management of the 21st, Wydawnictwo Naukowe PTTŻ, Kraków 2012.
[9] Babers C., *The enterprise architecture sourcebook*, Vol. 2, Data-Centric Architectures, El Paso 2006.