

ASPECTS REGARDING THE INTEGRATION OF THE MODERN INFORMATION SYSTEMS

Botezatu Cezar

The Romanian-American University, Faculty of Internal and International Commercial and Financial-Banking Relations

Carutasu George

The Romanian-American University, Faculty of Computer Science in Business Management

Botezatu Cornelia Paulina

The Romanian-American University, Faculty of Computer Science in Business Management

Developing an information system integration strategy, at enterprise level is one of the most difficult issues that the information systems managers must solve, within the present conditions. When settling the integration strategy, two aspects must mainly be taken into account: assuring a complete integration solution of the information systems and building an integration infrastructure of the applications.

The present paper aims at presenting few aspects regarding the integration of the information systems at organizational level and describing a specifically developed IBM architecture.

information systems integration, software modelling, Model Driven Architecture (MDA), Model Driven Development (MDD)

M15

1. The necessity to integrate the present information systems

The success of the business depends, mostly, on the quality and quantity of the information that it relies on. The internet is a new form of environment, which modifies both the nature of the business and the market, transforming the relations between the two. Technically, all long-term business in the real world is nowadays actively present on Web. Many of them have entirely transferred their operations on Internet. The information systems have become more and more complex; therefore it is compulsory to integrate them within the already existing software applications and to adjust them to the new actual conditions. The IT solutions were rapidly created, the attention of the software producers focusing on the development of high quality instruments for the integration of the information systems. Therefore, presently, in most of the companies, the process of developing the software solutions is no longer a unique process, based on costs, but it became a part of the company global strategy, also requiring the integration of the existing applications.

A viable solution for integrating the information systems within an environment which is continually developing must be based on the informational architecture at enterprise level, or most of the time, department level. This is how the information system integration can be accomplished, not only at technical infrastructure level, but also at the level of business concept, which are more stable in time [1].

OMG Group (Object Management Group) has been pursuing, for 12 years, the integration of the new developed applications, within the already implemented ones. Taking into account the dynamic transformation of the IT technologies, the growing demand of soft application integration, the new information online systems and new e-business patterns, a new standard based on integration and interoperability is required. Presently there are several operation systems, different programming languages, multiple networks architectures, which are rapidly developing and in order to keep up with them, they must resort to the modeling process and the increase of the abstracting degree within the software development process.

Each system was based exclusively on accomplishing the functions it was designed for. Nowadays, when transferring data from one system to another, the users have to re-enter the data or to manually transfer the information.

2. Integration solutions

One of the used solutions was settling some connections between some specific systems, which prove to be connected. This connection though, cannot be easily achieved and maintained for a great number of systems, at enterprise level. The purpose of the integration is to make these systems work within one single logic system. The integration of the applications within different departments or enterprises can be achieved on three directions: vertical, horizontal, mixed.

The vertical integration consists in accomplishing adaptive services that would connect the present applications and existing data, with new applications, new platforms or services; for instance, connecting an existing insurance system, to Web services.

The horizontal integration consists in building adaptive services that would connect already existing applications and data among them. For instance, integrating an on-line acquire system, with an accounting system. The integration can be achieved by using application integration instruments at enterprise level. (Enterprise Application Integration – EAI)

The mixed integration consists in integrating both existing and new applications.

The integration consists in integrating the applications and also data, by using adequate security standards and methods.

The integration of data represents the integration at the level of database, either by migration of the data from one system to another, or by creating common data warehouses which would be used by several systems. The integration consists in extraction, transportation, transformation and validation of the data from the source systems and loading the data in the destination systems or data common warehouses.

The integration of applications represents a strategic approach in order to unify several systems, at the level of both services and data. The user's way of interacting with the systems and the integrated application is important in this case. These architectures also include Web parts, the Internet access being already well known, they contain connections with the data warehouses and relations with other enterprises.

3. Software modeling – integration technology focused on models

Presently, most of the software developers are facing the problem of integrating a great number of software applications, written in different programming languages. These applications are designed to run on different software platforms, operating systems, to use different databases, being adapted to different network protocols or middleware platforms. This is the reason why OMG Group has been trying to simplify the issue of applications integration, by developing the new MDA architecture – based on patterns - and which lays the foundation of UML language.

By modeling the information systems, the software developers can create the system models before engaging additional resources; they can design software applications starting from requests and can apply the iterative development, where the obtained models have a high abstracting level. Modeling complex software applications implies therefore a number of benefits, but it only represents a part of the software development and integration process. The models oriented approach of the software development process was possible due to adopting some standards such as UML language, models representation language. Considering that the applications development process has become more and more complex, the solution based on increasing its abstracting level was compulsory.

4. Model Driven Architecture (MDA) – the methodology of modern software applications integration and interoperability

The new MDA approach is focused on integrating the software applications belonging to the beneficiary companies; by using the UML, the soft developers can model interface and connections between applications, irrespective of the software/hardware platform, programming language or

network type. The new MDA approach is based on the UML language and works at a high abstracting level, in order to provide a universal valid software applications integration mode.

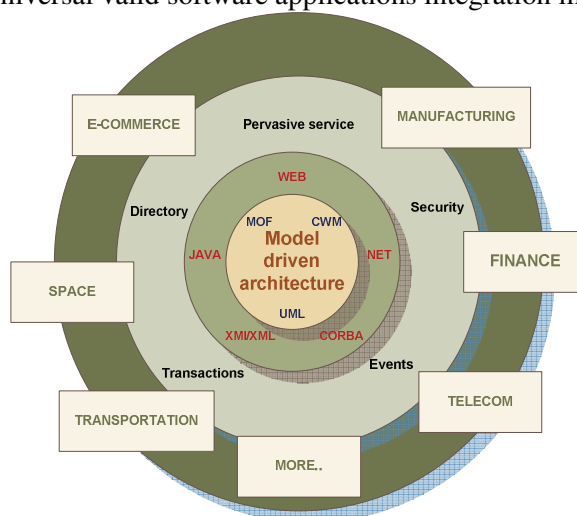


Figure 1. Model Driven Architecture [2]

This architecture can be graphically represented by a number of concentric circles (figure 1), in the middle of which lay the UML language and the MOF models (Meta-Object Facility). The purpose is creating a central, fully integrated model, based on which different applicants would be generated, in different programming languages, in order to run on different hardware or software platforms. It can be therefore obtained the integration of all software applications already existing in the beneficiary company and the clients and the fulfillment of the demands regarding the simplification of the developed IT system infrastructure. The MDA is not just another CASE instrument of generating a complete application, starting from the UML diagrams, but it consists in generating complete connections, for integrating software applications through UML diagrams. It is like a UML integration model, at high level, that could generate: connection bridges, protocols, relations, etc., necessary for the integration of a new application, with the already implemented, inherited ones.

The ideas laying the foundations of this new challenge are the following:

- Software solutions at a company level are considered a collection of incapsulated services communicating through their own interfaces;
- Reiteration of the approach way takes place at all levels, during the entire software project, starting with modeling the business environment, the business processes and ending with the implementation.

Therefore, the MDA architecture separates the functional demands of the system, from the implementation method, resulting thus an independent architecture of programming language, hard/soft platform or IT producers [3]. The MDA architecture can be applied during the entire process of development of the IT systems, using open standards. It allows the modeling of the IT systems, based on the functions starting from programming languages, hard/soft platforms, IT technologies, and this means that a system created through the MDA approach can be modified or extended at any time, without modifying the core infrastructure. The MDA approach facilitates the integration process, decreases the information systems development time and preserves the company's resources, enabling the accomplishing of a greater number of IT solutions without the increase of the necessary time or of the number of people involved.

The MDA architecture has three views (figure 2):

- *Computation Independent Viewpoint* – *CIV* – which separates the logical modeling of the IT system from the implementation specifications;

-*Platform Independent Viewpoint – PIV* – which focuses on the operation modeling of the system, without detailing the implementation specifications;

-*Platform Specific Viewpoint – PSV* – which contains the implementation specifications (hardware platform, software platform, middleware products, used IT technologies).

The true value of the MDA approach consists in the fact that the CIM view can become PIM and the latter one can become the PSV view, which can be directly translated into a source code. Its key-elements are the mapping between the three views and the software instruments allowing these mappings.

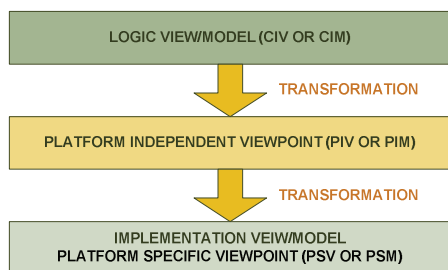


Fig.2. Views of the Model Driven Architecture (MDA) approach [4]

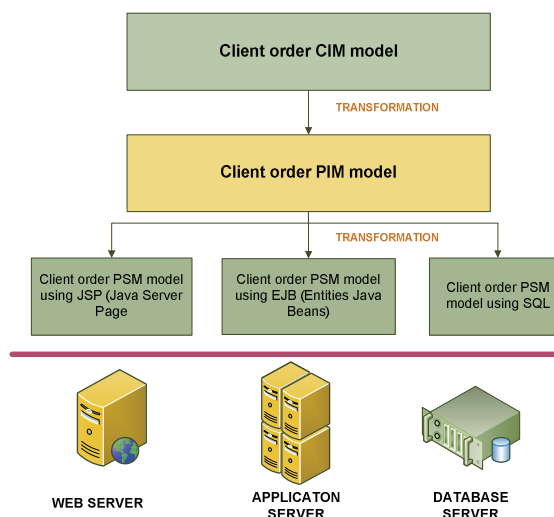


Fig.3. Transformation of CIM-PIM-PSV for the clients' orders

If integrating the clients' orders sent via Internet, within a company's information system, is being considered an issue, the CIM view shall consist in UML diagrams at a conceptual level that will show what the information system will specifically do; the PIM view is next and it describes its functions and structure through detailed UML diagrams and finally comes the PSV view, actually several such views (figure 3).

The company applications are focusing, more and more towards clients, products and services, dealing with the daily specific transactions. Considering that the business regulations are rapidly changing in a modern business environment, the information systems must be updated mostly by incorporating the new IT technologies. The MDA approach [5] is suitable for such applications, due to the benefits: portability, interoperability towards the platform, productivity, quality, fast process of developing the software applications, of integrating the modern IT technologies and the independence towards the imposed hard/soft platform, the decrease of costs and time connected to the software development process.

5. Soft instruments used with the MDA methodology

Once the new MDA standard was introduced, a new implementation instruments market was created. Considering that the MDA approach is based on the separation of the information system architecture in three different models (CIM, PIM and PSV), the chosen instruments must put up with the same separation, and not only the generating of source-code from UML diagrams [6].

The Model Driven Development (MDD) is a new method of designing the information systems and is applicable with the MDA methodology, which allows the building of an ensemble of models of the system to be analyzed and of the new system, based on which other models or the system's source-code

are being generated. Practically, everything focuses on transforming the models of the system to be achieved, and generating the source-code. This method needs an integrated development environment (IDE) that would support: UML language, patterns, transformation of UML models and source-code generation. Rational Software Architect (RSA) is such an integrated instrument for designing and developing information systems.

By going from analyzing to designing, for instance, the analyses model becomes a design model, the documentation connected to different models is created (e.g. the instrument Rational Software Architect's Report Generator or IBM Rational SoDA - IBM Rational Software Documentation Automation), the testing models are obtained (by applying JUnit), the building and development of scripts, the application of patterns.

The next image shows how an actual problem becomes the adequate IT solution, by applying the MDD method. (figure 4).

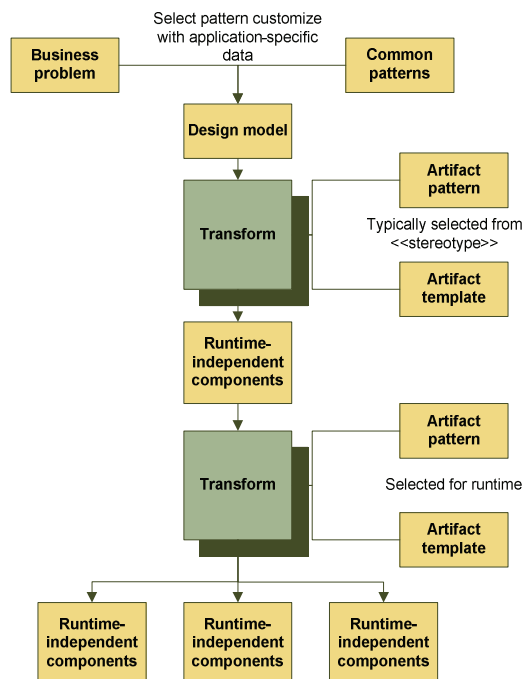


Figure 4 MDD method for transforming an actual problem, into an IT solution [7]

Starting from the clear separation between the logic models and the implementation ones, we can actually speak about a layered modeling technique, which allows the achieving of the analysis, design, and implementation model. The three models can be defined as follows: *the analysis model*—contains the utilization cases, *the design model* – includes the architecture of the system to be achieved, independently from implementation specifications, *the implementation model* – contains the implementation specifications.

Just as the architecture of a system is traditionally defined by 4+1 different perspectives (the centrally located utilization cases view, the logic view, the implementation view, the distribution and exploitation view), the *utilization cases view*, in MDD method, is also centrally located, alongside with the *implementation view*, which fits the structuring of the programs forming the system in components, *the logic view*, which describes the functional requirements of the system, *the distribution view*, which defines the space aspect of the system (hardware equipments, network nodes) and the *operational view* or the processes' one, which fits the structure of exploiting the programs and executable components. By viewing the utilization cases, the connections users-services and services-IT technologies are being accomplished.

Image 5 reveals the connection between the business processes sphere and the sphere of the IT technologies, achieved by services modeling. The beneficiaries look at the information system through the logic view and the utilization cases, while the IT specialists build and maintain the abstracting level between services and implementation IT technologies, through the implementation and distribution views [8].

The MDD method allows the logical modeling of the software applications integration, independently from the IT technologies used for their implementation [9].

Components specific to the patterns applying come in addition to the architecture of the RSA integrated environment (figure 6). We can notice the presence of a functional asset repository, obtained starting from models, applying patterns and eventually transformations, their saving into this repository, followed by their re-utilization, taking into account the demands [10].

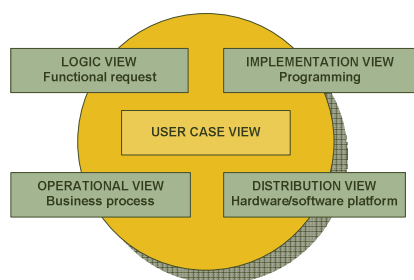


Figure 5 perspectives 4+1 of the IT system architecture in MDD method

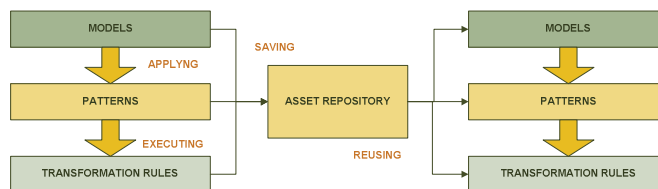


Figure 6 The components of RSA environment, specific to patterns appliance [11]

6. Conclusions:

The integration of the applications/IT systems represents a very up-to-date and complex issue, becoming a part of the global management strategy of an organization [12]. Within this context, the IT specialists are becoming more and more involved with the management process, they explore and offer solutions of software applications development and integration, the flux of the resulted data and processes providing a strategic advantage for the enterprises: the possibility to carry on a business in real time, in an environment based on events, with decreased response time, re-using the IT components under optimal conditions. This is why new specific IT techniques and technologies and new adequate Case instruments emerge. Some of these issues have been analyzed by authors and solutions have been suggested within a research project called Development and Implementation of the Integrated Management System (DI-IMS).

7. Bibliography

- [1] Lăzărică M., Lungu I., *The Impact of Electronic Commerce on firm's Business Models and Market Structure*, Conferința internațională de Comerț 2006, ASE București, Facultatea de Comerț, 27-29 martie 2006, pag.667-673, ISBN 10 973-594-785-4, 13 978-973-594-785-9
- [2] Paul Harmon, *MDA: An architecture for the e-business era*, 01 June 2001, <http://www.ibm.com/developerworks/webservices/library/co-omg/index.html>
- [3] Brown Allan, *An introduction to Model Driven Architecture Part I: MDA and today's systems*, 17 february 2004, <http://www.ibm.com/developerworks/rational/library/apr05/brown/index.html>
- [4] Kontio Mikko, *Architectural manifesto: MDA for the enterprise. An architect's approach to more productive development*, 29 iulie 2005, <http://www.ibm.com/developerworks/library/ar-archman10/>
- [5] Soley Richard and OMG Staff Strategy Group, *Model Driven Architecture*, november 2000, www.omg.org

- [6] Kontio Mikko, *Architectural manifesto: Choosing MDA tools. Three categories for evaluation*, 13 sep 2005, <http://www.ibm.com/developerworks/webservices/library/wi-arch18.html>
- [7] Colin Yu, *Model-driven and pattern-based development using Rational Software Architect, Part 1: Overview of the model-driven development paradigm with patterns*, 21 noiembrie 2006, http://www.ibm.com/developerworks/rational/library/06/1121_yu/
- [8] Jason Bloomberg, *The role of the service-oriented architect*, 2003, <http://www.oasis-open.org/committees/download.php/17568/06-04-00004.000.pdf>
- [9] Peter Swithinbank, Mandy Chessell, *Patterns: Model-Driven Development Using IBM Rational Software Architect, IBM Redbooks*, 2005, <http://www.redbooks.ibm.com/abstracts/sg247105.html>
- [10] Pîrnău Mironela, *The Importance of Web Services using the RPC and Rest architecture*, „International Conference on Computer Technology and Development 2009”, IEEE
- [11] Colin Yu, *Model-driven and pattern-based development using Rational Software Architect, Part 2: Model-driven development tooling support in IBM Rational Software Architect*, ianuarie 2007, http://www.ibm.com/developerworks/rational/library/06/1121_yu/
- [12] Botezatu Cezar, Botezatu Cornelia Paulina, Carutasu George, Joita Alin, *Software Integration – Necessity for integrated management system*, The 20th International Daaam Symposium „Intelligent Manufacturing&Automation: Theory, Practice&Education”, 25-28 nov 2009, Vienna, Austria