Evolutionary Student Research Projects in Domain Specific Modelling for an ERP-System with ADOxx

Frank Wolff[1],
Kai Bieler[2],

[1] DHBW-Mannheim, Information Science, Coblitzallee. 1-9,
68163 Mannheim, Germany
{frank wolff@dhbw-mannheim.de}
[2] L&I IT-solutions GmbH & Co. KG, Brodmühlweg 4,
85049 Ingolstadt, Germany
{kai.bieler@law-ict.de}

Abstract. This paper describes a course project to develop a model-based configuration interface for an open source ERP-system. The analysis of the requirements advocates for a domain specific modelling language. The project incorporates technical and methodological issues. Modelling to directly change an IT-system must be very well adjusted to details of the impacted system. Most prominent for the technical aspect is the implementation of the interface between the involved systems. Further experiences are gathered in regard of the combination of demanding requirements and the peculiarities of typical student teams. In the context of this research, a stepwise approach is pursued, and for the first prototype the comprehensive goals are focussed on some essential analysis and development steps.

Keywords: Student project, Meta-modelling, ERP-system, Model-driven system-configuration, Domain specific modelling language

1 Introduction

This paper embraces two main topics. Base topic is the idea to support the configuration of an ERP-System with appropriate enterprise models. This sets the goal and focuses the underlying project. The other topic is the process of this research oriented development in a one-year project course at a university, to be followed up by further similar projects in future.

Education at universities must serve a number of diverging needs. It should help students to learn important concepts in their field. Often this knowledge is regarded as the primary issue. [1] Anyway, in later phases of education also ventures, which are oriented towards the creation of new insights and artefacts, are part of the studies.

This paper describes a course project, which combines more sophisticated educative goals to foster innovation and thereby research in the area of model-support for ERP-system configuration. It reflects typical aspects of innovation projects which usually entail the cooperation of different groups stemming from or connected with diverse domains [2]. Further, this scenario implicates dispersed knowledge, and in consequence specific problems due to the crucial role of required communication and learning activities in complex innovation processes [3].

The challenges of necessary learning in this case are especially demanding as the involved IT-systems are not examined on a concrete level for a single company but on a general level for diverse companies utilizing ERP-systems.

The ambitious goals, and the capabilities and capacity of a university course demand for an evolutionary approach. A stepwise proceeding should at first identify some interrelated key issues for one major area. The issues identified are subsequently analysed and then one or more potential solutions will be developed and tested. These concepts will then be taken as base to solve further challenges. In subsequent steps, the first results will be tested and re-evaluated in the light of new ideas, resulting in a reflective research approach [4].[1]

2 Course projects at business informatics at DHBW

The context of the reported educative research project is a course at the cooperative state university in Baden-Württemberg (DHBW) in business informatics in the 3rd year of bachelor studies. The course project is part of the fundamental curriculum of the faculty. A typical project is divided in 2 phases with 3 months each and is realized by a group of around 30 students. Intention of this course is to practice a collaborative action with subdivided specialized groups aimed at a joint goal. Usually the course sets up an organization with project management and teams for different topics. Similar to projects in practice, some groups focus more on functional aspects and other groups more on technical issues.

In most cases first the requirements must be gathered and substantiated by the teams. Then a solution is proposed to some principals often the lecturers. In the course of the development, it is common that the projects encounter situations where decisions for a suitable solution are necessary, or even some former ideas must be revised. This often is not easy, as preconceived ideas must be adapted, but it is very important aspect of learning (comp. [5]). In the end of the project, intensive testing is required to validate, that the teams have been working together and that a feasible solution has been found.

Until now these course projects were typically oriented towards development projects, e.g. 1) a *management system for an apartment lease company*, 2) a *pizza delivery service* or 3) a system to *broker and ease the communication on topics for bachelor theses of students*. Also other projects have been completed, e.g. a few years ago a group of students analysed an extensive communication standard of partners in

[1] The authors are grateful to the students of the course WWI15AMB at DHBW Mannheim for their active participation and work in this research project. Also, the authors thank the reviewers for their valuable comments on an earlier version of this paper.

the German insurance industry (GDV interface specification) and proposed a restructured and improved version based on XML. So previous projects focussed on applying mostly available knowledge, while the reported project incorporates numerous unfamiliar and innovative aspects.

3 Models and ERP-systems

For some time, it has been proposed that models support companies to use ERP-systems easier and better. Also in practice, extensive process and data models have been published for some ERP-systems [6].

Enterprise models provide information about many important issues for stakeholders around an ERP-System. This information is crucial and dear due to main characteristics of ERP-systems: its large number of functions, its close integration and its central position for many business processes [7].

ERP-systems comprise vital functionality for companies, usually in the domains of sales, logistics, accounting and human resources. The challenge for a profound understanding of ERP-systems is increased by the fact, that the functions contained, themselves are composed of numerous sophisticated features. The integration of the diverse functions elevates the demands on required knowledge. On the other hand, the integration is the foundation for the high potential of ERP-systems to support smooth and effective operations, to supply with reliable information, and by that to achieve a high economic performance of a company ([8], [9]).

This research on model-based ERP-configuration and the corresponding project is inspired by Frank and Strecker [10]. In their research report the authors envision an ERP-system which is represented by models, in a way, that the models depict main relevant information for all stakeholders, and that a change in the models is reflected by associated adaptions of the ERP-system. At first glance, this may appear to be simple. Nevertheless, the realization of these ideas requires a close analysis of an ERP-system and a respective customization of appropriate modelling languages and modelling tools.

4 Requirements for model-driven configuration of an ERP-system

In preparation for the student project, typical enterprise modelling perspectives and their respective model types of comprehensive enterprise modelling methods were compared with corresponding required information for the configuration of ERP-systems. Table 1. contains a fundamental overview on the examined information. It also categorizes the content into different grades of difficulty and priority for the first project phases. The difficulty is closely connected to the standardization and the variety of their interrelationships to other content. The priority is based on the most prominent target of models for ERP-system the support for organizational and business process purposes.

Table 1. Interesting content for transfer.

Content category	*Difficulty*	*Priority*
Organizational unit	low	high
Role	low	high
Employee	low	high
Workflow (with elements contained)	high	high
Data structure	medium	medium
Product	low	low
IT and other machine resource	medium	low
Configuration data	medium	medium

Besides different content, an integrated system must also consider different modes of interaction, which have an enormous impact on a) *the ease of use* but also on b) *technical requirements on a respective interface*. The following list of interaction (Table 2) resulted from discussions during the first project. As many fundamental issues were open at the beginning the main focus had been on a prototype to create the first elements in the ERP-system from new model information in a batch transfer.

Table 2. Options for interaction between Modelling- and ERP-system.

Interaction	*Complexity*	*Priority*	*Comment*
Creation (Model → ERP in batch)	low	high	basic feature
Deletion (Model → ERP in batch)	low to medium	medium	requires exchange of DB-keys
Direct synchronization (Model → ERP)	medium	low	long-term goal
Bidirectional synchronization	high	low	visionary goal
Error handling	diverse	medium	all stages varying in difficulty

The data structures of the configuration data are fundamental to devise appropriate mechanisms and languages. So the format and naming of fields and classes must be identified and also the relations between different elements must be determined. At first a basic transfer will be helpful to establish a deeper understanding of the functions and underlying principles. Later further interactions and levels of complexity can be approached, after a base interconnection has been established.

Generally, some kind of error handling will be necessary. In the case of prototypes, it might be embedded in preparatory procedures, but in later versions it must be incorporated in dedicated algorithms. Nevertheless, a balance must be found for these procedures and algorithms, how to implement these checks: a) *convenient* for the users, b) *economically to design* for the developer and c) *most effective as possible* for secure operations.

5 Selection of modelling- and ERP -systems

A generic approach towards modelling for ERP-systems would be best theoretically. But the initial analysis for a practical support to configure an ERP-system soon revealed differing specific structures and mechanisms in diverse ERP-systems. These differences are due to the individual adoption to domains or historic decisions during the development of the respective ERP-system. However, many aspects can be represented in standard EM modelling languages, but for more specific properties a representation with a domain specific modelling language or domain specific extensions to a standard modelling language is considered mandatory.

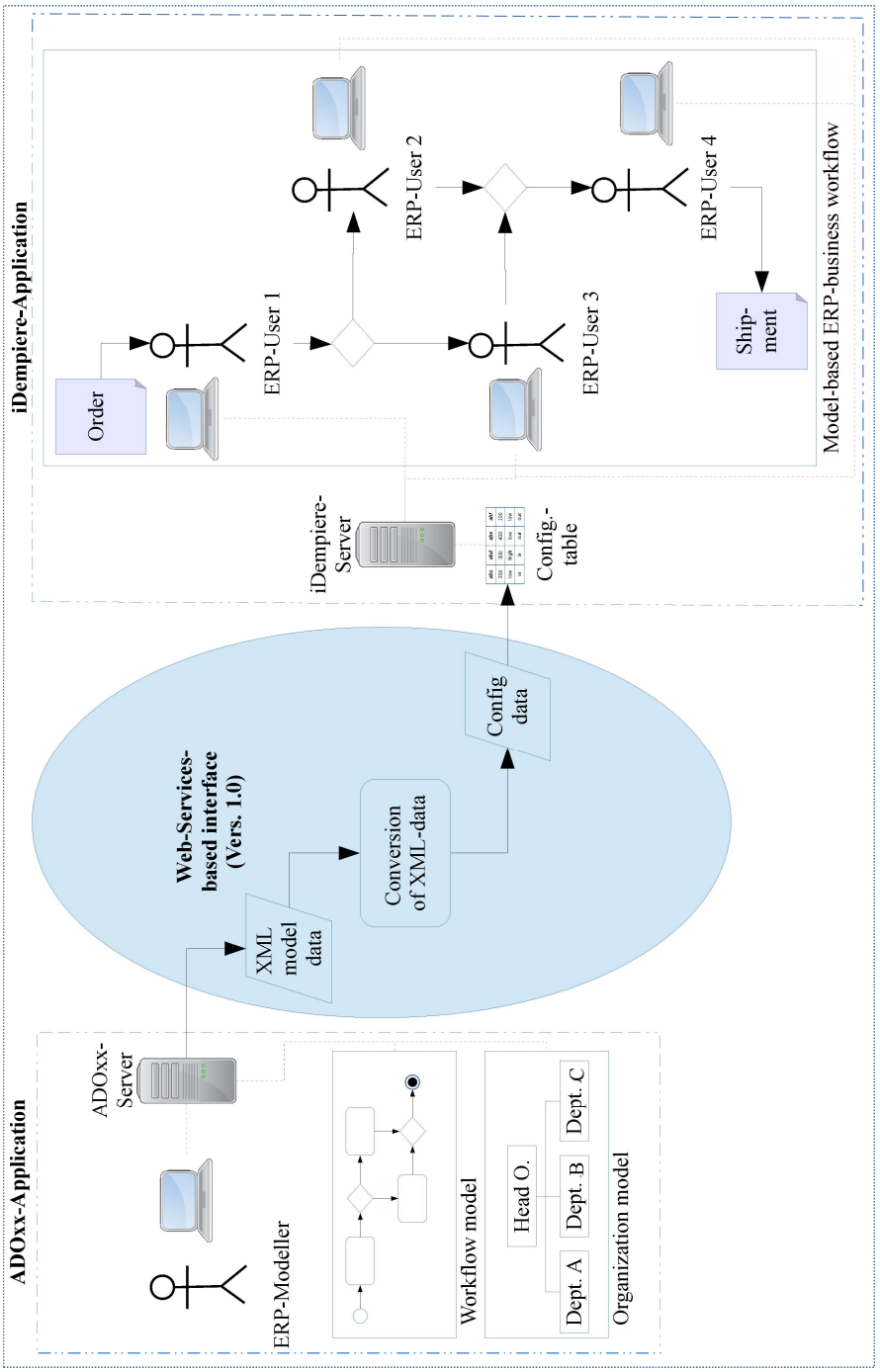
Due to the high complexity of the underlying ERP- and modelling systems, it was decided to concentrate on system instances for the ERP-system and a meta-modelling framework in an exploratory approach. The selection of the systems for this study was guided by their functionality and availability for this project. As the benefits of models would be most appreciated in a sophisticated and demanding environment, the functionality of the ERP-system had to be quite flexible, also for larger and international companies, and comprise adaptable workflows. Similarly, the models would serve the ERP-system configuration best if they represent also complex artefacts in a meaningful and appealing way. Besides other technical considerations, this was a major requirement for the modelling system.

As a meta modelling framework ADOxx was chosen. It has a proven record for being flexible in: a) its *definition of high quality graphics*, b) *adaptive incorporation of sophisticated symbols* and last but not least c) a *set of common interfaces*. It is offered openly to the research and educative community through the Open Modelling Initiative. [11]

There are numerous ERP-systems that could potentially be enhanced for configuration with modelling information. For SAP some dedicated partial solutions in this area have been devised. [12] But another selection for a first target ERP-system was advisable, because the basis of SAP is proprietary and also the size of the system with its complexity pose huge demands on everybody starting to work with the system. In respect of the mentioned criteria the open source ERP-system iDempiere has very advantageous characteristics. It contains adaptable workflows, multi-language and multi-currency features required for international business, and it supports more demanding enterprise functionalities including web-services for data exchange with other systems (comp. [13]). Besides, it is freely available and extensive documentation is published in the internet.

In Figure 1 an overview on the conceived interaction between the modelling system and ERP-system is presented. It depicts a unidirectional communication and is based on the first goals of the project. As described in the preceding chapter on the requirements, also more elaborated interfaces, e.g. exchanging in both directions and validation schemes are planned and investigated in regard of the extended vision of model support for ERP-configuration, development and support.

Figure 1. Overview on model driven configuration of iDempiere with ADOxx.



Central for the integration of the modelling tool ADOxx with the ERP-System iDempiere are the web-service interfaces for XML-data and its conversion and control between the two systems. The transfer may seem easy at first glance, but the analysis of the involved systems revealed substantial technical and logical differences, which had to be solved to transfer required information.

The core procedure in the overview starts with the modelling of workflows and the underlying organization in ADOxx. This model data is converted into XML-data and transferred to the interface. The interface transforms this data into a format suitable for the ERP-system iDempiere and forwards it to the web-service interface of iDempiere. The ERP-system checks the data and changes its own configuration when storing the data. Thereafter, all workflow instances will be executed following the scheme of the transferred models.

6 General outline of the course project

Previous projects were primarily common development projects focussed on the application of known technologies in a certain domain, that was general knowledge for the students (comp. Chapter 2). A research oriented project to develop some innovation is appealing because it constitutes an option to incorporate also other educative goals (see chap. 1). Especially, as it has a deeper learning perspective, in which new areas must be discovered, acquired and actively combined. These abilities are highly important for practice as technology proceeds with fast pace.

Table 3. Some anticipated challenges for the project

Anticipated challenges	*Countermeasures*
No complete definition of tasks	a) proposition of first steps b) openness for initiatives
Technical information on open source ERP-system often dispersed and sometimes outdated	Advice accordingly
High demand on abstraction as metamodeling is not concerned with direct reality, but with classifying elements in reality	Provision of example libraries in ADOxx
Difficult identification and representation of peculiarities in the established ERP-system	none

The new and augmented perspective of this course entails some challenges, as students cannot just apply learned knowledge, but they must also engage in novel topics. [14] In respect of the project in the centre of this report, most basic concepts have been taught in the courses before, like XML, process modelling and software development. But the concrete knowledge required is composed in a number of special ways, like new variants of programming languages e.g. AdoScript, Expressions, GraphRep, Node.js and architecture of iDempiere. Furthermore, also the organization of the project is challenging as it can be devised in many different ways.

Table 3 lists some of the potential problems which were anticipated before the project start. It also denotes some measures to help the students in the project to master them.

7 Advance of the first project and some experiences

At the beginning of the project the students were introduced to the functionality of the ERP-system, its flexibility and the underlying data to configure such systems with a presentation. Then the integration with enterprise models, like organizational charts and business processes was addressed focussed on the purpose of configuration of the ERP-system. Enterprise models contain a significant portion of information required to configure an ERP-system. The introductory session was complemented with references to documentation of relevant knowledge for the project, especially, about the systems in focus ADOxx and iDempiere.

Subsequently the students chose their task groups and started to investigate information on the topics of their tasks. The initial organization is depicted in Table 4 also noting the tasks and important knowledge for the team members. In the first sessions the students installed the software-systems on a number of computers. Some used preconfigured virtual machines with iDempiere, but others had to install their systems from scratch, because they could not use the virtual machines.

The documentation of ADOxx was well available, but nevertheless, its size and peculiarity posed some challenges to the students. The quality of documentation of iDempiere was not always consistent. Especially some older fragments confused the students, e.g. for web-services first some outdated documentation was found before the documentation of the current version was discovered.

The start of the project was particularly difficult. The students were looking for very concrete steps and did not appreciate the multitude of options and the freedom to design their own solutions and modelling languages. After some discussions first ideas came up. Most of them were fundamental to further development, only the outline of an underlying scenario of bakery processes proved to be too specific later. The scenario induced the group in charge to shape the symbols particularly for this domain. Later it had to be realigned, when the symbols were matched with the goal that the models should be applicable for all types of companies using the ERP-system.

After that, the individual groups explored and tested the required concepts and underlying systems. The data model of relevant iDempiere classes of objects were focal requirements. At first the classes of user and activity were investigated. The class definitions were provided by the ERP-system specialists. They compiled the data structure by analysing available data-base relationship diagrams and tables in the documentation and the user interfaces in iDempiere.

The symbols were designed by the business modelling specialists. The classes were implemented in ADOxx in corresponding object classes with particular symbols by the ADOxx technical group. At the beginning the symbols were very simplistic and there was little consideration on adjustment between the requirements of the individual groups. In many aspects the groups worked in parallel and investigated the mechanisms provided by the systems, e.g. the export mechanisms for XML-files were

investigated and the options to combine this procedure in a web-service was tested with AdoScript.

Generally, it was noted that ADOxx did not support the parallel collaboration of developers during the development of different facets of one modelling language library. Multi-user functionality for language development would be very instrumental, especially, if matured and standardized concepts will be implemented in large scale for many types of model elements in later phases.

Table 4. Initial organization of the project

Group	*Task*	*Relevant knowledge for the participants*
Project management	Organisation of cooperation overview on all tasks	Project management methods
Business modelling specialists	Definition of modelling languages, provision of examples	Business process and organizational modelling, experiences in a business domain
Modelling technical team	Implementation of new modelling languages and configuring the output web-service of the modelling tool	Technical infrastructure of ADOxx (GraphRep, AdoScript, Metamodelling Editor)
ERP-system specialists	Definition of fundamental model structures for model elements and cooperation for provision of examples	ERP-system functions, processes and data structures
ERP-system technical team	Preparing the input data for iDempiere, thereby conversion of XML-data	ERP-system web-services and technical data structures

The ERP-system technical team investigated the web-services to import data into the ERP-system. This feature is very powerful, as all data in the system, including configuration data, can be imported by web-services, but a number of particularities impeded its swift realization:

1. The web-services must be administered each at field level, a precaution to tailor their use quite strictly, by this improving the security.
2. The name of the web-service does not directly correspond with the name of the entity but is supplemented with some other fragment of the class name.
3. Relationships are stored with technical keys in the background. They are differently assigned in ADOxx and iDempiere. As the keys must be provided for a web-service to store a relationship, it must be retrieved in a first step to set up a respective web-service with containing a relation to another object. This is also fundamental for the sequence of web-services, if more connected elements e.g. an organisational chart shall be transferred.

For the interface several routines were conceptualized and programmed. The output from ADOxx was generated with an AdoScript. The data was transferred to a

JavaScript program based on the Node.js platform for converting the XML-data. This was necessary to prepare the XML-data for the iDempiere web-service. Another routine was intended to configure the inputs interface. In the final step, the model data was transferred to the iDempiere system by a web-service of the type "createData".

Table 5 lists main difficulties that were encountered during the project. Besides a number of technical difficulties, also project management encountered some obstacles, due to the complex nature of the interrelationships between the topics of the groups working in the project.

Table 5. Main difficulties during the project

Main difficulties	*Comment*
At the beginning students disliked the open definition of tasks given.	Solved by intensive discussions and later through more information gained in the process.
Project teams only focussed on their own task. Inputs for successive tasks of other groups were not prepared with their interest in mind. After some time the distribution of workload became quite unbalanced.	Redistribution of teams in the second phase, with more tasks crossing basic topic areas, e.g. by giving functional members tasks to document technical communication processes.
Project management group took an administrative approach to tasks required without bothering or discussing technical issues.	Due to this, emerging problems were not discovered early. This resulted in untimely deliveries of crucial deliveries. Therefore, some goals had to be abandoned.
Workflow definition in iDempiere and BPMN (basic pattern in ADOxx Bee-up) differs in some aspects, e.g. no explicit start element in iDempiere.	Problem was realized by students but as technical development did not demand a solution no concept mapping was performed.
High number of modelling elements for configuration and their interdependencies suggested a generic solution for interface routines.	Only few students were able to design and implement corresponding concepts.
Complexity of interface higher than estimated a) web-service security required individual configuration of each service b) relationships of elements require technical keys which had to be retrieved before storing connected elements	Appropriate concepts were developed additionally by some students, but this slowed down progress in other areas significantly

In the end the project realized first transfers of data between ADOxx and iDempiere, thereby, proving the feasibility of the underlying ideas. The transfers were not automated and just covered users and roles. But they are deemed to be a good foundation for the follow-up project, as numerous ambiguities from the start of the first project have been clarified and information on the concepts have been

documented extensively. The students of the next study year were already introduced to the topic in the final presentation of the first project.

8 Reflection and outlook

The critical review of the project revealed some options for improvement. This was expected to some extent, as the innovative kind of project incorporated a higher degree of required learning and establishing an appropriate direction. Before the project, the lecturers assumed that the students would like to have a high degree of self-determination. But in the course of the first weeks of the project it became clear, that this freedom overstrained the student group, especially in combination with the needs to learn new concepts. So later the lecturers provided additional guidance. Nevertheless, the problems in the beginning resulted in a modification of the project goals for the development of the interface.

In regard of the study curriculum and the purpose of a course project most aspects have been fulfilled, especially in regard of: a) the *importance of reconciliation of intermediate results*, b) *crucial cooperation* c) often *required changes of a preconceived design* due to results of analysis. Only the final testing phase had been abandoned in favour of a thorough documentation.

As has been indicated before, the follow-up project will incorporate the findings and experiences of its predecessor, thereby practicing a reflective approach in the development of a domain specific modelling language [4]. One general aspect is that communication must be organized and orchestrated more deliberately. This particularly applies to subjects that are no standard deliverables but interconnected sophisticated concepts (comp. [14]). Also better starting conditions in the technical infrastructure should be provided, so that the project will be fully operational from the beginning.

Two main objectives seem most important for the second course project:
a) enlargement of transferred information to complete workflows,
b) enhancements for the technical conversion of XML-data evaluating some options.

Two promising options in regard of the handling the XML-exchange were identified during or at the end of the first project. They are:
b1) more or less preparation of data in ADOxx for final usage in iDempiere
b2) incorporation of standard information brokers and conversion routines instead of rather custom programs for the conversion tools in the first project [15].

The list of requirements conveys a vision of numerous options for further development (comp. Table 1 and 2), which cannot be accomplished in one year. This leaves quite a few promising further steps of evolution for innovating the facilities for a swift and easy adoption of ERP-systems by models. By this, the research aspires to enhance the transparency of ERP-system configuration and facilitate control of these systems for IT- specialists and manager, and thereby to realize a model driven systems configuration for ERP-systems (MDSC analogue to MDSD [16]).

References

1. Montesa-Andres, J.O., Garrigós-Simón, F.J., Narangajavana, N.: A Proposal for Using Lego Serious Play in Education. In: M. Peris-Ortiz et al. (eds.), Innovation and Teaching Technologies: New Directions in Research, Practice and Policy, Springer, Cham, pp. 99-107 (2014)
2. Bonesso, S., Comacchio, A.: Leveraging on Projects to Strategically Organize Open Innovation. In: S. Bonesso et al. (eds.), Project-Based Knowledge in Organizing Open Innovation, Springer, London, pp. 1-18 (2014)
3. Becker, M., Errichiello, L., Zirpoli, F.: A Project-Based Perspective on Complex Product Development. In: S. Bonesso et al. (eds.), Project-Based Knowledge in Organizing Open Innovation, Springer, London, pp. 51-67 (2014)
4. Jannaber S., Riehle D.M., Delfmann P., Thomas O., Becker J.: Designing a Framework for the Development of Domain-Specific Process Modelling Languages. In: Maedche A., vom Brocke J., Hevner A. (eds) Designing the Digital Transformation. DESRIST 2017. Lecture Notes in Computer Science, vol 10243. Springer (2017)
5. Senge, P. M.: The Fifth Discipline: The Art and Practice of the Learning Organization. Doubleday Books, New York (1990)
6. Fettke P., Loos P., Zwicker J.: Business Process Reference Models: Survey and Classification. In: Bussler C.J., Haller A. (eds.) Business Process Management Workshops. BPM 2005. LNCS, vol. 3812. Springer (2006)
7. Wortmann J.C.: Evolution of ERP Systems. In: Bititci U.S., Carrie A.S. (eds) Strategic Management of the Manufacturing Value Chain. IFIP — The International Federation for Information Processing, vol 2. Springer, Boston, MA (1998)
8. Davenport, T.: Putting the Enterprise into the Enterprise System. In: Harvard Business Review, 76 no 4, pp. 121 - 131 (1998)
9. Worster, A.J., Weirich, T.R., Andera, F.J.C.: Maximizing Return on Investment Using ERP Applications. Wiley & Sons (2012)
10. Frank, U., Strecker, S.: Beyond ERP systems: An outline of self-referential enterprise systems. Requirements, conceptual foundation and design options. ICB Research Reports No 31, University of Duisburg-Essen (2009)
11. Fill, H-G., Karagiannis, D..: On the Conceptualisation of Modelling Methods Using the ADOxx Meta Modelling Platform. Enterprise Modelling and Information Systems Architectures 8(1): 4-25 (2013)
12. Götzfried M., Guppenberger M., Reiter M., Plechinger F.: Enabling a SAP ERP Based Standard Software Solution for BPM. In: Fischer H., Schneeberger J. (eds) S-BPM ONE - Running Processes. S-BPM ONE 2013. Communications in Computer and Information Science, vol 360. Springer, Berlin, Heidelberg (2013)
13. Surawijaya, S., Prakoso, G., Syahrial, I., Nina E.: Evaluation of ERP Selection System in Small & Medium Workshop Enterprise. In: Journal of Scientific and Engineering Research, vol. 4, issue 7, pp.134-139 (2017)
14. Badke, W.: Teaching Research Processes. Chandos Publishing, Oxford, (2012)
15. Hohpe, G., Woolf, B.: Enterprise Integration Patterns, Addison-Wesley, Boston MA (2004)
16. Roser, S., Lautenbacher, F., Bauer, B.: MDSD light for ERP. In Proc. of the ACM Symposium on Applied Computing (SAC '08). ACM, New York, pp. 1042-1047 (2008)