JOURNAL OF APPLIED
COMPUTER SCIENCE
Vol. 20 No. 1 (2012), pp. 7-21

Business Process- and Graph Grammar-Based Approach to ERP System Modelling

Mariusz Flasiński, Janusz Jurek

Jagiellonian University
IT Systems Department
ul. prof. St. Lojasiewicza 4, Cracow 30-348, Poland
mariusz.flasinski@uj.edu.pl

Abstract. *Methods of ERP (Enterprise Resource Planning) systems modelling are presented in the paper. The modelling is necessary to adapt an ERP system to a company in such a way, that it fully supports a management at the operational level, i.e. it supports the business processes in the best possible way. The choice of methods (business process and graph grammar-based) described in the paper is based on the authors' experiences in several dozen of implementation projects, realized in large Polish enterprises. The methods can be used to select the best ERP system for a company, to design its implementation, and to customize the system accordingly to the requirements of a company.*
Keywords: *ERP Systems, Business Processes, Graph Grammars.*

1. Introduction

In the paper we discuss the problem of integration of a business process architecture of a company and an ERP (Enterprise Resource Planning) system [1, 2]. The integration means such an adaptation of an ERP system in a company, that the system fully supports a management at the operational level, i.e. it supports the business processes in the best possible way. The main requirement for achieving

the integration is a proper model of an ERP system. Our approach to ERP system modelling assumes the use of business processes maps and graph grammars. The approach is based on our own experiences in several dozen of implementation projects, realized in large Polish enterprises (eg. *LOTOS Group* - a national petroleum concern, *PGE Polska Grupa Energetyczna* – the largest electricity company in Poland, *ENERGA Group* – a large Polish energy supplier with over twelve thousand employees, *Mostostal Warszawa Group* – one of the main players on Polish construction market, acting as a general contractor of investment projects, *The Poznan International Fair* – the leader of the Polish trade fair market and the second largest exhibition organizer in Central Eastern Europe, *USP Group* – a dynamically developing company operating in healthcare industry on Central and Eastern European markets). All these experiences concern whole projects of ERP system implementation. Such projects consist of several phases: the selection of the most suitable ERP system, the design how to embed the system in a company, the configuration and customization of the system, the implementation of necessary program add-ons, data migration, training of users, and finally the start of productive exploitation of the system (system is being used to support business activities in a company by its employees). It's worth to point out that the projects of ERP system implementation could be as much difficult and time-consuming as the projects of a construction of a new software system being the main topic of interest of software engineering [3].

In section 2 we describe a company as a system of business processes (on the ground of the theory of management). In section 3 we show models of process description in software engineering discipline. Section 4 contains the choice of practical methods and tools based on process approach, which could be used to support each separate phase of an ERP system implementation. Advanced applications of models based on business processes and graph grammars are shown in section 5. Concluding remarks are included in the final section.

2. Company as a system of processes

One of the main approaches in the management theory is based on perceiving a company as an adaptive system. This approach has been initiated by Ludwig von Bertalanffy, and then developed in cybernetics sciences. The company could be treated as a system consisting of many collaborating sub-systems (which again could consist of smaller sub-systems). The system functions to realize certain (su-

perior, strategic) goals while its sub-systems function to realize intermediate (minor) goals. It is related to the concept of synergy: elements collaborating witch each other in a system, could perform better than individually.

At the end of last century, *business processes* of a company have become the center of attention in the field of the management theory and the system approach. A business process is a sequence of activities or tasks inside a company, performed in its different units, that produces a specific service or product (serve a particular goal) for a particular customer (inside or outside the company). As a result of the investigation of the role of business processes, the new models of a company management at the operational level have appeared, as well as the models of improvement or reengineering of the management through processes called *BPR – Business Process Reengineering* and *BPI – Business Process Improvement* [4, 5]. All these models are based on the description of company functioning in the form of *business process maps*. Such a map is a graph, where the nodes are process steps (related to activities, task or decisions), and the edges between nodes show the sequence of performing the steps. The graph is placed on a table which rows correspond to particular organization units in a company. If a given process step is shown in a row labelled by a unit, than it means that the unit is responsible for realization of the step. An example of a process map is shown in Figure 1.

A company can be viewed as a hierarchical system of business processes. We can identify several main processes in a company, then their sub-processes, and then sub-processes of the (higher level) sub-processes, and so on. When a (sub)process is small enough and cohesive we can model a process map showing its functioning.

Although the model of viewing a company as a system of processes has been developed in management sciences (for business improvement purposes), there are some equivalent models developed independently in software engineering discipline. The models are described in next section.

3. Software engineering models related to processes

Process models started to appear in software engineering in the seventies of last century. Well-known tools for requirements modelling are data flow diagrams (DFD) developed on the ground of the structural methodology [6]. Data flow diagrams can be used to reflect process architecture of a company. Although they serve to show rather data and data transformation, than the sequence of activities,

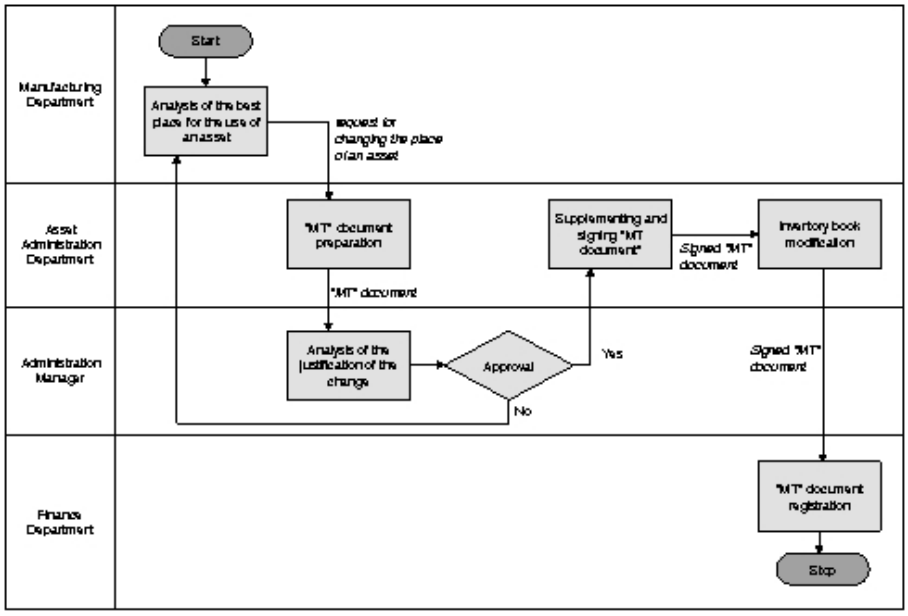

Figure 1. An example of a process map

they can be also used to model processes in a company in a hierarchical way (at different levels of detail). Figure 2 contains an example of a data flow diagram related to the process map in Figure 1.

In object-oriented approach and the UML [7, 8] there are also ways to show business processes in a company. Activity diagrams can be used to identify tasks and objects (eg. documents) in a company. Let us notice that such diagrams can contain not only activities performed in a software system, but performed by a human as well. An example of activity diagram is presented in Figure 3. It corresponds to the process in Figure 1.

Recently, one can observe dynamical development of such models of business processes description that could be directly used to construct software supporting the processes. Let us mention BPMN (Business Process Modelling Notation), i.e. graphical notation for a very detailed representation of business processes developed and promoted by Business Process Management Initiative. This notation is similar to the one shown in Figure 1, but it contains much more advanced elements (eg. gateways or artifacts). On the other hand, there are some "technical"

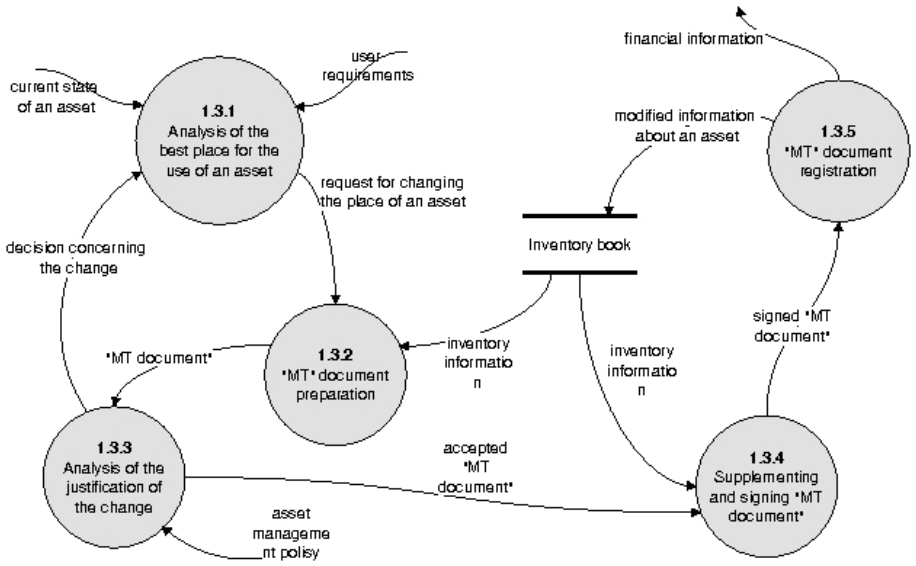

Figure 2. An example of a data-flow diagram

(not graphical) languages for business process description like BPML (Business Process Modelling Language), also developed by Business Process Management Initiative, and BPEL (Business Process Execution Language for Web Services) developed by IBM, Microsoft, BEA Systems and other organizations. Both languages could be used for web services specification.

4. Process models in ERP system implementation projects

ERP systems have been designed for the support of a company management mainly at the operational level. They are derived from MRP systems (Material Requirements Planning), and then MRP II systems (Manufacturing Resource Planning), i.e. systems for manufacturing management. The operational level in a company is based on business processes. Therefore the process model of a company should be also a base for ERP system implementation [9, 10].

In following subsections we describe methods of the use of a process model during different phases of an ERP implementation project: during the selection of a system, during design and configuration of the system, and during the construc-

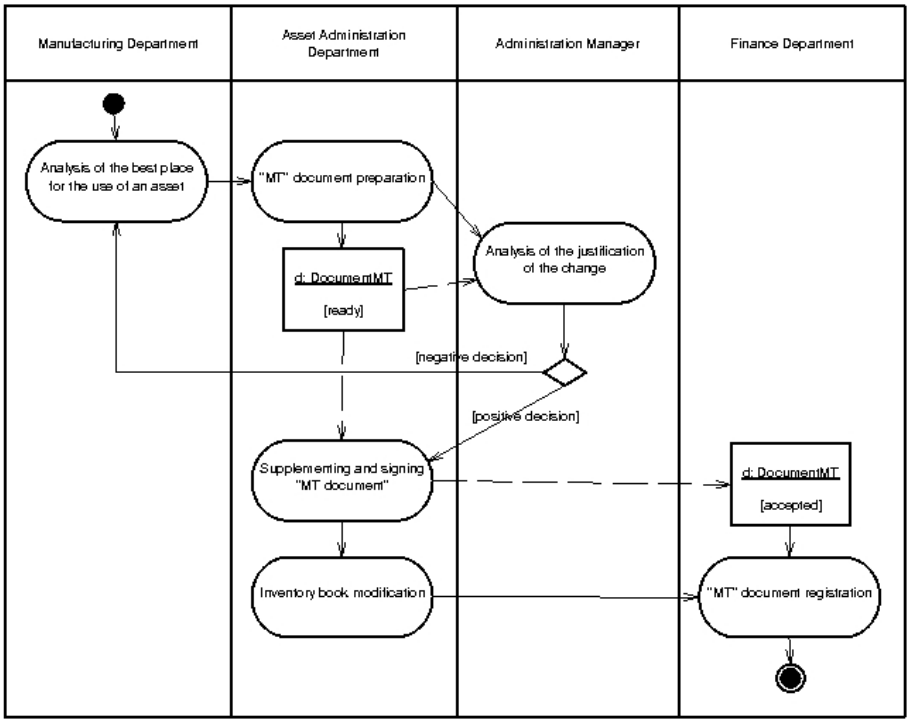

Figure 3. An example of an activity diagram

tion of program add-ons. As we've noticed in the introduction, these methods are based on our own experiences related to the project realized in several dozen Polish companies.

4.1. Selection of ERP system

ERP systems are Commercial-Off-The-Shelf (COTS) software. Making a choice of an ERP system for our company we should select such system that can support business processes in the best possible way. Therefore user requirements analysis (proceeding the selection procedure) should be based on a process model. Each process, sub-process and process step should be accompanied with functional requirements expected to be fulfilled in a system. The result of the user requirements analysis is then a list of all functionalities of an ERP system, which are neces-

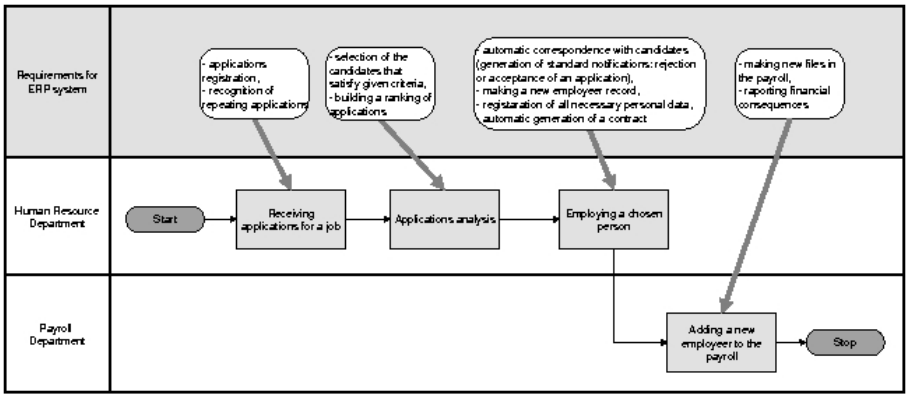

Figure 4. Assigning ERP system requirements to process steps

sary to fully support business processes (usually no less than several hundred of requirements).

In previous section we have shown that there are different methods and models of business process description: originated from the management theory (process maps) and from the software engineering discipline (data flow diagrams, activity diagrams, BPEL, BPML, etc.). Our goal is the identification of requirements for an ERP system, so we should use a tool which is as simple as possible and easy to understand for people who are not computer specialists (i.e. employees in a company responsible for formulating the requirements). As it has been proven in practice, such a tool is a process map (see: Figure 1). What's more, a process map should not contain any excess elements (eg. supported by BPMN), which are not necessary for the identification of the requirements.

For the user requirements analysis purposes, a process map (in its simplest form) is supplemented by a row: "Requirements for ERP system". This row is used to specify detailed requirements related to particular process steps as it has been shown in Figure 4. Each step in a process map should be analyzed by a person who is responsible for this step, to identify all requirements that should be fulfilled by the system (functional and technological, like for example maximum execution time).

The list of all requirements for an ERP system should be the base of the request for an offer, that is send to ERP software vendors and IT firms specializing in ERP implementation services. If it is possible (taking into account confidentiality policy

Process index and name	xxx						
Description of the process	xxx						
Process input	xxx						
Process output	xxx						
Process step index	Description of the process step	ERP system transaction	Description of the ERP system support	Necessary program add-ons	Printouts	Reports	Interfaces
xxx	xxx	xxx	xxx	xxx	xxx	xxx	xxx
xxx	xxx	xxx	xxx	xxx	xxx	xxx	xxx
...	...	...	...	...	...	...	...

Figure 5. An example of a form supporting design of an ERP system

of a company), we recommend to attach a whole process model of the company (in the form of process maps) to the request. It could help the tenderer to understand how the company works and what most important business needs are. As the result it should allow to construct an offer more precisely and better assess the price of the implementation project.

4.2. Design and configuration of ERP system

The functional requirements for an ERP system are the result of the user requirement analysis (needed to select a proper system). The goal of the design phase, is to define how these requirements are to be fulfilled with the use of the purchased ERP system.

The design specifying how to configure an ERP system should be based on the process model previously developed. System transactions realizing needed software support should be defined for each step of each process being in the scope of the ERP system implementation. On the basis of our experiences, we can notice that a correct and complete design of a ERP system should specify for each process at least elements shown in a table (form) below.

The use of the form allows to create such a design of an ERP system that is oriented on business processes in a company. The well-known weakness of many implementation projects is "module orientation". It means that the implementation of an ERP system is treated in this case like a several independent implementation projects for each of the modules of the system. As the result we can obtain a system that supports business units in a company (each one separately), but does not support business processes flowing through many units. As it is practically

proven (eg. [5], the greatest opportunities for performance improvement lie in the interfaces between business units. Therefore we should assure that the implementation of an ERP system is oriented on processes that cut across unit (functional) boundaries.

4.3. Construction of program add-ons

Even if we choose the best ERP system with respect to the user requirements, it could be necessary to construct some program add-ons that customize system according to the user needs and enrich it functional possibilities.

In such a case, a process model should be the base for defining the construction design of a suitable program add-on. This design can be prepared in object-oriented methodology (e.g. with the use of activity diagrams, see: Figure 3) or in structural methodology (with the use of data flow diagrams, see: Figure 2). The problem of software engineering models related to processes is discussed in section 3. Let us only stress that the construction design should necessarily reflect the specificity of a business process that is to be supported.

5. Graph grammar-based approach

Several applications of a process model in ERP implementation projects have been shown in previous sections. They are relatively typical and should be a standard element in all ERP projects. Nevertheless, it worth to point out that there are also advanced applications of a process model. Let us present the results of our research concerning such applications.

Firstly, let us notice that both process models and software models are subject to modifications. As it has been already mentioned, the improvement of the management at the operational level is based mainly on the improvement of business processes. Therefore, the processes (and their model) should "live" and be changing permanently in order to enhance their efficiency. On the other hand, the ERP system supporting the processes should also be changing accordingly to the alterations in processes. Hence, we need a tool for the control and support the changes both in "the problem domain" (process maps), and in "the solution domain" (software models related to the processes, eg. UML diagrams).

Our research has started from an observation that a business process map is a graph structure from a formal point of view. Therefore, to analyze business processes we should construct efficient algorithms allowing us to analyze a graph

structure corresponding to a structure of the process. Recently, an efficient formal model for an analysis of graph structures has been constructed and verified in many different practical applications [11, 12, 13, 14]. This model is based on graph grammars. Additionally, we assume the use of string grammars of a big descriptive power to control the changes [15, 16, 17]. In order to apply our approach, we have to define a structure of a business process as a graph belonging to an ETPL(k) graph language. Such a graph is called an IE-graph and it is defined as below [11].

Definition 1 *An indexed edge-unambiguous graph, IE-graph over* Σ *and* Γ *is a quintuple*

$$H = (V, E, \Sigma, \Gamma, \phi), \text{ where}$$

V is a finite, non-empty set of nodes that indices have been ascribed to in an unambiguous way on the basis of an object represented with a graph,

Σ *is a finite non-empty set of node labels,*

Γ *is a finite non-empty set of edge labels,*

E is a set of edges of the form (v, γ, w)*, where* $v, w \in V, \gamma \in \Gamma$*, such that index of v is less than index of w,*

$\phi : V \longrightarrow \Sigma$ *is a node-labelling function.*

As a tool for analysis a graph structure we use ETPL(k) graph grammars. Now, let us define such a graph grammar [11].

Definition 2 *An ETPL(k) graph grammar G is a grammar of the Rozenberg edNLC class defined in the following way.*

$$G = (\Sigma, \Delta, \Gamma, P, Z), \text{ where}$$

Σ *is a finite nonempty set of node labels,*

$\Delta \subseteq \Sigma$ *is a set of terminal node labels,*

Γ *is a finite nonempty set of edge labels,*

P is a finite set of productions of the form (l, R, C)*, where*

$l \in \Sigma$,

R is the right-hand side IE-graph over Σ *and* Γ *of at most two levels [11],*

$C : \Gamma \times \{in, out\} \longrightarrow 2^{\Sigma \times \Sigma \times \Gamma \times \{in,out\}}$ *is the embedding transformation having a property of preserving a potential previous context [11],*

Z is the starting IE-graph over Σ *and* Γ*, called the axiom,*

and the derivation of the grammar fulfils two following conditions:

- *it is closed in relation to the regular left-hand side derivation [11] and*
- *it is k-left nodes unambiguous [11].*

For the ETPL(k) graph grammar we can define the efficient ETPL(k) syntax analyzer [11] and efficient ETPL(k) inference algorithm [13]. In this way it is possible to analyze the graph structure corresponding to a structure of a business process. Moreover, if we have a base of reference business processes, we can construct a self-learning system (with the use of the grammatical inference algorithm) that is able to analyze thousands of business process maps in order to find the common pattern of the business process improvements.

In our approach changes in business processes are reflected by the applications of the graph grammar productions. Production labels can be treated as strings in a given string language. We construct a GDPLL(k) string grammar generating the language and we use it as a tool for the analysis and control of changes in the business processes (since GDPLL(k) grammars are of a very big generative/discriminative power and there is an efficient syntax analyzers for the grammars). Let us define the grammars [15, 16].

Definition 3 *A generalized dynamically programmed LL(k) grammar, GDPLL(k) grammar, is a six-tuple:*

$$G = (V, \Sigma, O, P, S, M), \text{ where}$$

V is a finite, nonempty alphabet;

$\Sigma \subset V$ *is a finite, nonempty set of terminal symbols (with N we denote a set of nonterminal symbols* $N = V \setminus \Sigma$*);*

O is a set of basic operations on the values stored in the memory (assignment, addition, subtraction, multiplication);

$S \in N$ is the starting symbol;

M is the memory;

P is a finite set of productions of the form:

$$p_i = (\mu_i, L_i, R_i, A_i), \text{ in which:}$$

$\mu_i : M \longrightarrow \{TRUE, FALSE\}$ is the predicate of applicability of the production p_i defined with the use of operations ($\in O$) performed over M;

$L_i \in N$ and $R_i \in V^*$ are left- and right-hand sides of p_i respectively; a pair (L_i, R_i) will be called a core of p_i;

A_i is the sequence of operations ($\in O$) over M, which should be performed if the production is to be applied,

and the derivation of the grammar fulfils two following conditions:

- *the LL(k) condition of deterministic derivation is fulfilled, and:*
- *the number of steps during derivation of any terminal symbol is limited by a constant.*

Formal specifications of the two conditions are included in [15].

The results we have already achieved are very promising. The graph and string grammars model has good computational properties as well as is of a strong descriptive power. It can be used not only for the improvement of the management at the operational level [14], but for the support of the management at the strategic level as well [18]. Our research is now aimed at developing methods of automatic (or semi-automatic) reconfiguration of an ERP system on the base of the changes made in business processes. In order to achieve this goal we are going to define a translation scheme: productions applied to change processes should be converted to the sequence of configuration actions in an ERP system (of course it strictly depends on a particular systems and its programming language).

6. Conclusions

In today's business world, ERP systems play a critical role. ERP systems create plenty of opportunities for growth and increased productivity of a company. However, the success of using an ERP in a company depends on many vital conditions. One of the most important conditions is the use of the business process approach in all phases of an ERP system implementation project. In the paper we've described some methods of the application of the business process approach in the context of ERP system modelling. They can be used to select the best ERP system for a company, to design its implementation, and to customize the system accordingly to the requirements of a company. The selection of methods is based on the authors' experiences in implementation projects realized in large Polish enterprises. The business process approach develops dynamically both in management theory and computer science. New methods and advanced tools of business process optimization and software support have appeared recently. An automation of IT management systems modelling seems to be the main objective of a research in this area [19, 20]. One of the biggest challenges in this field is to provide a mechanism of automatic generation of an ERP system configuration on the base of a business process model. Graph grammars and syntax directed translation are powerful formal tools for achieving this goal. The results of the research into this area will be the subject of further publications.

References

[1] Vaman, J., *ERP in Practice*, Tata McGraw-Hill Publishing Company Ltd., 2007.

[2] Leon, A., *Enterprise Resource Planning, 2nd Edition*, Tata McGraw-Hill Publishing Company Ltd., 2008.

[3] Flasiński, M., *IT Project Management*, Polish Scientific Publishers PWN, 2006, (in Polish).

[4] Hammer, M. and Champy, J., *Reengineering the Corporation*, Harper Business, 1993.

[5] Rummler, G. and Brache, A., *Improving Performance*, Wiley, 1995.

[6] Yourdon, E., *Modern Structured Analysis*, Prentice Hall, 1988.

[7] Rumbaugh, J., Jacobson, I., and Booch, G., *The Unified Modeling Language Reference Manual*, Addison-Wesley, 2004.

[8] Rittgen, P. e., *Enterprise Modeling and Computing with UML*, Idea Group Publishing, 2007.

[9] Monk, E. and Wagner, B., *Concepts In Enterprise Resource Planning, 3rd Edition*, Course Technology, 2008.

[10] Magal, S. and Word, J., *Essentials of Business Processes and Information Systems*, John Wiley & Sons, 2009.

[11] Flasiński, M., *On the parsing of deterministic graph languages for syntactic pattern recognition*, Pattern Recognition, Vol. 26, 1993, pp. 1–16.

[12] Flasiński, M., *Power properties of NLC graph grammars with a polynomial membership problem*, Theoretical Computer Science, Vol. 201, 1998, pp. 189–231.

[13] Flasiński, M., *Inference of parsable graph grammars for syntactic pattern recognition*, Fundamenta Informaticae, Vol. 80, 2007, pp. 379–413.

[14] Flasiński, M., *On the use of graph grammars for business process optimization*, Foundations of Control and Management Science, Vol. 9, 2008, pp. 23–28.

[15] Flasiński, M. and Jurek, J., *Dynamically programmed automata for quasi context sensitive languages as a tool for inference support in pattern recognition-based real-time control expert systems*, Pattern Recognition, Vol. 32, 1999, pp. 671–690.

[16] Jurek, J., *Recent developments of the syntactic pattern recognition model based on quasi-context sensitive languages*, Pattern Recognition Letters, Vol. 26, 2005, pp. 1011–1018.

[17] Jurek, J., *Grammatical inference as a tool for constructing self-learning syntactic pattern recognition-based agents*, Lecture Notes in Computer Science, Vol. 5103, 2005, pp. 1011–1018.

[18] Jurek, J., *On the use of syntactic pattern recognition methods for strategic management*, System Science, Vol. 35, 2009, pp. 55–61.

[19] Tadeusiewicz, R., Ogiela, L., and Ogiela, M., *The automatic understanding approach to systems analysis and design*, International Journal of Information Management, Vol. 28, 2008, pp. 38–48.

[20] Tadeusiewicz, R., Ogiela, M., and Ogiela, L., *Automatic information processing and understanding in cognitive business systems*, In: Proceedings of 11th International Conference on Enterprise Information System, 2009.